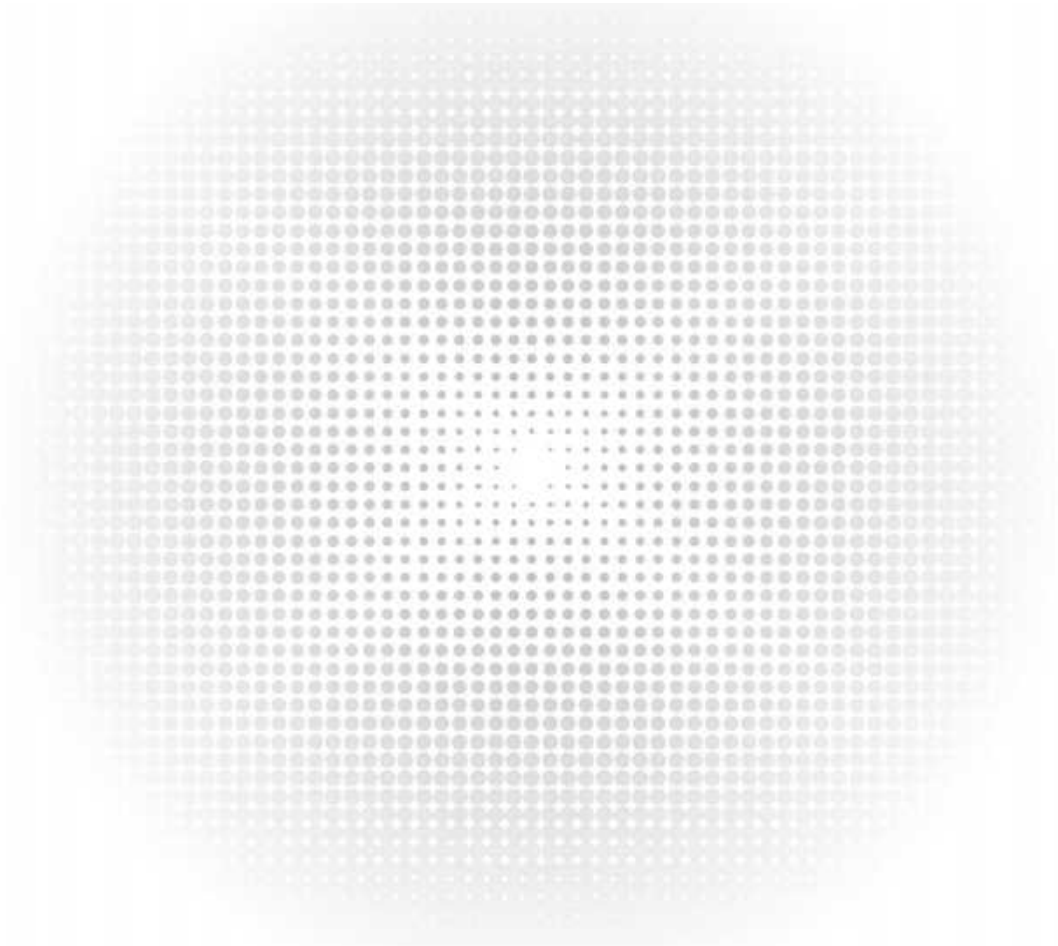
S5850-24XMG-U Switch Configuration Guide

Models: S5850-24XMG-U



Configuration Guide

Contents

1 Preface 24
1.1 Declaration 24
1.2 Suggestion feedback 24
1.3 Audience 24
2 Basic Configuration Guide 25
2.1 Configuring System Management 25
2.1.1 Overview 25
2.1.2 Configuration 25
2.1.3 Application cases 26
2.2 Configuring User Management 27
2.2.1 Overview 27
2.2.2 Configuration 27
2.2.3 Application cases 28
2.3 Configuring FTP 29
2.3.1 Overview 29
2.3.2 Configuration 29
2.3.3 Application cases 30
2.4 Configuring FTP SERVER 30
2.4.1 Overview 30
2.4.2 Configuration 30
2.4.3 Application cases 31
2.5 Configuring TFTP 31
2.5.1 Overview 31
2.5.2 Configuration 31
2.5.3 Application cases 32
2.6 Configuring SCP 32
2.6.1 Overview 32
2.6.2 Configuration 32
2.6.3 Application cases 32
2.7 Configuring Telnet 33
2.7.1 Overview 33
2.7.2 Configuration 33
2.7.3 Application cases 34
2.8 Configuring SSH 34
2.8.1 Overview 34
2.8.2 Configuration 34
2.8.3 Application cases 35
2.9 Configuring Time&timezone 35
2.9.1 Overview 35
2.9.2 Configuration 35

2.9.3 Application cases 36
2.10 Configuring License 36
2.10.1 Overview 36
2.10.2 Configuration 36
2.11 RPC API Configuration Guide 37
2.11.1 Overview 37
2.11.2 Configuration 40
2.11.3 Application cases 41
2.12 Configuring HTTP 41
2.12.1 Overview 41
2.12.2 Configuration 41
2.12.3 Application cases 42
3 Ethernet Configuration Guide 43
3.1 Configuring Interface 43
3.1.1 Overview 43
3.1.2 Configuration 43
3.1.3 Application cases 45
3.2 Configuring Group-speed of Interfaces 45
3.2.1 Overview 45
3.2.2 Configuration 45
3.2.3 Application cases 46
3.3 Configuring Layer3 Interfaces 46
3.3.1 Overview 46
3.3.2 Configuration 47
3.4 Configuring Interface Errdisable 49
3.4.1 Overview 49
3.4.2 Configuration 50
3.4.3 FAQ 52
3.5 Configuring MAC Address Table 53
3.5.1 Overview 53
3.5.2 Configuration 53
3.5.3 Application cases 58
3.6 Configuring VLAN 58
3.6.1 Overview 58
3.6.2 Configuration 59
3.6.3 Application cases 61
3.7 Configuring Voice VLAN 61
3.7.1 Overview 61
3.7.2 Configuration 62
3.7.3 Application cases 62
3.8 Configuring VLAN Classification 63
3.8.1 Overview 63
3.8.2 Configuration 63

3.8.3 Application cases 65
3.9 Configuring VLAN Mapping 65
3.9.1 Overview 65
3.9.2 Configuration 66
3.9.3 Application cases 71
3.10 Configuring Link Aggregation 71
3.10.1 Overview 71
3.10.2 Restrictions and Precautions 77
3.10.3 Configuration 79
3.10.4 Application cases 89
3.11 Configuring Flow Control 93
3.11.1 Overview 93
3.11.2 Configuration 93
3.11.3 Application cases 94
3.12 Configuring Storm Control 94
3.12.1 Overview 94
3.12.2 Configuration 95
3.12.3 Application cases 96
3.13 Configuring Loopback Detection 96
3.13.1 Overview 96
3.13.2 Configuration 96
3.13.3 Application cases 98
3.14 Configuring Layer 2 Protocols Tunneling 98
3.14.1 Overview 98
3.14.2 Configuration 99
3.14.3 Application cases 101
3.15 Configuring MSTP 101
3.15.1 Overview 101
3.15.2 Configuration 102
3.15.3 Application cases 106
3.16 Configuring MLAG 106
3.16.1 Overview 106
3.16.2 Restrictions and Precautions 111
3.16.3 Configuration 111
3.17 Configuring Hash Load-balance 145
3.17.1 Configuring Linkagg Hash 145
3.17.2 Configuring ECMP Hash 154
3.17.3 Configuring EFD Hash 160
3.18 Configuring PORT-XCONNECT 160
3.18.1 Overview 160
3.18.2 Configuration 160
3.18.3 Application cases 161
4 IP Service Configuration Guide 162

4.1 Configuring Arp 162
4.1.1 Overview 162
4.1.2 Configuration 162
4.1.3 Application cases 164
4.2 Configuring Arp proxy 164
4.2.1 Overview 164
4.2.2 Configuration 164
4.2.3 Application cases 170
4.3 ConfiguringARP host-route 170
4.3.1 Overview 170
4.3.2 Configuration Examples 170
4.4 Configuring DHCP Client 171
4.4.1 Overview 171
4.4.2 Configuration 171
4.4.3 Application cases 173
4.5 Configuring DHCP Relay 173
4.5.1 Overview 173
4.5.2 Configuration 173
4.5.3 Application cases 175
4.6 Configuring DHCP server 175
4.6.1 Overview 175
4.6.2 Configuration 175
4.6.3 Application cases 180
4.7 Configuring DNS 180
4.7.1 Overview 180
4.7.2 Configuration 180
4.7.3 Application cases 181
5 IP Routing Configuration Guide 182
5.1 Configuring IP Unicast-Routing 182
5.1.1 Overview 182
5.1.2 Configuration 182
5.1.3 Application cases 184
5.2 Configuring RIP 184
5.2.1 Overview 184
5.2.2 Configuration 185
5.2.3 Application cases 206
5.3 Configuring OSPF 206
5.3.1 Overview 206
5.3.2 Configuration 206
5.3.3 Application cases 227
5.4 Configuring Prefix-list 230
5.4.1 Overview 230
5.4.2 Configuration 230

5.4.3 Application cases 232
5.5 Configuring Route-map 233
5.5.1 Overview 233
5.5.2 Configuration 233
5.5.3 Application cases 235
5.6 Configuring Policy-Based Routing 235
5.6.1 Overview 235
5.6.2 Configuration 235
5.6.3 Application cases 239
5.6.4 Configuring BGP 239
5.6.5 Overview 239
5.6.6 Configuration 239
5.6.7 Application cases 244
5.7 Configuring ISIS 244
5.7.1 Overview 244
5.7.2 Configuration 245
5.7.3 Application cases 248
6 Multicast Configuration Guide 249
6.1 Configuring IP Multicast-Routing 249
6.1.1 Overview 249
6.1.2 Configuration 249
6.1.3 Application cases 249
6.2 Configuring IGMP 249
6.2.1 Overview 249
6.2.2 Configuration 250
6.2.3 Application cases 252
6.3 Configuring PIM-SM 252
6.3.1 Overview 252
6.3.2 Configuration 254
6.3.3 Application cases 260
6.4 Configuring PIM-DM 260
6.4.1 Overview 260
6.4.2 Configuration 261
6.4.3 Application cases 262
6.5 Configuring IGMP Snooping 263
6.5.1 Overview 263
6.5.2 Configuration 263
6.5.3 Application cases 268
6.6 Configuring MVR 268
6.6.1 Overview 268
6.6.2 Configuration 269
6.6.3 Application cases 271
7 PoE 272

7.1 PoE Overview **272**
7.2 Configuring Global PoE Non-Standard Compatibility **272**
7.3 Configuring Global PoE power management scheme **272**
7.4 Configuring power-reserved **279**
7.5 Configure threshold-power-percentage **279**
7.6 Enable port PoE feature **279**
7.7 Configure max-power supply **279**
7.8 Configuring the Power Supply Priority **279**
7.9 Configure Port Compatibility **280**
7.10 Trigger an autoclass measurement **280**
7.11 Example for Configuring PoE **280**
8 Security Configuration Guide **282**
8.1 Configuring Port Security **282**
8.1.1 Overview 282
8.1.2 Configuration 282
8.1.3 Application cases 284
8.2 Configuring Vlan Security **284**
8.2.1 Overview 284
8.2.2 Configuration 284
8.2.3 Application cases 285
8.3 Configuring Time-Range **285**
8.3.1 Overview 285
8.3.2 Configuration 286
8.3.3 Application cases 286
8.4 Configuring ACL **286**
8.4.1 Overview 286
8.4.2 Configuration 292
8.5 Configuring Extern ACL **304**
8.5.1 Overview 304
8.5.2 Configuration 304
8.6 Configuring IPv6 ACL **306**
8.6.1 Overview 306
8.6.2 Configuration 306
8.6.3 Application cases 308
8.7 Configuring Port-Group **308**
8.7.1 Overview 308
8.7.2 Configuration 308
8.7.3 Application cases 308
8.8 Configuring Vlan-Group **309**
8.8.1 Overview 309
8.8.2 Configuration 309
8.8.3 Application cases 309
8.9 Configuring COPP ACL **309**

8.9.1 Overview 309
8.9.2 Configuration 310
8.9.3 Application cases 311
8.10 Configuring dot1x 311
8.10.1 Overview 311
8.10.2 Configuration 312
8.10.3 Application cases 315
8.11 Configuring Guest VLAN 316
8.11.1 Overview 316
8.11.2 Configuration 317
8.11.3 Application cases 322
8.12 Configuring ARP Inspection 322
8.12.1 Overview 322
8.12.2 Configuration 323
8.12.3 Application cases 325
8.13 Configuring DHCP Snooping 325
8.13.1 Overview 325
8.13.2 Configuration 325
8.13.3 Application cases 327
8.14 Configuring IP source guard 327
8.14.1 Overview 327
8.14.2 Configuration 328
8.14.3 Application cases 329
8.15 Configuring Private-vlan 329
8.15.1 Overview 329
8.15.2 Configuration 330
8.15.3 Application cases 331
8.16 Configuring AAA 331
8.16.1 Overview 331
8.16.2 Configuration 332
8.16.3 Application cases 333
8.17 Configuring TACACS+ 336
8.17.1 Overview 336
8.17.2 Configuration 336
8.17.3 Application cases 338
8.18 Configuring Port Isolate 338
8.18.1 Overview 338
8.18.2 Configuration 339
8.18.3 Application cases 340
8.19 Configuring DDoS 340
8.19.1 Overview 340
8.19.2 Configuration 340
8.19.3 Application cases 342

8.20 Configuring Key Chain 342
8.20.1 Overview 342
8.20.2 Configuration 342
8.20.3 Application cases 342
8.21 Configuring Port-Block 343
8.21.1 Overview 343
8.21.2 Configuration 343
8.21.3 Application cases 343
8.22 Configuring Global ACL 343
8.22.1 Overview 343
8.22.2 Configuration Examples 343
8.23 Configuring MACsec 344
8.23.1 Overview 344
8.23.2 Configuration Examples 346
9 Device Management Configuration Guide 350
9.1 Configuring STM 350
9.1.1 Overview 350
9.1.2 Configuration 350
9.1.3 Application cases 352
9.2 Configuring syslog 352
9.2.1 Overview 352
9.2.2 Configuration 353
9.2.3 Application cases 355
9.3 Configuring mirror 355
9.3.1 Overview 355
9.3.2 Configuration 357
9.3.3 Application cases 366
9.4 Configuring Device Management 366
9.4.1 Overview 366
9.4.2 Configuration 367
9.5 Configuring Bootrom 371
9.5.1 Overview 371
9.5.2 Configuration 371
9.6 Upgrade firmware 374
9.6.1 Overview 374
9.6.2 Configuration 374
9.7 Configuring SmartConfig 375
9.7.1 Overview 375
9.7.2 Configuration 376
9.8 Reboot Logs 377
9.8.1 Overview 377
9.8.2 Configuration 378
9.9 Upload config 378

9.9.1 Overview 378
9.9.2 Configuration 379
10 Network Management Configuration Guide 381
10.1 Configuring Network Diagnosis 381
10.1.1 Overview 381
10.1.2 Configuration 381
10.1.3 Application cases 381
10.2 Configuring NTP 382
10.2.1 Overview 382
10.2.2 Configuration 382
10.2.3 Application cases 385
10.3 ConfiguringPTP 386
10.3.1 Overview 386
10.3.2 Configuration Examples 389
10.4 Configuring Phy Loopback 393
10.4.1 Overview 393
10.4.2 Configuration 393
10.4.3 Application cases 395
10.5 Configuring L2 ping 395
10.5.1 Overview 395
10.5.2 Configuration 396
10.6 Configuring RMON 396
10.6.1 Overview 396
10.6.2 Configuration 397
10.7 Configuring SNMP 398
10.7.1 Overview 398
10.7.2 Configuration 399
10.8 Configuring SFLOW 402
10.8.1 Overview 402
10.8.2 Configuration 402
10.8.3 Application cases 404
10.9 Configuring LLDP 404
10.9.1 Overview 404
10.9.2 Configuration 404
10.10 Configuring IPFIX 406
10.10.1 Overview 406
10.10.2 Configuration 408
10.10.3 Application cases 411
10.11 ConfiguringHardware Monitor 411
10.11.1 Overview 411
11 Traffic Managemant Configuration Guide 420
11.1 Configuring QoS 420
11.1.1 Overview 420

11.1.2 Configuration 425
11.1.3 Configuration for Queue 428
11.1.4 Configuration for Port shaping & port policing 432
11.1.5 Application cases 434
12 IPv6 Service Configuration 438
12.1 Configuring IPv6 over IPv4 Tunnel 438
12.1.1 Overview 438
12.1.2 Configuration 440
12.1.3 Application cases 451
12.2 Configuring ND 451
12.2.1 Overview 451
12.2.2 Configuration 452
12.3 Configuring DHCPv6 Relay 453
12.3.1 Overview 453
12.3.2 Configuration 453
12.3.3 Application cases 455
12.4 Configuration DHCPv6 Server 455
12.4.1 Overview 455
12.4.2 Configuration 456
13 IPv6 Security Configuration Guide 458
13.1 DHCPv6 Snooping Configuration 458
13.1.1 Overview 458
13.1.2 Configuration 458
14 IPv6 Routing Configuration 461
14.1 Configuring IPv6 Unicast-Routing 461
14.1.1 Overview 461
14.1.2 Configuration 461
14.1.3 Application cases 463
14.2 Configuring OSPFv3 463
14.2.1 Overview 463
14.2.2 Configuration 464
14.2.3 Application cases 489
14.3 Configuring RIPng 489
14.3.1 Overview 489
14.3.2 Configuration 490
14.3.3 Application cases 503
14.4 Configuring Ipv6 Prefix-list 503
14.4.1 Overview 503
14.4.2 Configuration 503
14.4.3 Application cases 506
14.5 ConfiguringIPv6 PBR 506
14.5.1 Overview 506
14.5.2 Configuration Examples 507

15 IPv6 Multicast Configuration Guide 509
15.1 Configuring IPv6 Multicast-Routing 509
15.1.1 Overview 509
15.1.2 Configuration 509
15.1.3 Application cases 509
15.2 Configuring MLD 509
15.2.1 Overview 509
15.2.2 Configuration 510
15.3 Configuring PIMv6-SM 512
15.3.1 Overview 512
15.3.2 Configuration 513
15.4 Configuring PIMv6-DM 520
15.4.1 Overview 520
15.4.2 Configuration 521
15.5 Configuring MLD Snooping 523
15.5.1 Overview 523
15.5.2 Configuration 523
15.5.3 Application cases 529
15.6 Configuring MVR6 529
15.6.1 Overview 529
15.6.2 Configuration 529
16 VPN Configuration Guide 532
16.1 Configuring VPN 532
16.1.1 Overview 532
16.1.2 Configuration 532
16.2 Configuring IPv4 GRE Tunnel 533
16.2.1 Overview 533
16.2.2 Configuration 534
16.2.3 Application cases 536
16.3 Configuring IPv4 ERSPAN Tunnel 536
16.3.1 Overview 536
16.3.2 Configuration Examples 536
17 Reliability Configuration Guide 540
17.1 Configuring BHM 540
17.1.1 Overview 540
17.1.2 Configuration 540
17.1.3 Application cases 540
17.2 Configuring EFM OAM 540
17.2.1 Overview 540
17.2.2 Configuration 541
17.3 Configuring CFM 545
17.3.1 Overview 545
17.3.2 Configuration 546

17.4 Configuring CPU Traffic 562
17.4.1 Overview 562
17.4.2 Configuration 565
17.5 Configuring UDLD 567
17.5.1 Overview 567
17.5.2 Configuration 568
17.6 Configuring ERPS 569
17.6.1 Overview 569
17.6.2 Configuration 576
17.7 Configuring Smart Link 594
17.7.1 Overview 594
17.7.2 Configuration 594
17.7.3 Application cases 598
17.8 Configuring Multi-Link 598
17.8.1 Overview 598
17.8.2 Configuration 599
17.8.3 Application cases 601
17.9 Configuring Monitor Link 606
17.9.1 Overview 606
17.9.2 Configuration 607
17.9.3 Application cases 608
17.10 Configuring VRRP 608
17.10.1 Overview 608
17.10.2 Configuration 609
17.10.3 Application cases 626
17.11 Configuring Track 626
17.11.1 Overview 626
17.11.2 Configuration 626
17.11.3 Application cases 640
17.12 Configuring IP BFD 640
17.12.1 Overview 640
17.12.2 Configuration 641
17.13 Configuring VARP 648
17.13.1 Overview 648
17.13.2 Configuration 648
17.13.3 Application cases 649
17.14 Configuring UDP Helper 649
17.14.1 Overview 649
17.14.2 Configuration Examples 650
18 Network Virtualization Configuration Guide 652
18.1 Configuring VXLAN 652
18.1.1 Overview 652
18.1.2 Configuration 658

18.1.3 Deployment Suggestion 684
18.2 Configuring NVGRE 692
18.2.1 Overview 692
18.2.2 Configuration Examples 692
18.3 Configuring GENEVE 697
18.3.1 Overview 697
18.3.2 Configuration 698
18.4 Configuring Overlay 703
18.4.1 Overview 703
18.4.2 Configuration 703
18.5 Configuring EVPN 709
18.5.1 Overview 709
18.5.2 Application Scenarios 719
18.5.3 Configuration Examples 721
18.5.4 Deployment Suggestion 760
19 Intelligent Lossless Network Configuration Guide 773
19.1 Configuring Prioprity-based Flow Control 773
19.1.1 Overview 773
19.1.2 Configuration 773
19.2 PFC deadlock detection configuration 775
19.3 Configuring EFD 776
19.3.1 Overview 776
19.3.2 Configuration 776
19.4 ConfiguringECN 778
19.4.1 Overview 778
19.4.2 Configuration Examples 779
20 Stacking Configuration Guide 781
20.1 Overview 781
20.1.1 Background 781
20.1.2 Technical Advantages 781
20.2 Principle Description 782
20.2.1 Basic Concepts 782
20.2.2 Stack Formation 783
20.2.3 Stack Management 785
20.2.4 Stack Maintenance 785
20.2.5 Stack splitting and Dual-Active Detection 786
20.2.6 Principle of packet forwarding in stacking 788
20.3 Configuration Notes 789
20.3.1 Stacking Device Requirements 789
20.3.2 Device Version and License Requirements 789
20.3.3 Stack Port and Cable Requirements 790
20.3.4 Stack Feature Dependencies and Limitations 790
20.4 Stack Configuration Example 792

20.4.1 Example for Stacking of two switches(Linear topology) 792
20.4.2 Example for Stacking of multiple devices(Ring topology) 794
20.4.3 Example for Stacking of Dual-Active Detection 797
21 MPLS Configuration Guide 801
21.1 Configuring LDP 801
21.1.1 Overview 801
21.1.2 Configuration Examples 801
21.2 Configuring MPLS 803
21.2.1 Overview 803
21.2.2 Configuration Examples 803
21.3 Configuring VPLS 805
21.3.1 Overview 805
21.3.2 Configuration Examples 806
21.4 Configuring VPWS 818
21.4.1 Overview 818
21.4.2 Configuration Examples 818
21.5 Configuring MPLS QoS 823
21.5.1 Overview 823
21.5.2 Configuration Examples 825
21.6 Configuring L3VPN 832
21.6.1 Overview 832
21.6.2 Configuration Examples 832

List of Figures

Figure 2-1 SSH system application 34
Figure 3-1 Interface Name 43
Figure 3-2 Mac address aging 53
Figure 3-3 Static mac address table 54
Figure 3-4 Static multicast mac address table 55
Figure 3-5 mac address filter 56
Figure 3-6 mac address filter with VLAN 57
Figure 3-7 Tagged Frame 58
Figure 3-8 Trunk link 58
Figure 3-9 Access link 59
Figure 3-10 Access link 59
Figure 3-11 Trunk link 60
Figure 3-12 vlan classification 63
Figure 3-13 QinQ Tunnel 65
Figure 3-14 vlan mapping 66
Figure 3-15 QinQ Tunnel 67
Figure 3-16 QinQ Tunnel 68
Figure 3-17 QinQ Tunnel 69
Figure 3-18 LACPDU Packet 73
Figure 3-19 Link Aggregation 75
Figure 3-20 Link Redundancy 76
Figure 3-21 Load balance 76
Figure 3-22 Weighted link aggregation 77
Figure 3-23 Master-slave mode link aggregation 77
Figure 3-24 Static Agg 80
Figure 3-25 Dynamic LACP 81
Figure 3-26 Static LACP 83
Figure 3-27 weighted link aggregation 85
Figure 3-28 Master-slave mode link aggregation 87
Figure 3-29 Aggregation application 90
Figure 3-30 Flow control 93
Figure 3-31 Percentage Storm Control 95
Figure 3-32 PPS Storm Control 95
Figure 3-33 L2 protocol tunnel 99
Figure 3-34 MSTP 102

Figure 3-35 MLAG topology sketch 107
Figure 3-36 MLAG traffic forwarding sketch 108
Figure 3-37 Protection switching to unicast messages from MLAG 109
Figure 3-38 MLAG Protection of flood traffic 109
Figure 3-39 Peer-link failure protection 110
Figure 3-40 MLAG 111
Figure 3-41 MLAG+RSTP 114
Figure 3-42 MLAG 119
Figure 3-43 Network gateway diagram of MLAG Layer 3 - Single address 123
Figure 3-44 Network gateway diagram of MLAG Layer 3 - multiple addresses) 128
Figure 3-45 MLAG orphan port 133
Figure 3-46 Network diagram of MLAG domain cascading 137
Figure 4-1 arp 162
Figure 4-2 arp proxy 164
Figure 4-3 local arp proxy 167
Figure 4-4 ARP host-route 170
Figure 4-5 dhcp client 172
Figure 4-6 DHCP relay 173
Figure 4-7 DHCP server 176
Figure 4-8 DHCP relay 178
Figure 4-9 DNS 180
Figure 5-1 ip unicast routing 182
Figure 5-2 enable rip 185
Figure 5-3 rip version 187
Figure 5-4 rip metric 190
Figure 5-5 rip distance 192
Figure 5-6 rip redistribute 194
Figure 5-7 rip split-horizon 196
Figure 5-8 rip filter list 199
Figure 5-9 rip authentication 201
Figure 5-10 rip authentication 203
Figure 5-11 ospf 207
Figure 5-12 ospf priority 209
Figure 5-13 ospf area 212
Figure 5-14 ospf redistribute 215
Figure 5-15 ospf cost 221
Figure 5-16 ospf authentication 224

Figure 5-17 pbr 235
Figure 5-18 pbr 236
Figure 5-19 EBGP 239
Figure 5-20 IBGP 242
Figure 5-21 RIPng 245
Figure 6-1 Pim sm 254
Figure 6-2 bsr 258
Figure 6-3 Pim dm 261
Figure 6-4 mvr 269
Figure 8-1 Port Security 282
Figure 8-2 ACL Rule description 287
Figure 8-3 Flow chart for ACL match-any mode 288
Figure 8-4 Flow chart for ACL match-all mode 288
Figure 8-5 acl 292
Figure 8-6 Enterprise departments network 294
Figure 8-7 Enterprise departments network 298
Figure 8-8 ACL flow Redirect 302
Figure 8-9 extern acl 304
Figure 8-10 ipv6 acl 306
Figure 8-11 copp_acl 310
Figure 8-12 dot1x 312
Figure 8-13 Select “Setting-> System” 315
Figure 8-14 Configure the shared-key, authorization port and account port 315
Figure 8-15 Add user name and password on the server 316
Figure 8-16 Guest vlan: before authenticated 317
Figure 8-17 Guest vlan: after authenticated 317
Figure 8-18 arp inspection 323
Figure 8-19 dhcp snooping 325
Figure 8-20 ip source guard 328
Figure 8-21 private vlan 330
Figure 8-22 private vlan 332
Figure 8-23 Telnet connecting test 333
Figure 8-24 Set IP address for PC 334
Figure 8-25 Connectivity test 334
Figure 8-26 WinRadius 335
Figure 8-27 WinRadius 335
Figure 8-28 Add user and password 335

Figure 8-29 Connectivity test 336
Figure 8-30 TACACS+ 336
Figure 8-31 Telnet connecting test 337
Figure 8-32 Connectivity test 338
Figure 8-33 Port Isolate 339
Figure 8-34 Topology for DDoS test 340
Figure 9-1 syslog server 353
Figure 9-2 syslog on server 355
Figure 9-3 Mirror 356
Figure 9-4 port Mirror 358
Figure 9-5 Multi-destination Mirror 360
Figure 9-6 Remote Mirror 362
Figure 9-7 Mirror to cpu 364
Figure 9-8 smart config 376
Figure 9-9 update configuration 379
Figure 10-1 NTP 382
Figure 10-2 PTP packets 388
Figure 10-3 PTP Timing Domain 389
Figure 10-4 external phy topology 394
Figure 10-5 Internal phy topology 394
Figure 10-6 Port level topology 395
Figure 10-7 ping a switch port 396
Figure 10-8 rmon 397
Figure 10-9 snmp 399
Figure 10-10 sflow 402
Figure 10-11 lldp 404
Figure 10-12 IPFIX basic Topology 407
Figure 10-13 IPFIX Packet 408
Figure 10-14 Buffer Monitor Principle description 412
Figure 10-15 Latency Monitor Principle description 412
Figure 10-16 Topology of notification function 417
Figure 12-1 IPv6 over IPv4 Tunnel 438
Figure 12-2 IPv6 over IPv4 Tunnel 439
Figure 12-3 ISATAP Tunnel 440
Figure 12-4 Manual Tunnel 440
Figure 12-5 6to4 tunnel 443
Figure 12-6 6to4 relay 446

Figure 12-7 ISATAP tunnel 449
Figure 12-8 NDP 452
Figure 12-9 DHCP Relay 453
Figure 12-10 DHCPv6 Server 456
Figure 13-1 DHCPv6 Snooping 458
Figure 14-1 ipv6 unicast routing 461
Figure 14-2 OSPFv3 464
Figure 14-3 OSPFv3 priority 467
Figure 14-4 OSPFv3 area 470
Figure 14-5 OSPFv3 Redistribute 475
Figure 14-6 OSPFv3 Cost 481
Figure 14-7 RIPng 490
Figure 14-8 RIPng Metric 493
Figure 14-9 RIPng Distance 495
Figure 14-10 RIPng redistribute 496
Figure 14-11 RIPng Split-horizon 499
Figure 14-12 RIPng Route Distribute Filters 502
Figure 14-13 Typical networking diagram of IPv6 PBR 507
Figure 15-1 PIMv6 Sparse-mode 514
Figure 15-2 BSR 519
Figure 15-3 PIMv6 dense-mode 521
Figure 15-4 MVR6 529
Figure 16-1 IPv4 gre over IPv4 533
Figure 16-2 IPv4 gre Tunnel 534
Figure 16-3 IPv4 ERSPAN Tunnel Single-Destination Address Mode 537
Figure 16-4 IPv4 ERSPAN Tunnel Multi-Destination Address Mode 538
Figure 17-1 EFM 541
Figure 17-2 EFM 543
Figure 17-3 EFM 544
Figure 17-4 EFM 545
Figure 17-5 CFM 547
Figure 17-6 CFM 554
Figure 17-7 CFM CSF 555
Figure 17-8 CFM 558
Figure 17-9 CFM 560
Figure 17-10 CFM 561
Figure 17-11 UDLD 568

Figure 17-12 ERPS diagram 570
Figure 17-13 ERPS Ring type 571
Figure 17-14 Flow chart of polling mechanism 572
Figure 17-15 Status of Master Node 572
Figure 17-16 Diagram of master node's status migration 572
Figure 17-17 Diagram of transit node's status migration 572
Figure 17-18 Channel detection mechanism of sub ring protocol 573
Figure 17-19 Edge node sends edge hello message 573
Figure 17-20 Sub ring protocol link diagram 574
Figure 17-21 Edge node responses major fault block edge port 574
Figure 17-22 Loop in multi sub ring 575
Figure 17-23 Master node of sub ring get into failed status 575
Figure 17-24 Edge node of sub ring opens edge port 575
Figure 17-25 Normal ERPS networking diagram 575
Figure 17-26 Multiple instances of network forwarding 576
Figure 17-27 ERPS Single-Ring 576
Figure 17-28 ERPS Intersecting-Ring 580
Figure 17-29 Tangency ring networking diagram 585
Figure 17-30 Multi sub rings networking diagram 588
Figure 17-31 Multiple instances networking diagram 591
Figure 17-32 Smart-Link Typical Topology 595
Figure 17-33 Multi-Link Typical Topology 599
Figure 17-34 Multilink-enhance Typical Topology 602
Figure 17-35 monitor link 607
Figure 17-36 Without VRRP 609
Figure 17-37 With VRRP 609
Figure 17-38 VRRP with one virtual router 610
Figure 17-39 VRRP with two virtual router 612
Figure 17-40 VRRP Circuit Failover 615
Figure 17-41 IPv6 VRRP with one virtual router 618
Figure 17-42 IPv6 VRRP with two virtual router 620
Figure 17-43 IPv6 VRRP Circuit Failover 623
Figure 17-44 IP SLA 627
Figure 17-45 IP SLA 628
Figure 17-46 IP SLA 630
Figure 17-47 Track interface 632
Figure 17-48 Track ip sla 633

Figure 17-49 Track ip sla 634
Figure 17-50 Track bfd 636
Figure 17-51 VRRP Track 637
Figure 17-52 Static Route Track 639
Figure 17-53 BFD single hop 641
Figure 17-54 BFD multi hop 644
Figure 17-55 BFD WTR 646
Figure 17-56 VARP with MALG 648
Figure 17-57 UDP-Helper configuration 650
Figure 18-1 The packet format of VXLAN 653
Figure 18-2 The process of broadcast packets forwarding 654
Figure 18-3 The process of unicast packets forwarding 655
Figure 18-4 Diagram of concentrated gateway 656
Figure 18-5 Diagram of distributed gateway 657
Figure 18-6 Vxlan 658
Figure 18-7 Vxlan Routing 661
Figure 18-8 EBGP_EVPN 664
Figure 18-9 IBGP_EVPN 669
Figure 18-10 VXLAN network communication 676
Figure 18-11 Concentrated VXLAN gateway networking diagram 679
Figure 18-12 Active-active connects to VXLAN and distributed gateway 684
Figure 18-13 NVGRE 692
Figure 18-14 NVGRE distributed route 694
Figure 18-15 GENEVE 698
Figure 18-16 GENEVE 700
Figure 18-17 Overlay multiple source ip 704
Figure 18-18 OVERLAY without Horizon Split 707
Figure 18-19 BGP Update message format 711
Figure 18-20 Withdraw Routes message format 711
Figure 18-21 EXTEMDED_COMMUNITIES message format 712
Figure 18-22 Community Transitive Two-Octet AS Route target message format 712
Figure 18-23 Community Transitive Opaque Encapsulation message format 712
Figure 18-24 Type 2 routing's message format 712
Figure 18-25 Type 3 routing's prefix and PMSI attributes 713
Figure 18-26 Type 5 routing's message format 714
Figure 18-27 Application of Type 3 routes to automatic discovery of VXLAN tunnelsl 715
Figure 18-28 VXLAN Tunnel 715

Figure 18-29 Application of Type 2 routes to host MAC address notification 716
Figure 18-30 MAC table synchronization 716
Figure 18-31 Application of Type 2 routes to host ARP notification 717
Figure 18-32 Host Route synchronization 717
Figure 18-33 Application of Type 5 routes in host IP address advertisement 718
Figure 18-34 Segment route synchronization 719
Figure 18-35 Route-reflector-client 719
Figure 18-36 Retain route target 720
Figure 18-37 allowas-in 720
Figure 18-38 attribute-unchanged 721
Figure 18-39 BGP EVPN concentrated gateway 721
Figure 18-40 BGP EVPN concentrated gateway 727
Figure 18-41 BGP EVPN Import External Route 734
Figure 18-42 GP EVPN segmented VXLAN Tunnel 739
Figure 18-43 MLAG TOR in EVPN Network 748
Figure 18-44 Concentrated gateway 760
Figure 18-45 Layer 2 packet control plane Creating tunneling 761
Figure 18-46 Schematic diagram of learning MAC addresses on the Layer 2 packet control plane 762
Figure 18-47 Forwarding plane process 763
Figure 18-48 BUM 763
Figure 18-49 Unicast 764
Figure 18-50 Layer 3 networking of the concentrated gateway 765
Figure 18-51 concentrated gateway ARP learning 765
Figure 18-52 Route learning in the concentrated gateway segment 766
Figure 18-53 Topology of Layer 3 packet forwarding on the concentrated gateway 767
Figure 18-54 Process of Layer 3 packet forwarding on the concentrated gateway 767
Figure 18-55 Distributed gateway 768
Figure 18-56 ARP learning of the distributed gateway 769
Figure 18-57 Route learning on the distributed gateway 769
Figure 18-58 Layer 3 packet forwarding on the distributed gateway 770
Figure 18-59 Extended community attributes for Mac mobility 771
Figure 18-60 VM Migration 771
Figure 18-61 Dual host active 772
Figure 19-1 Priority-based Flow Control 773
Figure 19-2 EFD 776
Figure 19-3 ToS Field 778
Figure 19-4 ECN mechanism 779

Figure 20-1 Stack Virtual Device 781
Figure 20-2 Schematic diagram of the physical connection of the stack 784
Figure 20-3 Box-type equipment stacking effect diagram 785
Figure 20-4 Schematic diagram of stacking splitting 787
Figure 20-5 Schematic diagram of DAD business port direct connection detection 787
Figure 20-6 Forwarding unicast messages within a device 788
Figure 20-7 Cross-device unicast message forwarding 789
Figure 20-8 Multicast message forwarding 789
Figure 20-9 Stacking map 792
Figure 20-10 Stacking map 795
Figure 20-11 dual-active detection map 798
Figure 21-1 LSP map 801
Figure 21-2 MPLS LSP model 803
Figure 21-3 VPLS model 806
Figure 21-4 Topology of vpws configuration 819
Figure 21-5 Uniform model 824
Figure 21-6 Pipe model 824
Figure 21-7 Short pipe model 825
Figure 21-8 MPLS QoS LSP model 825
Figure 21-9 L3VPN model 832

1 Preface

1.1 Declaration

This document updates at irregular intervals because of product upgrade or other reason.

This document is for your reference only.

1.2 Suggestion feedback

If you have any questions when using our product and reading this document, please contact us

1.3 Audience

This document is for the following audiences:

- System maintenance engineers
- Debugging and testing engineers
- Network monitoring engineers
- Field maintenance engineers

2 Basic Configuration Guide

2.1 Configuring System Management

2.1.1 Overview

Function Introduction

Banner function is used for configuring messages on the devices. User can specify any messages to notify other users. Improper operations might cause critical situation such as service interrupt, in this case, a notification in advance is necessary. (E.g. to notify users "Don't reboot")

Three types of messages are supported by now:

- MOTD(message-of-the-day). Messages will display on the terminal when user connect to the device.
- login banner. Messages will display on the terminal when user login to the device. "Login mode" is required for displaying this message. Please reference the section of "Configuring User Management".
- exec banner. Messages will display on the terminal when user enter the EXEC mode.

Principle Description

This function displays notification on the terminal to reduce misoperation.

2.1.2 Configuration

Configuring a MOTD Login Banner

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create the notification

User can create a notification (one line or multiple lines) to display on all connected terminals. In the following example, the delimiting character is #. All characters between two delimiting characters will display on the terminals when user connect the device.

The message length is at most 99 lines with 1023 character in each line.

```
Switch(config)# banner motd # This is a switch #
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

step 4 Validation

Use the following command to display the configuration:

```
switch# show running
banner motd ^C
 This is a switch
^C
```

Configuring a Login Banner

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create the notification

User can create a notification (one line or multiple lines) to display on all connected terminals. "Login mode" is required for displaying this message. Please reference the section of "Configuring User Management".

In the following example, the delimiting character is #. All characters between two delimiting characters will display on the terminals when user connect the device.

The message length is at most 99 lines with 1023 character in each line.

```
Switch(config)# banner login # admin login #
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

step 4 Validation

Use the following command to display the configuration

```
switch# show running
banner login ^C
 admin login
^C
```

Configuring an Exec Banner

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create the notification

User can create a notification (one line or multiple lines) to display on all connected terminals. In the following example, the delimiting character is #. All characters between two delimiting characters will display on the terminals when user enter the EXEC mode.

The message length is at most 99 lines with 1023 character in each line.

```
Switch(config)# banner exec # do not reboot! #
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

step 4 Validation

Use the following command to display the configuration:

```
switch# show running
banner exec ^C
 do not reboot!
^C
```

2.1.3 Application cases

Case 1: mark the usage of the device

Set the MOTD message as "This is a switch of some area/department", user can see this message when connect to the device. If the user needs to operate a switch of another department, he can realize that he connected to a wrong device and stop misoperation.

Configuration steps

```
Switch# configure terminal
Switch(config)# banner motd # This is a switch of IT DEPARTMENT ！ ！ ！ #
Switch(config)# exit
```

Configuration files

```
switch# show running
banner motd ^C
```

```
This is a switch of IT DEPARTMENT ！ ！ ！
^C
```

2.2 Configuring User Management

2.2.1 Overview

Function Introduction

User management increases the security of the system by keeping the unauthorized users from guessing the password. The user is limited to a specific number of attempts to successfully log in to the switch.

There are three load modes in the switch.

- In "no login" mode, anyone can load the switch without authentication.
- In "login" mode, there is only one default user.
- In "login local" mode, if you want to load the switch you need to have a user account. Local user authentication uses local user accounts and passwords that you create to validate the login attempts of local users. Each switch has a maximum of 32 local user accounts. Before you can enable local user authentication, you must define at least one local user account. You can set up local user accounts by creating a unique username and password combination for each local user. Each username must be fewer than 32 characters. You can configure each local user account with a privilege level; the valid privilege levels are 1 or 4. Once a local user is logged in, only the commands those are available for that privilege level can be displayed.

There is only one user can enter the configure mode at the same time.

Principle Description

N/A

2.2.2 Configuration

Configuring the user management in login local mode

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 et username and password

```
Switch(config)# username testname privilege 4 password 123abc<>
```

step 3 Enter the configure mode and set user management mode

```
Switch(config)# line vty 0 7
Switch(config-line)# login local
Switch(config-line)# exit
```

step 4 Exit the configure mode

```
Switch(config)# exit
```

step 5 Validation

After the above setting, login the switch will need a username and password, and user can login with the username and password created before. This is a sample output of the login prompt.

```
Username:
```

After the input the username, a password is required.

```
Username: testname
Password:
```

Authentication succeed:

```
Password:

Switch#
```

Configuring the user management in login mode

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the configure mode and set password

```
Switch(config)# line vty 0 7
Switch(config-line)# line-password abc
Switch(config-line)# login
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

step 4 Validation

After the above setting, login the switch will need the line password, and user can login with the password created before. This is a sample output of the login prompt.

```
Password:
```

Configuring Password recovery procedure

If the password is forgotten unfortunately, it can be recovered by following steps.

Step 1 Power on the system. Boot loader will start to run. The follow information will be printed on Console.

```
CPU:  MPC8247 (HiP7 Rev 14, Mask 1.0 1K50M) at 350 MHz
Board: 8247 (PCI Agent Mode)
I2C:  ready
DRAM:  256 MB
In:  serial
Out:  serial
Err:  serial
Net:  FCC1 ETHERNET, FCC2 ETHERNET [PRIME]
Press ctrl+b to stop autoboot: 3
```

Step 2 Press ctrl+b. stop autoboot.

```
Bootrom#
```

Step 3 Under boot loader interface, use the following instructions.

```
Bootrom# boot_flash_nopass
Bootrom# Do you want to  revert  to the default config file ? [Y|N|E]:
```

Please remember your username and password.

Recovering the password may lead configuration lost or service interrupted; we strongly recommend that user should remember the username and password.

2.2.3 Application cases

N/A

2.3 Configuring FTP

2.3.1 Overview

Function Introduction

You can download a switch configuration file from an FTP server or upload the file from the switch to an FTP server. You download a switch configuration file from a server to upgrade the switch configuration. You can overwrite the current startup configuration file with the new one. You upload a switch configuration file to a server for backup purposes. You can use this uploaded configuration for future downloads to the switch or another switch of the same type.

Principle Description

N/A

2.3.2 Configuration

You can copy configurations files to or from an FTP server. The FTP protocol requires a client to send a remote username and password on each FTP request to a server.

Before you begin downloading or uploading a configuration file by using FTP, do these tasks:

- Ensure that the switch has a route to the FTP server. The switch and the FTP server must be in the same network if you do not have a router to route traffic between subnets. Check connectivity to the FTP server by using the ping command.
- If you are accessing the switch through the console or a Telnet session and you do not have a valid username, make sure that the current FTP username is the one that you want to use for the FTP download.
- When you upload a configuration file to the FTP server, it must be properly configured to accept the write request from the user on the switch.

For more information, see the documentation for your FTP server.

Downloading a configuration file by using FTP in IPv4 network

step 1 copy the configuration file

```
Switch# copy mgmt-if ftp://test:test@10.10.10.163/ startup-config.conf  flash:/startup-config.conf
```

step 2 Validation

Use the following command to display the configuration

```
Switch# show startup-config
```

Uploading a configuration file by using FTP in IPv4 network

step 1 copy the configuration file

```
Switch# copy flash:/startup-config.conf mgmt-if ftp://test:test@10.10.10.163/startup-config.conf
```

Downloading a configuration file by using FTP in IPv6 network

Username and password settings are same as IPv4 network.

step 1 copy the configuration file

```
Switch# copy ftp://root: root@2001:1000::2/startup-config.conf  flash:/startup-config.conf
```

Uploading a configuration file by using FTP in IPv6 network

Username and password settings are same as IPv4 network.

step 1 copy the configuration file

```
Switch# copy flash:/startup-config.conf mgmt-if ftp://root:root@2001:1000::2 startup-config.conf
```

2.3.3 Application cases

N/A

2.4 Configuring FTP SERVER

2.4.1 Overview

Function Introduction

You can download a switch configuration file from an FTP server or upload the file from the switch to an FTP server. You download a switch configuration file from a server to upgrade the switch configuration. You can overwrite the current startup configuration file with the new one. You upload a switch configuration file to a server for backup purposes. You can use this uploaded configuration for future downloads to the switch or another switch of the same type.

Principle Description

N/A

2.4.2 Configuration

You can copy configurations files to or from an FTP server. The FTP protocol requires a client to send a remote username and password on each FTP request to a server.

Before you begin downloading or uploading a configuration file by using FTP, do these tasks:

- Ensure that the switch has a route to the FTP server. The switch and the FTP server must be in the same network if you do not have a router to route traffic between subnets. Check connectivity to the FTP server by using the ping command.
- If you are accessing the switch through the console or a Telnet session and you do not have a valid username, make sure that the current FTP username is the one that you want to use for the FTP download.
- When you upload a configuration file to the FTP server, it must be properly configured to accept the write request from the user on the switch.

For more information, see the documentation for your FTP server.

configuration of FTP server

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable FTP server on management interface

```
Switch(config)# ftp server mgmt-if enable
```

step 3 Config switch system users

Users should config password and the privilege should be 4

```
Switch(config)# username admin privilege 4 password admin
```

step 4 Validation

Clent connect to FTP server， enter the username and password。The IP address of server management interface is 10.10.10.10

```
Switch# ftp mgmt-if 10.10.10.10
Connected to 10.10.10.10.
220---------- Welcome to FTP-SERVER ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 06:41. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (10.10.10.10:LOGIN): admin
```

```
331 User admin OK. Password required
Password:
230 OK. Current directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
```

Other optional configuration

Config FTP server port，default is 21

```
Switch# ftp server port 10000
```

Config FTP server time-out，default is 15min

```
Switch# ftp server time-out 5
```

Config FTP server VRF or IP address inband. config source IP is 1.1.1.1，config source vrf is test，config source vrf is test and source IP is 2.2.2.2.

```
Switch# ftp server source address 1.1.1.1
Switch# ftp server source address vrf test 0.0.0.0
Switch# ftp server source address vrf test 2.2.2.2
```

2.4.3 Application cases

N/A

2.5 Configuring TFTP

2.5.1 Overview

Function Introduction

You can download a switch configuration file from a TFTP server or upload the file from the switch to a TFTP server. You download a switch configuration file from a server to upgrade the switch configuration. You can overwrite the current file with the new one. You upload a switch configuration file to a server for backup purposes; this uploaded file can be used for future downloads to the same or another switch of the same type.

Principle Description

N/A

2.5.2 Configuration

Before you begin downloading or uploading a configuration file by using TFTP, do these tasks:

Ensure that the workstation acting as the TFTP server is properly configured.

Ensure that the switch has a route to the TFTP server. The switch and the TFTP server must be in the same network if you do not have a router to route traffic between subnets. Check connectivity to the TFTP server by using the ping command.

Ensure that the configuration to be downloaded is in the correct directory on the TFTP server.

For download operations, ensure that the permissions on the file are set correctly.

During upload operations, if you are overwriting an existing file (including an empty file, if you had to create one) on the server, ensure that the permissions on the file are set correctly.

Downloading a configuration file by using TFTP in IPv4 network

```
Switch# copy mgmt-if tftp://10.10.10.163/startup-config.conf flash:/startup-config.conf
```

Uploading a configuration file by using TFTP in IPv4 network

```
Switch# copy flash:/startup-config.conf mgmt-if tftp://10.10.10.163/startup-config.conf
```

Downloading a configuration file by using TFTP in IPv6 network

```
Switch# copy mgmt-if tftp://2001:1000::2/startup-config.conf flash:/startup-config.conf
```

Uploading a configuration file by using TFTP in IPv6 network

```
Switch# copy flash:/startup-config.conf mgmt-if tftp://2001:1000::2/startup-config.conf
```

2.5.3 Application cases

N/A

2.6 Configuring SCP

2.6.1 Overview

Function Introduction

SCP， which is short for secure copy, is a part of SSH protocol. It is a remote copy technology which is based on SSH protocol. User can download a switch configuration file from a SCP server or upload the file from the switch to a SCP server. User can download a switch configuration file from a server to upgrade the switch configuration and overwrite the current file with the new one. User can upload a switch configuration file to a server for backup purposes; this uploaded file can be used for future downloads to the same or another switch of the same type.

Principle Description

N/A

2.6.2 Configuration

Before you begin downloading or uploading a configuration file by using SCP, do these tasks:

Ensure that the workstation acting as the SCP server is properly configured.

Ensure that the switch has a route to the SCP server. The switch and the SCP server must be in the same network if you do not have a router to route traffic between subnets. Check connectivity to the SCP server by using the ping command.

Ensure that the configuration to be downloaded is in the correct directory on the SCP server.

For download operations, ensure that the permissions on the file are set correctly.

During upload operations, if you are overwriting an existing file (including an empty file, if you had to create one) on the server, ensure that the permissions on the file are set correctly.

Downloading a configuration file by using SCP in IPv4 network

```
Switch# copy mgmt-if scp://10.10.10.163/startup-config.conf flash:/startup-config.conf
```

Uploading a configuration file by using SCP in IPv4 network

```
Switch# copy flash:/startup-config.conf mgmt-if scp://10.10.10.163/startup-config.conf
```

Downloading a configuration file by using SCP in IPv6 network

```
Switch# copy mgmt-if scp://2001:1000::2/startup-config.conf flash:/startup-config.conf
```

Uploading a configuration file by using SCP in IPv6 network

```
Switch# copy flash:/startup-config.conf mgmt-if scp://2001:1000::2/startup-config.conf
```

2.6.3 Application cases

N/A

2.7 Configuring Telnet

2.7.1 Overview

Function Introduction

Telnet is a network protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communications facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). Telnet was developed in 1969 beginning with RFC 15, extended in RFC 854, and standardized as Internet Engineering Task Force (IETF) Internet Standard STD 8, one of the first Internet standards. Historically, Telnet provided access to a command-line interface (usually, of an operating system) on a remote host. Most network equipment and operating systems with a TCP/IP stack support a Telnet service for remote configuration (including systems based on Windows NT). Because of security issues with Telnet, its use for this purpose has waned in favor of SSH.

Principle Description

N/A

2.7.2 Configuration

Telnet switch with inner port

Example 1 IPv4 Network

```
Switch# telnet 10.10.29.247
Entering character mode
Escape character is '^]'.
Switch #
```

Example 2 IPv6 Network

```
Switch# telnet 2001:1000::71
Entering character mode
Escape character is '^]'.
Switch #
```

Telnet switch with management port

Example 1 IPv4 Network

```
Switch# telnet mgmt-if 10.10.29.247
Entering character mode
Escape character is '^]'.
Switch #
```

Example 2 IPv6 Network

```
Switch# telnet mgmt-if 2001:1000::2
Entering character mode
Escape character is '^]'.
Switch #
```

Configure telnet server

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable Telnet service

```
Switch(config)# service telnet enable
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

2.7.3 Application cases

N/A

2.8 Configuring SSH

2.8.1 Overview

Function Introduction

The Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. SSH supports the Data Encryption Standard (DES) encryption algorithm, the Triple DES (3DES) encryption algorithm, and password-based user authentication. The SSH feature has an SSH server and an SSH integrated client, which are applications that run on the switch. You can use an SSH client to connect to a switch running the SSH server. The SSH server works with the SSH client supported in this release and with SSH clients. The SSH client also works with the SSH server supported in this release and with SSH servers.

Principle Description

N/A

2.8.2 Configuration



Figure 2-1 SSH system application

Create key for SSH

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a key

```
Switch(config)# rsa key a generate
```

step 3 Create a private key named a.pri with key a and save it to flash

```
Switch(config)# rsa key a export url flash:/a.pri private ssh2
```

step 4 Create a private key named a.pub with key a and save it to flash

```
Switch(config)# rsa key a export url flash:/a.pub public ssh2
```

step 5 Exit the configure mode

```
Switch(config)# exit
```

Import the key

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Import the key a.pub we created as importKey

```
Switch(config)# rsa key importKey import url flash:/a.pub public ssh2
```

step 3 Create username and password

```
Switch(config)# username aaa privilege 4 password abc
```

step 4 Assign the key to user aaa

```
Switch(config)# username aaa assign rsa key importKey
```

step 5 Exit the configure mode

```
Switch(config)# exit
```

Use SSH to connect

step 1 Download the a.pri key on SSH client

step 2 Connect to the client

```
[root@test1 tftpboot]# ssh -i a.pri aaa@10.10.39.101
aaa@10.10.39.101's password:
Switch#
```

2.8.3 Application cases

N/A

2.9 Configuring Time&timezone

2.9.1 Overview

Function Introduction

If no other source of time is available, you can manually configure the time and date after the system is restarted. The time remains accurate until the next system restart. We recommend that you use manual configuration only as a last resort. If you have an outside source to which the switch can synchronize, you do not need to manually set the system clock.

Principle Description

N/A

2.9.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configuring time and timezone

```
Switch(config)# clock set datetime 11:30:00 10 26 2013
Switch(config)# clock set summer-time dst date 6 1 2013 02:00:00 10 31 2013 02:00:00 120
```

step 3 Exit the configure mode

```
Switch(config)# exit
```

step 4 Validation

Use the following command to display the information of time and date:

```
Switch# show clock detail
13:31:10 dst Sat Oct 26 2013
Time zone: (GMT + 08:00:00)  beijing
Summer time starts at beijing 02:00:00  06/01/2013
Summer time ends at dst 02:00:00  10/31/2013
Summer time offset:  120 minutes
```

2.9.3 Application cases

N/A

2.10 Configuring License

2.10.1 Overview

Function Introduction

License will control the features on the switch; each switch has its own license to avoid the unauthorized user to use the advanced features. There are totally three kinds of licenses: Enterprise Base, Metro Service, and Metro Advanced. Different license will contain different features. Customer can apply different license to satisfy different requirement. If switch has no license, it can only provide L2 features. Different switch can't share the same license. In order to get the license for the specify switch, first generate the unique device identifier(UDI) for the switch and then send the UDI to vendor to apply the license, at last get the license from vendor and use the license on the switch.

Principle Description

N/A

2.10.2 Configuration

step 1 Create UDI for the device and send it to remote FTP server

```
Switch# generate device identifier mgmt-if ftp://test:test@10.10.25.33/device.udi
```

step 2 Apply license

Send UDI file to vendor, vendor will generate license for customer requirement.

step 3 Use license

Get the license to local from remote FTP server, and reload the system.

```
Switch# copy mgmt-if ftp://test:test@10.10.25.33/device.lic flash:/device.lic
Switch# reload
```



You must reload the switch for the license to take effect.

If the switch has no license, it can only work with L2 features.

If the switch has more than one license, all the features contain by the licenses can take effect

step 4 Validation

Use the following command to display the information of the license:

```
Switch# show licence
Licence files:
```

```
================================================================

flash:/MA.lic:
    [License]
    CreatedTime=Tue Dec 26 04:46:31 EST 2023
    Vendor=Unspecified
    Customer=Unspecified
    Mac=D8:5B:22:0B:BA:84
    LicenseType=MA
    Features=PIM_SM IGMP BGP BFD IPV6 OSPF6 PIM_SM6 MVR6 RIPNG TUNNEL_V6 SSM
 PIM_DM PBR OSPF MLAG ISIS MACSEC BGP4+ VRF MPLS LDP QINQ MVR ERPS MEF ETHOAM EX
TEND_ACL PTP HW_MONITOR IPFIX EFD VPWS VPLS HVPLS TPOAM RSVP OSPF_TE EVPN MPLSSR

Switch#
```

2.11 RPC API Configuration Guide

2.11.1 Overview

Function Introduction

RPC API service allows user to configure and monitor the switch system through Remote Procedure Calls (RPC) from your program.

The service currently supports JSON-RPC over HTTP protocol together with HTTP Basic authentication.

Principle Description

RPC API service uses standard JSON-RPC over HTTP protocol to communicate the switch and your program. User may issue switch CLI commands through JSON-RPC method: 'executeCmds'. By default, the CLI mode is in privileged EXEC mode (#).

User could send JSON-RPC request via an HTTP POST request to URL: http://:/command-api. The detailed JSON-RPC request and response are show below:

JSON-RPC Request

```
{
  "params":[                          Parameters for command
    {
      "format":"text",                 Expected response format, can be 'text' or 'json',
the default format is 'text'
      "version":1,                     The API version
      "cmds":[                        List of CLI commands
        "show run",                    CLI command 1
        "config t",                   CLI command 2
        "vlan database",                CLI command 3
        "vlan 1-8",                   CLI command 4
        "interface eth-0-1",            CLI command 5
        "switchport mode trunk",          CLI command 6
        "switchport trunk allowed vlan add 2",  CLI command 7
        "shutdown",                     CLI command 8
        "end",                        CLI command 9
        "show interface switchport"         CLI command 10
      ]
    }
  ],
  "jsonrpc":"2.0",                      JSON RPC protocol version. Always 2.0.
  "method":"executeCmds",                   Method to run the switch CLI commands
```

```
  "id":"70853aff-af77-420e-8f3c-fa9430733a19"       JSON RPC unique identifier
}
```

JSON-RPC Response

```
{
  "jsonrpc":"2.0",                          JSON RPC protocol version. Always 2.0.
  "id":"70853aff-af77-420e-8f3c-fa9430733a19",       JSON RPC unique identifier
  "result":[                                Result list of objects from each CLI command executed.
    {
      "sourceDetails":"version 5.1.6.fcs\n!\n …",  Output information of CLI Command 1.
                                    The Original ASCII output information returned from CLI command if this command is successfully
executed.
      "errorCode":-1003,                    Error code if it is available.
      "errorDesc":"unsupported command…",       Error description if it is available.
      "warnings":"% Invalid…",                  Warnings if it is available.
                                    Formatted JSON object will also be returned if it is available.
    },
    {},                                     Output information of CLI Command 2.
    {},                                     Output information of CLI Command 3.
    {},                                     Output information of CLI Command 4.
    {},                                     Output information of CLI Command 5.
    {},                                     Output information of CLI Command 6.
    {},                                     Output information of CLI Command 7.
    {},                                     Output information of CLI Command 8.
    {},                                     Output information of CLI Command 9.
    {
      "sourceDetails":" Interface name       : eth-0-1\n Switchport mode       : trunk\n …\n"
    }                                       Output information of CLI Command 10.
  ]
}
```

Python Client Example Code

Here is an example code using 'pyjsonrpc' library:

```
import pyjsonrpc
import json

http_client = pyjsonrpc.HttpClient(
  url = "http://10.10.39.64:80/command-api",
  username = "username",
  password = "password"
)

cmds = {}
cmd_list = ["show run", "config t", "vlan database", "vlan 1-8", "interface eth-0-1", "switchport mode trunk", "switchport trunk
allowed vlan add 2", "shutdown", "end", "show interface switchport"]

cmds['cmds'] = cmd_list
cmds['format'] = 'text'
cmds['version'] = 1

try:
  response = http_client.call("executeCmds", cmds)
  print("json response:");
  json_result = json.dumps(response, indent=4)
```

```
  print(json_result)
except Exception, e:
  if e.code == 401:
    print "Unauthorized user"
  else:
    print e.message
    print e.data
```

Error code

Here is a list of JSON-RPC 2.0 error code:

Error Code	Description
-32700	Parse error
-32600	Invalid Request
-32601	Method not found
-32602	Invalid param
-32603	Internal error

Here is a list of RPC-API error code:

Error Code	Description
-1000	General error
-2001	JSON RPC API Error: unsupported API version
-2002	JSON RPC API Error: must specify 'params' with 'cmds' in JSON RPC
-2003	JSON RPC API Error: unsupported command response format
-3001	Command execution failed: timed out
-3002	Command execution failed: unsupported command
-3003	Command execution failed: unauthorized command
-3004	Command execution failed: the string does not match any command in current mode
-3005	Command execution failed: can't convert to JSON format
-3006	Command execution failed: command list too short
-3007	Command execution failed: command list too long

2.11.2 Configuration

Configuring RPC API service

User could enable the RPC API service by the following steps.

The default port is 80.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable RPC API service

```
Switch(config)# service rpc-api enable
```

 Use the following command to disable rpc-api service:

```
Switch(config)# service rpc-api disable
```

step 3 Exit the configure mode

```
Switch(config)# end
```

Configuring RPC API service with HTTP Authentication

User could configure the HTTP authentication mode of RPC API service.

Currently, only HTTP Basic authentication is supported. User will receive status code: 401 (Unauthorized access) if user provides invalid user name or password.

step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Set the username and password, then enable the rpc-api authentication

```
Switch(config)# username myuser password mypass privilege 4
Switch(config)# service rpc-api auth-mode basic
```

 Use the following command to disable authentication:

```
Switch(config)# no service rpc-api auth-mode
```

 HTTP authentication settings of RPC API service will take effect after you restart this service or reboot the system.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show services rpc-api
RPC API service configuration:
 Server State      : enable
 Port              : 80
 Authentication Mode : basic
 VRF               : default
```

2.11.3 Application cases

N/A

2.12 Configuring HTTP

2.12.1 Overview

Function Introduction

This chapter describes how to configure the switch to start the Web management function.

Principle Description

N/A

2.12.2 Configuration

Preparatory

Put a valid web image to flash: directory. Please reference to FTP or TFTP guide.

Configure HTTP server

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Load WEB image

```
Switch(config)# http server load flash:/webImage.bin
```

step 3 Configure HTTP server address （Optional）

Use this step to specify the source address of WEB http server， only loopback address is supported. If the source address of WEB http server is specified, it will be the only address to access the WEB. If the source address of WEB http server is not specified, user can access the WEB via the same address as telnet. The route between the device and the client is necessary.

```
Switch(config)# interface loopback 0
Switch(config-if)# ip address 192.168.1.100/32
Switch(config-if)# quit
Switch(config)# http server source address 192.168.1.100
This operation will cause all the online HTTP(S) users to be offline.
Continue? [yes/no]: yes
Switch(config)# ip route 0.0.0.0/0 192.168.1.1
```

step 4 Enable HTTP service

```
Switch(config)# service http enable
This operation will cause all the online HTTP(S) users to be offline.
Continue? [yes/no]: yes
```

step 5 Exit the configure mode

```
Switch(config)# exit
```

step 6 Login the web via the browser

Enter the IP address to login the web.

2.12.3 Application cases

N/A

3 Ethernet Configuration Guide

3.1 Configuring Interface

3.1.1 Overview

Function Introduction

Interface status, speed and duplex are configurable.

When the interface is configured as "no shutdown", it can work normally after cable is connected. When the interface is configured as "shutdown", no matter the cable is connected or not, the interface can not work.

If the device supports combo ports, user can choose to enable copper or fiber mode. The two modes of one port can not work together at same time. The configuration of speed or duplex at combo ports cannot be effective when combo port is working at fiber mode.

The rule of physical port name is as following: interface name format is eth-[slot]-[port]; [slot] is 0 for single pizza-box switch; when stacking is enabled, the [slot] number is according to the configuration. The [port] number is begin with 1, and increase from up to down, from left to right. The following figure shows the interface name of the device:

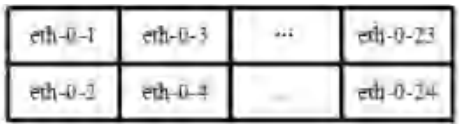


Figure 3-1 Interface Name


To get more information about the interface type and number， please reference to the product spec.

Principle Description

N/A

3.1.2 Configuration

Configuring Interface State

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Turn on an interface

```
Switch#(config)# interface eth-0-1
Switch(config-if)# no shutdown
```

step 3 Shut down an interface

```
Switch(config-if)# interface eth-0-2
Switch(config-if)# shutdown
```

step 4 Exit the configure mode

```
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the status of the interfaces:

```
Switch# show interface status
Port     Status    Duplex  Speed   Mode    Type
```

```
------------------------------------------------------------------
eth-0-1   up       a-full  a-1000  access  1000BASE_T
eth-0-2   admin down auto    auto    access  1000BASE_T
```

Configuring Interface Speed

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the speed

Set speed of interface eth-0-1 to 100M

```
Switch(config)# interface eth-0-1
Switch(config-if)# speed 100
Switch(config-if)# no shutdown
```

Set speed of interface eth-0-2 to 1000M

```
Switch(config-if)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# speed 1000
```

Set speed of interface eth-0-3 to auto

```
Switch(config-if)# interface eth-0-3
Switch(config-if)# no shutdown
Switch(config-if)# speed auto
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

Use the following command to display the status of the interfaces:

```
Switch# show interface status
Port     Status    Duplex  Speed   Mode   Type
------------------------------------------------------------------
eth-0-1   up       a-full  100     access  1000BASE_T
eth-0-2   up       a-full  1000    access  1000BASE_T
eth-0-3   up       a-full  a-1000  access  1000BASE_T
```

Configuring Interface Duplex

There are 3 duplex mode supported on the device:

- full mode: the interface can transmit and receive packets at same time.
- half mode: the interface can transmit or receive packets at same time.
- auto mode: the interface should negotiate with the other side to decide the duplex mode.

User can choose proper duplex mode according to the network state.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the duplex

Set duplex of interface eth-0-1 to full

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# duplex full
```

Set duplex of interface eth-0-1 to half

```
Switch(config-if)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# duplex half
```

Set duplex of interface eth-0-1 to auto

```
Switch(config)# interface eth-0-3
Switch(config-if)# no shutdown
Switch(config-if)# duplex auto
```

step 4 Validation

Use the following command to display the status of the interfaces:

```
Switch# show interface status
Port     Status   Duplex  Speed   Mode   Type
-------------------------------------------------------------
eth-0-1  up       full    a-1000  access 1000BASE_T
eth-0-2  up       half    a-100   access 1000BASE_T
eth-0-3  up       a-full  a-1000  access 1000BASE_T
```

3.1.3 Application cases

N/A

3.2 Configuring Group-speed of Interfaces

3.2.1 Overview

Function Introduction

The interfaces in the same group shares the same speed， which is called “group-speed”; Use the “group-speed” command in any interface will affect all other members in this group.

Principle Description

N/A

3.2.2 Configuration

View group ID info of interface

Before interface speed is set， users need know which interfaces belong to interface-group and which interfaces are in the same interface-group; the interfaces not in interface-group use speed command to set speed mode, and the interfaces in interface-group use group-speed command to set speed mode.

if group-id of interface is 0, it’s means that the interface doesn’t belong to interface-group, otherwise the interface belongs to interface-group; and the same group-id interfaces belong to the same interface-group.

step 1 show group ID info of interfaces

Use the following command to display the interface-group info of the interfaces:

```
Switch# show group-id info
Group Id   Member Ports
-----------+--------------------
0          eth-0-1  eth-0-2
```

```
        eth-0-3   eth-0-4
        eth-0-5   eth-0-6
        eth-0-7   eth-0-8
        eth-0-9   eth-0-10
        eth-0-11  eth-0-12
        eth-0-13  eth-0-14
        eth-0-15  eth-0-16
        eth-0-17  eth-0-18
        eth-0-19  eth-0-20
        eth-0-21  eth-0-22
        eth-0-23  eth-0-24
        eth-0-25  eth-0-26
        eth-0-27  eth-0-28
        eth-0-29  eth-0-30
        eth-0-31  eth-0-32
        eth-0-33  eth-0-34
        eth-0-35  eth-0-36
        eth-0-37  eth-0-38
        eth-0-39  eth-0-40
        eth-0-41  eth-0-42
        eth-0-43  eth-0-44
        eth-0-45  eth-0-46
        eth-0-47  eth-0-48
-----------------------------------
1         eth-0-49  eth-0-50
        eth-0-51  eth-0-52
Switch#
```

step 2 Set the group speed

```
Switch# configure terminal
Switch(config)# interface eth-0-49
Switch(config-if)# group-speed 1000
% Warning: Speed of Port eth-0-49, eth-0-50, eth-0-51, eth-0-52 will all be switched to 1000M after entering this CLI
```

3.2.3 Application cases

N/A

3.3 Configuring Layer3 Interfaces

3.3.1 Overview

Function Introduction

3 types of Layer3 interface are supported:

- VLAN interfaces: Logical interface with layer3 features. Connect different VLANs via IP address on the VLAN interface. VLAN interfaces can be created and deleted.
- Routed Ports: Ports are physical ports configured to be in Layer 3 mode by using the no switchport in interface configuration command.
- Layer 3 Link Aggregation Ports: Link Aggregation interfaces made up of routed ports.
- Layer 3 sub-interface: Virtual interface configured on physical port or link aggregation interface, it used to achieve device communication between vlans on an interface.

A Layer 3 switch can have an IP address assigned to each routed port and VLAN interface. All Layer 3 interfaces require an IP address to route traffic. This section shows how to configure an interface as a Layer 3 interface and how to assign an IP address to an interface.

Principle Description

N/A

3.3.2 Configuration

Configuring Routed Port

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set IP address

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 1.1.1.1/24
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

Use the following command to display the brief status of the interfaces:

```
Switch# show ip interface brief
Interface          IP-Address     Status            Protocol
eth-0-1            1.1.1.1        up                up
Switch# show ip interface
Interface eth-0-1
 Interface current state: UP
 Internet address(es):
  1.1.1.1/24 broadcast 1.1.1.255
 Joined group address(es):
  224.0.0.1
 The maximum transmit unit is 1500 bytes
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
ICMP unreachables are always sent
 ICMP mask replies are always sent
 ARP timeout 01:00:00,  ARP retry interval 1s
 VRRP master of:  VRRP is not configured on this interface
```

Configuring VLAN Interfaces

This chapter describes configuring VLAN interfaces and using them. Several Virtual LAN (VLAN) interfaces can be configured on a single Ethernet interface. Once created, a VLAN interface functions the same as any physical interface, and it can be configured and displayed like any physical interface. Routing protocols, such as, RIP, OSPF and BGP can run across networks using VLAN interfaces.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create a vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set switch port attributes

```
Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Enter the vlan interface configure mode and set IP address

```
Switch(config)# interface vlan10
Switch(config-if)# ip address 2.2.2.2/24
```

step 5 Exit the configure mode

```
Switch(config-if)# end
```

step 6 Validation

Use the following command to display the brief status of the interfaces:

```
Switch# show ip interface brief
Interface        IP-Address     Status          Protocol
vlan10           2.2.2.2        up              up

Switch# show ip interface
Interface vlan10
 Interface current state: UP
 Internet address(es):
  2.2.2.2/24 broadcast 2.2.2.255
 Joined group address(es):
  224.0.0.1
 The maximum transmit unit is 1500 bytes
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ICMP redirects are always sent
 ICMP unreachables are always sent
 ICMP mask replies are always sent
 ARP timeout 01:00:00,  ARP retry interval 1s
 VRRP master of :  VRRP is not configured on this interface
```

Configuring Layer3 sub-interface

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Step 3 Create layer3 sub-interface, configure terminal vlan, set IP address and enable statistic function

```
Switch(config)# interface eth-0-1.1
Switch(config-if)# dot1q termination vid 5
Switch(config-if)# ip address 2.2.2.2/24
```

```
Switch(config-if)# statistic enable both
Switch(config-if)# exit
```

Step 4 Exit the configure mode

```
Switch(config)# end
```

Step 5 Validation

Use the following command to display the status of the layer3 sub-interface:

```
Switch# show ip interface brief
Interface        IP-Address     Status            Protocol
eth-0-1.1        2.2.2.2        up                up

Switch# show interface eth-0-1.1
Interface eth-0-1.1
 Interface current state: UP
 Hardware is Subif, address is 98a4.a56e.ef00 (bia 98a4.a56e.ef00)
 Index 12288 , Metric 1 , Encapsulation ARPA
 The maximum transmit unit (MTU) is 1500 bytes
 VRF binding: not bound
 VRRP master of :  VRRP is not configured on this interface
 ARP timeout 01:00:00,  ARP retry interval 1s
 ARP Proxy is disabled, Local ARP Proxy is disabled
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes
  0 packets output, 0 bytes
```

3.4 Configuring Interface Errdisable

3.4.1 Overview

Function Introduction

Errdisable is a mechanism to protect the system through shutdown the abnormal interface. If an interface enters errdisable state, there are two ways to recovery it from errdisabled state. The first one is to enable errdisable recovery of this reason before errdisable detection; the interface will be recovered automatically after the configured time. But if errdisable occurred first, then errdisable recovery is enabled, the errdisable will not be recovered automatically. The secondary one is configuring “no shutdown” command on the errdisabled interface.

The flap of interface link state is a potential error caused by hardware or line problem. The administrator can also configure the detection conditions of interface link flap to suppress the flap.

Principle Description

N/ATerminology and definition

Errdisable Status: Interface which is shutdown by Errdisable feature because of abnormal event, we call this status as “Errdisable status” in this document.

Errdisable Reason: The abnormal event is called as “Errdisable reason”. The reasons are listed in the following table:

Table 3-1 Errdisable Reason Description

Name	Explanation
bpduguard	System received a BPDU packet on stp edge port
bpduloop	System received a BPDU packet sent by local device
link-monitor-failure	EFM exceed threshold detected
oam-remote-failure	EFM failure detected

port-security	System received a Packet with trustless MAC address
link-flap	Link Flapping
monitor-link	The uplink port is down， lead related downlink port enter errdisable status
udld	Unidirectional detected
fdb-loop	MAC address flapping
loopback-detection	Network loop back detection
reload-delay	The interface is in reload-delay period. The reload delay period specifies the interval that non-peer links are disabled after an MLAG peer reboots
dual-active-conflict	Dual active conflict in stacking environment detected

3.4.2 Configuration

Configuring Errdisable Detection

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable detect link flap errdisable

```
Switch(config)# errdisable detect reason link-flap
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the configuration of error disable：

```
Switch# show errdisable detect
ErrDisable Reason       Detection status
-----------------       ----------------
bpduguard               Enabled
bpduloop                Enabled
link-monitor-failure    Enabled
oam-remote-failure      Enabled
port-security           Enabled
link-flap               Enabled
monitor-link            Enabled
udld                    Disabled
fdb-loop                Disabled
loopback-detection      Enabled
reload-delay            Enabled
```

Configuring Errdisable Recovery

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable errdisable and set recovery interval

```
Switch(config)# errdisable recovery reason link-flap
Switch(config)# errdisable recovery interval 30
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the configuration of error disable recovery：

```
Switch# show errdisable recovery
ErrDisable Reason      Timer Status
-----------------      --------------
bpduguard              Disabled
bpduloop               Disabled
link-monitor-failure   Disabled
oam-remote-failure     Disabled
port-security          Disabled
link-flap              Enabled
udld                   Disabled
fdb-loop               Disabled
loopback-detection     Disabled
Timer interval: 30 seconds
```

Configuring suppress Errdisable link Flap

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set link flap condition

```
Switch(config)# errdisable flap reason link-flap 20 60
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the configuration of error disable flap：

```
Switch# show errdisable flap
ErrDisable Reason      Flaps    Time (sec)
-----------------      ------   ----------
link-flap              20       60
```

Checking Errdisable Status

Administrator can check the interface errdisable status though two commands.

Case 1 Enable errdisable recovery

If link flap errdisable is enabled recovery, the command will display the left time for recovery; Otherwise, will display "unrecovery".

```
Switch# show errdisable recovery
ErrDisable Reason      Timer Status
-----------------      --------------
bpduguard              Disabled
bpduloop               Disabled
link-monitor-failure   Disabled
oam-remote-failure     Disabled
port-security          Disabled
link-flap              Enabled
udld                   Disabled
```

```
fdb-loop              Disabled
loopback-detection    Disabled
Timer interval: 300 seconds
Interfaces that will be enabled at the next timeout:
Interface Errdisable Reason Time Left(sec)
--------- ----------------- --------------
eth-0-3   link-flap       25
```

Case 2 Disalbe errdisable recovery

```
Switch# show errdisable recovery
ErrDisable Reason      Timer Status
-----------------      --------------
bpduguard              Disabled
bpduloop               Disabled
link-monitor-failure   Disabled
oam-remote-failure     Disabled
port-security          Disabled
link-flap              Disabled
udld                   Disabled
fdb-loop               Disabled
loopback-detection     Disabled
Timer interval: 300 seconds
```

case 3 Display interface brief information to check errdisable state.

```
Switch# show interface status
Port     Status    Duplex  Speed   Mode   Type       Description
--------------------------------------------------------------------------------
eth-0-1   up       a-full  a-1000  TRUNK  1000BASE_SX
eth-0-2   down     auto    auto    TRUNK  Unknown
eth-0-3   errdisable a-full  a-1000  TRUNK  1000BASE_SX
eth-0-4   down     auto    auto    ACCESS Unknown
```

3.4.3 FAQ

The following example shows how to analyze the abnormality event fdb-loop and link-flap.

fdb-loop

Fdb-loop is also called as MAC address flapping. The switch records the source MAC address，VLAN，received interface into the MAC address table when learning MAC address. If the packets with same VLAN and same MAC address are continuously received from different interfaces, the MAC address table will be updated again and again by MAC learning function. This phenomenon is called fdb-loop (or MAC address flapping). Fdb loop may lead traffic forwarding error or network storm. It is an abnormality of the network.

Fdb-loop may cause by traffic loop back. The most common reason is BUM traffic cyclic forwarding when there is a loop in the network. The system checks frequency of flapping. If the frequency of flapping exceeds the specific threshold, it will meet the condition of fdb-loop. In the case of Errdisable reason fdb-loop enabled, system will set one of the fdb-loop interface to Errdisable status randomly. Otherwise in the case of Errdisable reason fdb-loop disabled, system only records a log and do nothing else.

Possible reasons analyze:

- It has misconnected network cables and network error form to a ring network, which will lead to MAC address flapping frequently.
- There are bunch of virtual machines drifting in the network to form MAC address flapping frequently.
- There is illegal traffic in the network, like unauthorized users launch MAC address attack and so on.
- There are different physical devices in the network, like testers and physical machines sending traffics that have same source MAC and VLAN.

Errdisable protecting fdb-loop is disabled by default， in order to prevent uplink port down and lead bigger service malfunction. If it is necessary to be enabled, the "errdisable fdb-loop trust" command can be used to protect the important service port such as uplink port which should never be shut down by Errdisable functions.

link-flap

Link-flap is defined as interface that converts UP/DOWN status continuously. Link-flap may lead service interrupt. A common scene is the primary link down and the service traffic switch to the backup link. If link flap happened on the primary link, the primary link and the backup link will switch frequently and impact the service. To prevent the instability status, system checks frequency of flapping. If the frequency of flapping exceeds the specific threshold, it will meet the condition of link-flap. In the case of Errdisable reason link-flap enabled, system will set the interface to Errdisable status. Otherwise in the case of Errdisable reason link-flap disabled, system only records a log and do nothing else.

Link-flap is mainly caused by physical reasons, such as cable/DAC/optical fiber/module issues, wavelength division devices rotating, etc.

Errdisable protecting link-flap is enabled by default. We do not recommend users to disable it.

3.5 Configuring MAC Address Table

3.5.1 Overview

Function Introduction

MAC address table contains address information for the switch to forward traffic between ports. The address table includes these types of address:

- Dynamic address: the source address learnt by the switch and will be aged after aging time if this address is not hit. We only support IVL learning mode.
- Static address: the source address manually added by administrators.

Following is a brief description of terms and concepts used to describe the MAC address table:

- IVL: Independent VLAN Learning: for a given set of VLANs, if a given individual MAC Address is learned in one VLAN, it can't be used in forwarding decisions taken for that address relative to any other VLAN in the given set.
- SVL: Shared VLAN Learning: for a given set of VLANs, if an individual MAC Address is learned in one VLAN, it can be used in forwarding decisions taken for that address relative to all other VLANs in the given set.

Reference to standard:IEEE 802.1D，IEEE 802.1Q

Principle Description

N/A

3.5.2 Configuration

Configuring Address Aging Time

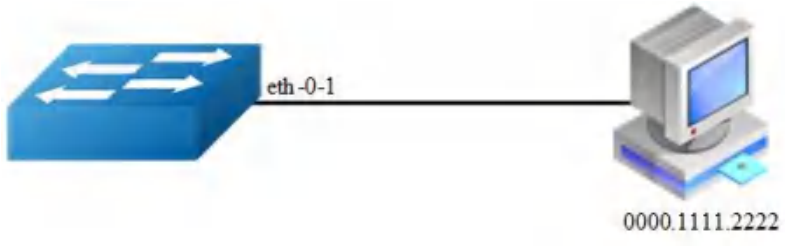


Figure 3-2 Mac address aging

The aging time is not exact time. If aging time set to N, then the dynamic address will be aged after N~2N interval. The default aging time is 300 seconds.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set dynamic address aging time

```
Switch(config)# mac-address-table ageing-time 10
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the aging time:

```
Switch# show mac address-table ageing-time
MAC address table ageing time is 10 seconds
```

Configuring Static Unicast Address

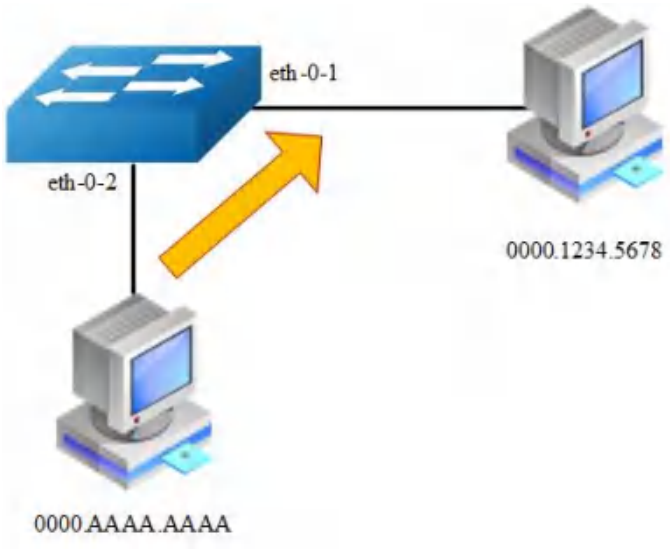


Figure 3-3 Static mac address table

Unicast address can be only bound to one port. According to the picture, Mac-Da 0000.1234.5678 should forward via eth-0-1.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set static mac address table

```
Switch(config)# mac-address-table 0000.1234.5678 forward eth-0-1 vlan 1
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the mac address table:

```
Switch# show mac address-table
Mac Address Table
-------------------------------------------
(*) - Security Entry
```

```
Vlan   Mac Address     Type    Ports
----  --------------  -------  --------
1     0000.1234.5678  static   eth-0-1
```

Configuring Static Multicast Address

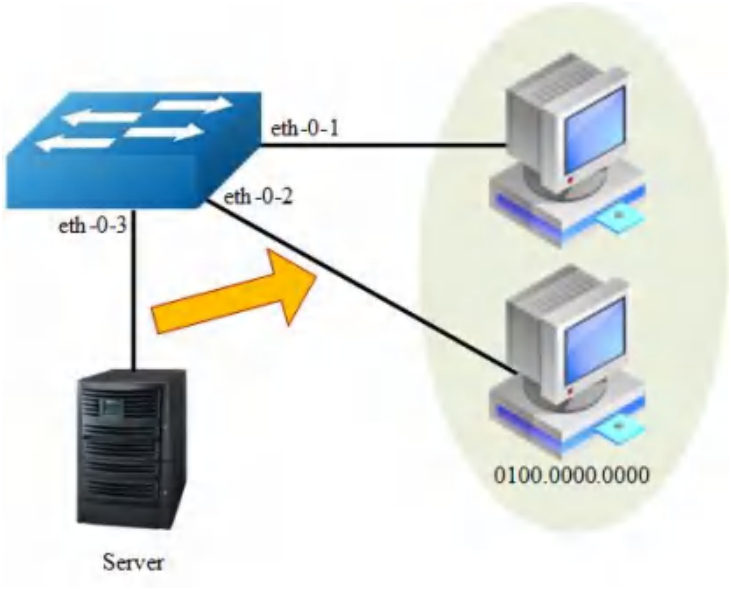


Figure 3-4 Static multicast mac address table

Multicast address can be bound to multi-port.According to the picture, Mac-Da 0100.0000.0000 can forward via eth-0-1 and eth-0-2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set static multicast mac address table

```
Switch(config)# mac-address-table 0100.0000.0000 forward eth-0-1 vlan 1
Switch(config)# mac-address-table 0100.0000.0000 forward eth-0-2 vlan 1
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the mac address table:

```
Switch# show mac address-table
  Mac Address Table
-------------------------------------------
(*) - Security Entry
Vlan   Mac Address      Type      Ports
----  --------------   -------    -------
1     0100.0000.0000   static     eth-0-1
                                  eth-0-2
```

Configuring MAC Filter Address

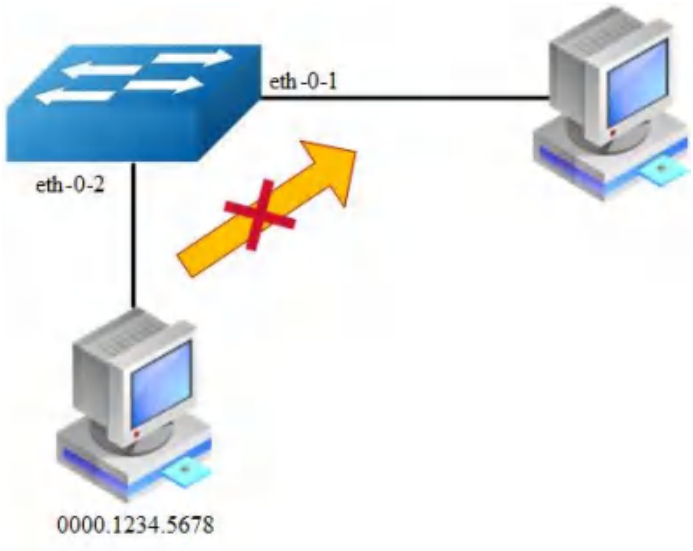


Figure 3-5 mac address filter

MAC filter will discard these frames whose source or destination address is set to discard. The MAC filter has higher priority than MAC address.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Add unicast address to be discarded

```
Switch(config)# mac-address-table 0000.1234.5678 discard
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the mac address filter:

```
Switch# show mac-filter address-table
   MAC Filter Address Table
-----------------------------------
Current count     : 1
Max count         : 128
Left count        : 127
Filter address list :
-----------------------------------
0000.1234.5678
```

Configuring MAC Filter Address with special VLAN

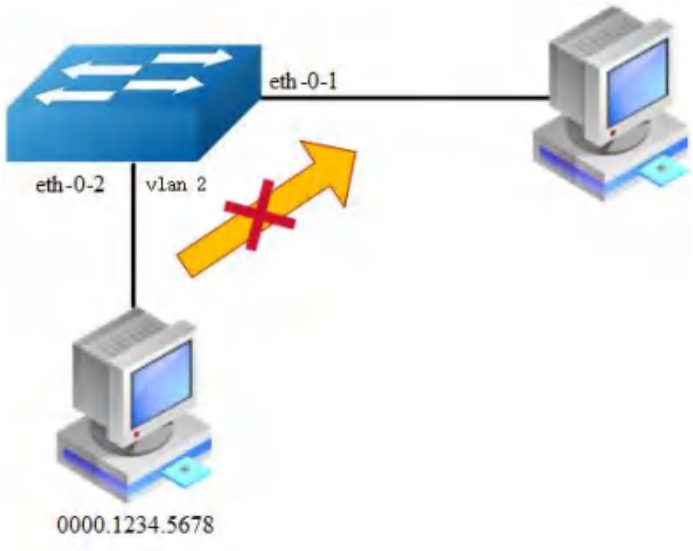


Figure 3-6 mac address filter with VLAN

MAC filter will discard these frames whose source or destination address is set to discard with VLAN matches. The MAC filter has higher priority than MAC address.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create VLAN

```
Switch(config)# vlan 2
```

step 3 Exit VLAN view

```
Switch(config-vlan)# exit
```

step 4 Add unicast address to be discarded

```
Switch(config)# mac-address-table 0000.1234.5678 discard vlan 2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the mac address filter with special VLAN:

```
Switch# show mac address-table blackhole
       Mac Address Table
------------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   Mac Address     Type       Ports
----   -----------     --------   -----
2      0000.1234.5678   blackhole   drop
```

3.5.3 Application cases

N/A

3.6 Configuring VLAN

3.6.1 Overview

Function Introduction

VLAN (Virtual Local Area Network) is a switched network that is logically segmented the network into different broadcast domain so that packets are only switched between ports that are designated for the same VLAN. Each VLAN is considered as a logical network, and packets send to stations that do not belong to the same VLAN must be forwarded through a router.

Reference to standard: IEEE 802.1Q

Principle Description

Following is a brief description of terms and concepts used to describe the VLAN:

- VID: VLAN identifier
- LAN: Local Area Network
- VLAN: Virtual LAN
- PVID: Port VID, the untagged or priority-tagged frames will be assigned with this VID

Tagged Frame: Tagged Frame is inserted with 4 Bytes VLAN Tag, show in the picture below:

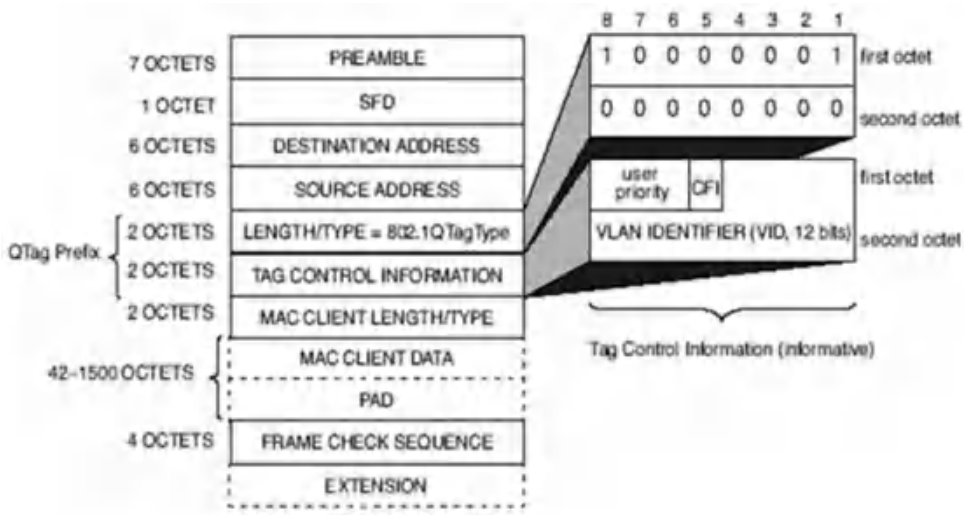


Figure 3-7 Tagged Frame

Trunk Link: Both tagged and untagged frames can be transmitted on this link. Trunk link allow for multiple VLANs to cross this link, show in the picture below:

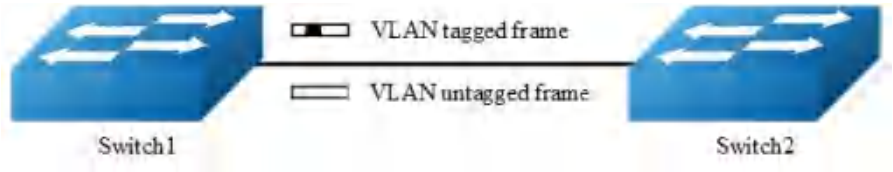


Figure 3-8 Trunk link

Access Link: Only untagged frames can be transmitted on this link. Access link is at the edge of the network, where end stations attach, show in the picture below:

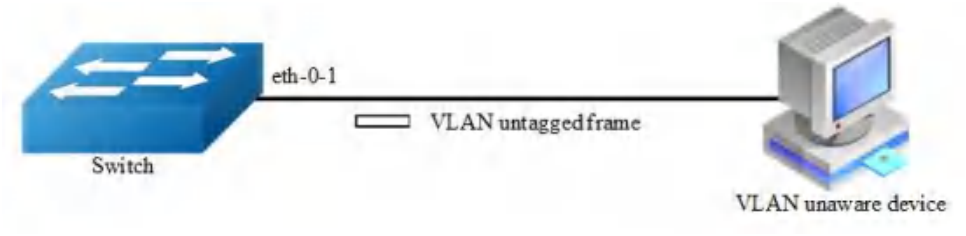


Figure 3-9 Access link

3.6.2 Configuration

Configuring Access Port

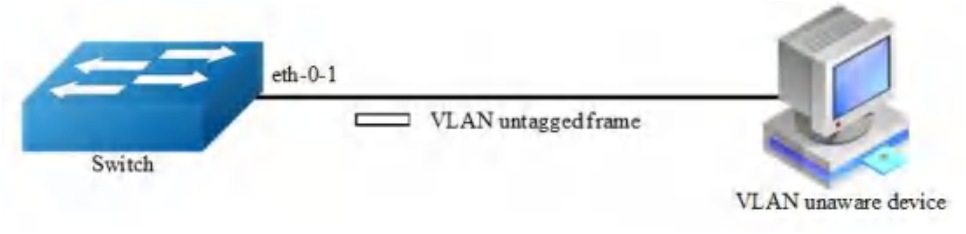


Figure 3-10 Access link

Access port only receives untagged or priority-tagged frames, and transmits untagged frames.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode, set the switch port mode and bind to the vlan

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2
```

step 4 Exit the configure mode

```
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of the switch port interface:

```
Switch# show interface switchport interface eth-0-1
Interface name        : eth-0-1
Switchport mode       : access
Ingress filter        : enable
Acceptable frame types  : vlan-untagged only
Default Vlan          : 2
Configured Vlans      :   2
```

Use the following command to display the vlan brief information:

```
Switch# show vlan brief
VLAN ID  Name          State  STP ID    Member ports
                                (u)-Untagged, (t)-Tagged
======= ================ ======= ======= ========================
1     default        ACTIVE  0     eth-0-2(u) eth-0-3(u)
                                   eth-0-4(u) eth-0-5(u)
                                   eth-0-6(u) eth-0-7(u)
                                   eth-0-8(u) eth-0-9(u)
                                   eth-0-10(u) eth-0-11(u)
                                   eth-0-12(u) eth-0-13(u)
                                   eth-0-14(u) eth-0-15(u)
                                   eth-0-16(u) eth-0-17(u)
                                   eth-0-18(u) eth-0-19(u)
                                   eth-0-20(u) eth-0-21(u)
                                   eth-0-22(u) eth-0-23(u)
2     VLAN0002       ACTIVE  0     eth-0-1(u)
```

Configuring Trunk Port

Trunk port receives tagged, untagged, and priority-tagged frames, and transmits both untagged and tagged frames. If trunk port receives an untagged frame, this frame will be assigned to the VLAN of the trunk port's PVID; if a frame send out from the trunk port and the frame's VID is equal to the trunk port's PVID, this frame will be send out without VLAN tag.

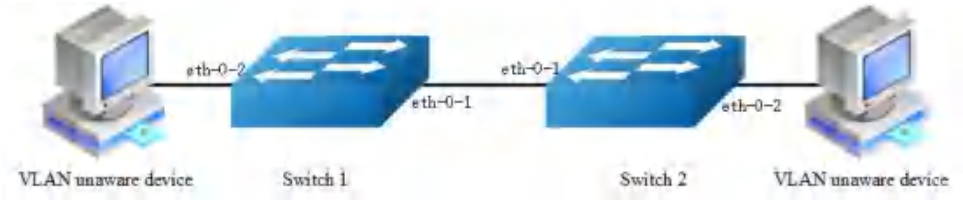


Figure 3-11 Trunk link

Network topology is shown in the picture above. The following configuration steps are same for Switch1 and Switch2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode, set the switch port mode and bind to the vlan

Set eth-0-1's switch port mode as trunk, set native vlan as 10, and allow all VLANs on this interface:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# switchport trunk native vlan 10
Switch(config-if)# exit
```

Set eth-0-2's switch port mode as access, and bind to vlan 10:

```
Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of the switch port interface:

```
Switch# show interface switchport
 Interface name        : eth-0-1
 Switchport mode       : trunk
 Ingress filter        : enable
 Acceptable frame types  : all
 Default Vlan          : 10
 Configured Vlans      :   1   10   20
 Interface name        : eth-0-2
 Switchport mode       : access
 Ingress filter        : enable
 Acceptable frame types  : vlan-untagged only
 Default Vlan          : 10
 Configured Vlans      :   10
```

Use the following command to display the vlan brief information:

```
Switch# show vlan brief
VLAN ID  Name          State   STP ID     Member ports
                                (u)-Untagged, (t)-Tagged
======= ================ ======= ========  ========================
1     default        ACTIVE  0      eth-0-1(t) eth-0-3(u)
                                    eth-0-4(u) eth-0-5(u)
                                    eth-0-6(u) eth-0-7(u)
                                    eth-0-8(u) eth-0-9(u)
                                    eth-0-10(u) eth-0-11(u)
                                    eth-0-12(u) eth-0-13(u)
                                    eth-0-14(u) eth-0-15(u)
                                    eth-0-16(u) eth-0-17(u)
                                    eth-0-18(u) eth-0-19(u)
                                    eth-0-20(u) eth-0-21(u)
                                    eth-0-22(u) eth-0-23(u)
10    VLAN0010       ACTIVE  0      eth-0-1(t) eth-0-2(u)
20    VLAN0020       ACTIVE  0      eth-0-1(t)
```

3.6.3 Application cases

N/A

3.7 Configuring Voice VLAN

3.7.1 Overview

Function Introduction

With the development of the voice technology, the use of IP Phone/IAD(Integrated Access Device) is becoming more and more widespread in broadband community. Voice and data traffics are usually present in the network at the same time, therfore, voice traffics need higher priority to improve the performance and reduce the packet loss rate.

The traditional method to improve the quality of voice traffic is using ACL to separate the voice packets, and using QoS to ensure the transmit quality.

The voice VLAN feature can identify the voice packets by source mac, which makes the conguration more convenient.

Principle Description

N/A

3.7.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2
Switch(config-vlan)# exit
```

step 3 Set the cos of voice vlan (Optional)

The default cos is 5.

```
Switch(config)# voice vlan set cos to 7
```

step 4 Set the voice vlan and create a mac entry for it

```
Switch(config)# voice vlan 2
Switch(config)# voice vlan mac-address 0055.0000.0000 ffff.ff00.0000 description test
```

step 5 Enter the interface configure mode and enable voice vlan

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# voice vlan enable
Switch(config-if)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
```

step 6 Validation

Send packet to eth-0-1, the format of the packet is as below（priority in Vlan tag is 0）：

```
0x0000:  0000 0a02 0001 0055 0000 0011 8100 0002  .......k.......
0x0010:  0800 aadd aadd aadd aadd aadd aadd aadd  ................
0x0020:  aadd aadd aadd aadd aadd aadd aadd aadd  ................
0x0030:  aadd aadd aadd aadd aadd aadd            ............
```

Receive packet from eth-0-2, the format of the packet received is as below（priority in Vlan tag is 5）：.

```
0x0000:  0000 0a02 0001 0055 0000 0011 8100 a002  .......k.......
0x0010:  0800 aadd aadd aadd aadd aadd aadd aadd  ................
0x0020:  aadd aadd aadd aadd aadd aadd aadd aadd  ................
0x0030:  aadd aadd aadd aadd aadd aadd            ............
```

3.7.3 Application cases

N/A

3.8 Configuring VLAN Classification

3.8.1 Overview

Function Introduction

VLAN classification is used to define specific rules for directing packets to selected VLANs based on protocol or subnet criteria. Sets of rules can be grouped (one group per interface).

VLAN classification rules have 3 types: mac based, ip based and protocol based. MAC based vlan classification rule will classify packets to specified VLAN according to the source MAC address of incoming packets; IP based vlan classification rule will classify packets according to the source IP address of incoming packets; And protocol based vlan classification rule will classify packets according to the layer3 type of incoming packets. The following layer3 types can be supported: ARP, IP(v4), MPLS, Mcast MPLS, PPPoE, RARP.

Different types of vlan classification rules can be added to same vlan classification group. VLAN classification group can only be applied on switchport. Only one type of vlan classification rules can take effect on one switchport.

Principle Description

N/A

3.8.2 Configuration

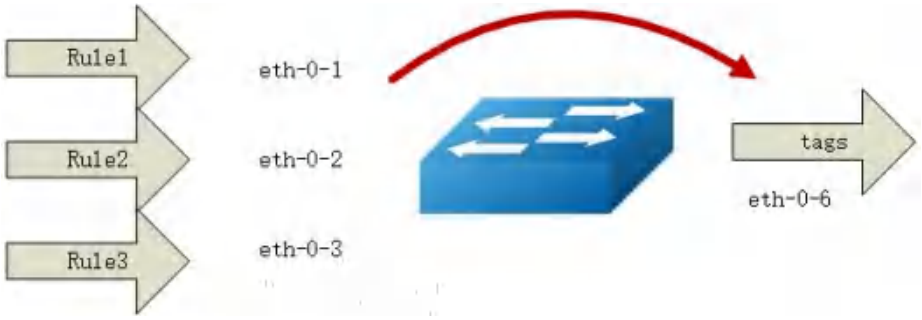


Figure 3-12 vlan classification

In this configuration example, three VLAN classifier rules are created:

Rule 1 is mac based rule, it will classify the packets with MACSA 2222.2222.2222 to vlan 5;

Rule 2 is ip based rule, it will classify the packets sourced from IP adress 1.1.1.1/24 to vlan 5;

Rule 3 is protocol based rule, it will classify all arp packets to vlan 5.

Add rule 1, rule2, rule3 to group 31. Then apply group 31 to 3 interfaces: eth-0-1, eth-0-2, eth-0-3. These 3 interfaces have different vlan classification type. eth-0-1 is configured to ip based vlan class, this means only ip based rules can take effect on this interface. eth-0-2 is configured to mac based vlan class, this means only mac based rules can take effect on this interface. eth-0-3 is configured to protocol based vlan class, this means only protocol based rules can take effect on this interface.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 5
Switch(config-vlan)# vlan 6
Switch(config-vlan)# exit
```

step 3 Create vlan classifier rule and add the rules to the group

```
Switch(config)# vlan classifier rule 1 mac 2222.2222.2222 vlan 5
Switch(config)# vlan classifier rule 2 ip 1.1.1.1 vlan 5
```

```
Switch(config)# vlan classifier rule 3 protocol arp vlan 5

Switch(config)# vlan classifier group 31 add rule 1
Switch(config)# vlan classifier group 31 add rule 2
Switch(config)# vlan classifier group 31 add rule 3
Switch(config)# arp as-layer-3 enable
```

step 4 Apply the vlan classifier group on the interface

interface eth-0-1：

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 6
Switch(config-if)# switchport access allowed vlan add 5
Switch(config-if)# vlan classifier activate 31 based ip
Switch(config-if)# exit
```

interface eth-0-2：

```
Switch(config)# interface eth-0-2
Switch(config-if)# switchport access vlan 6
Switch(config-if)# switchport access allowed vlan add 5
Switch(config-if)# vlan classifier activate 31 based mac
Switch(config-if)# exit
```

interface eth-0-3：

```
Switch(config)# interface eth-0-3
Switch(config-if)# switchport access vlan 6
Switch(config-if)# switchport access allowed vlan add 5
Switch(config-if)# vlan classifier activate 31 based protocol
Switch(config-if)# exit
```

interface eth-0-6：

```
Switch(config)# interface eth-0-6
Switch(config)#switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 5
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Verify the VLAN classifier rules:

```
Switch# show vlan classifier rule
vlan classifier rule 1 mac 2222.2222.2222 vlan 5
vlan classifier rule 2 ip 1.1.1.1 vlan 5
vlan classifier rule 3 protocol arp vlan 5
```

Verify the VLAN classifier group:

```
Switch# show vlan classifier group
vlan classifier group 31 add rule 1
vlan classifier group 31 add rule 2
vlan classifier group 31 add rule 3
```

Verify the VLAN classifier interface:

```
Switch# show vlan classifier interface grou
vlan classifier group 31 on interface eth-0-2, based mac
vlan classifier group 31 on interface eth-0-1, based ip
vlan classifier group 31 on interface eth-0-3, based protocol
```

3.8.3 Application cases

N/A

3.9 Configuring VLAN Mapping

3.9.1 Overview

Function Introduction

Service-provider business customers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning different VIDs to each customer to mapping their own's would bring the traffic from different customers separate. Using the VLAN translation feature, service providers can use a series of VLANs to support customers who have their own VLANs. Customer VLAN IDs are translated, and traffic from different customers is segregated within the service-provider infrastructure, even when they appear to be on the same VLAN.

802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets, and the maximal VLAN number can reach 4096 × 4096 . Using the 802.1Q tunneling feature, service providers can use a single VLAN to support clients which have multiple VLANs. The ISP usually builds a VLAN model to monitor whole VLAN of backbone network by using GARP or GVRP and accelerate network convergence speed by using STP. Using 802.1Q tunneling as initial solution is right at first, but it can cause expansibility problem as clients increased. Some clients hope to bring their own VLAN ID which will face two problems. Firstly, the first client's VLAN tag may clash with the other clients. Secondly, the usable tags may be severely limited for the service-provider. The core network will have limits on the 4096 numbers VLAN, if the clients are permitted to use their respective VLAN ID by their own manner.

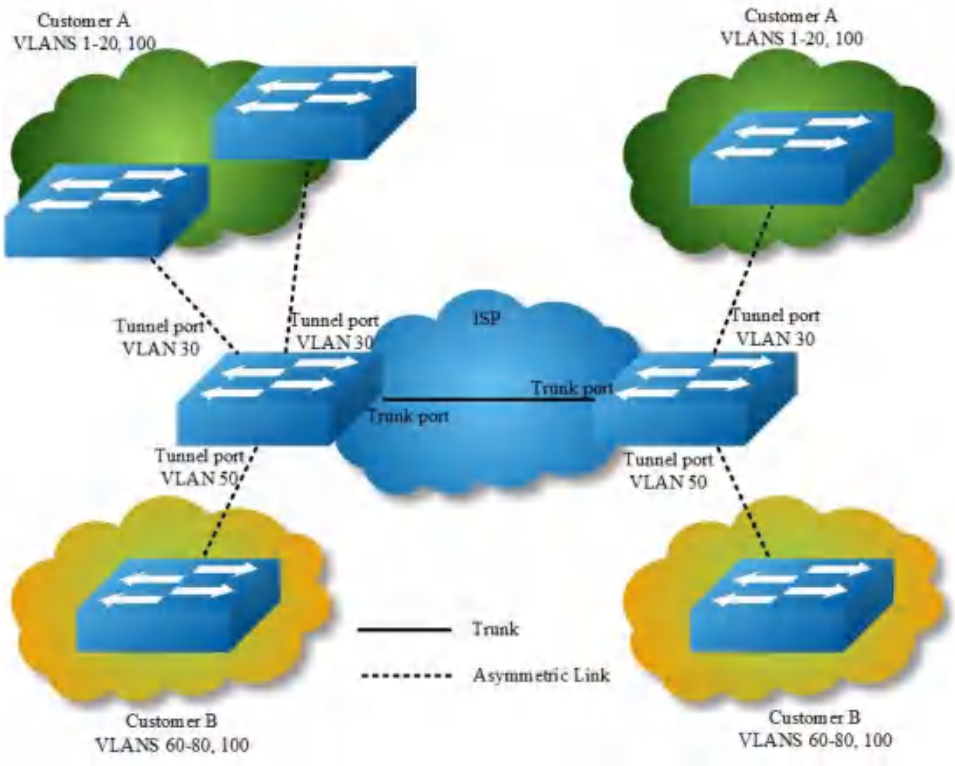


Figure 3-13 QinQ Tunnel

Using 802.1Q tunneling, the client's VLAN tag is encapsulated in the public VLAN tag and packets with two tags will traverse on backbone network. The client's VLAN tag will be shield and only the public VLAN tag will be used to transmit. By separating data stream,

the client's VLAN tag is transmitted transparently and different VLAN tags can be used repeatedly. Therefore, using 802.1Q tunneling expands the available VLAN tags. Two types of 802.1q tunneling are supported: basic 802.1Q tunneling and selective 802.1Q tunneling. Basic 802.1Q tunneling is founded on tagging on ports and all dates will be encapsulated a common VLAN tag of the same port, so this type has great limitations in practical applications. While selective 802.1Q tunneling can separate data stream and encapsulate different VLAN tags base on different data.

Principle Description

N/A

3.9.2 Configuration

Configuring VLAN Translation



Figure 3-14 vlan mapping

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2,3
Switch(config-vlan)# exit
```

step 3 Create evc and set dot1q mapped vlan

```
Switch(config)# ethernet evc evc_c1
Switch(config-evc)# dot1q mapped-vlan 2
Switch(config)# ethernet evc evc_c2
Switch(config-evc)# dot1q mapped-vlan 3
```

step 4 Create vlan mapping table and bind the vlan and evc

```
Switch(config)# vlan mapping table vm
Switch(config-vlan-mapping)# raw-vlan 10 evc evc_c1
Switch(config-vlan-mapping)# raw-vlan 20 evc evc_c2
Switch(config-vlan-mapping)# exit
```

step 5 Enable vlan translation on the interface and apply the vlan mapping table

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk vlan-translation
Switch(config-if)# switchport trunk vlan-translation mapping table vm
Switch(config-if)# switchport trunk allowed vlan add 2,3

Switch(config-if)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 2,3
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Use the following command to display the information of the switch port interface:

```
Switch# show interface switchport interface eth-0-1
Interface name        : eth-0-1
 Switchport mode      : trunk
 VLAN traslation      : enable
 VLAN mapping table   : vm
 Ingress filter       : enable
 Acceptable frame types : all
 Default Vlan         : 1
 Configured Vlans     : 1   2   3
```

Use the following command to display the information of the vlan mapping table：

```
Switch# show vlan mapping table
Table Name      EVC Name        Mapped VLAN Raw VLAN
 ================ ================ =========== ======================
vm            evc_c1         2        10
              evc_c2         3        20
```

Configuring 802.1q Tunneling （Basic 802.1Q tunneling）



Figure 3-15 QinQ Tunnel

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the switch port mode

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode dot1q-tunnel
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

This example shows how to configure a switchport to basic dot1q-tunnel port. You can use show the configuration on the switchport:

```
Switch# show interface switchport interface eth-0-1
Interface name        : eth-0-1
 Switchport mode      : dot1q-tunnel(basic)
 Ingress filter       : enable
 Acceptable frame types : all
```

```
Default Vlan        : 1
Configured Vlans    : 1
```

Configuring 802.1q Tunneling （Selective 802.1Q tunneling, Add one tag for incoming untagged packet.）

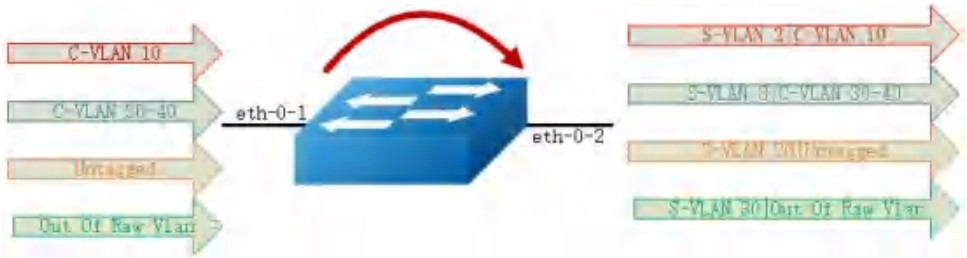


Figure 3-16 QinQ Tunnel

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2,3,20,30
Switch(config-vlan)# exit
```

step 3 Create evc and set dot1q mapped vlan

```
Switch(config)# ethernet evc evc_c1
Switch(config-evc)# dot1q mapped-vlan 2
Switch(config)# ethernet evc evc_c2
Switch(config-evc)# dot1q mapped-vlan 3
Switch(config)# ethernet evc evc_c3
Switch(config-evc)# dot1q mapped-vlan 20
Switch(config)# ethernet evc evc_c4
Switch(config-evc)# dot1q mapped-vlan 30
Switch(config-evc)# exit
```

step 4 Create vlan mapping table and bind the vlan and evc

```
Switch(config)# vlan mapping table vm
Switch(config-vlan-mapping)# raw-vlan 10 evc evc_c1
Switch(config-vlan-mapping)# raw-vlan 30-40 evc evc_c2
Switch(config-vlan-mapping)# raw-vlan untagged evc evc_c3
Switch(config-vlan-mapping)# raw-vlan out-of-range evc evc_c4
Switch(config-vlan-mapping)# exit
```

step 5 Enable vlan translation on the interface and apply the vlan mapping table

eth-0-1:

```
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# switchport dot1q-tunnel type selective
Switch(config-if)# switchport dot1q-tunnel vlan mapping table vm
Switch(config-if)# switchport dot1q-tunnel allowed vlan add 2,3,20,30
```

eth-0-2:

```
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 2,3,20,30
```

step 6 Exit the configure mode

```
Switch(config-if)# end
```

step 7 Validation

This example shows how to configure a switchport to selective dot1q-tunnel port:

```
Switch# show interface switchport interface eth-0-1
Interface name        : eth-0-1
Switchport mode       : dot1q-tunnel(selective)
VLAN mapping table    : vm
Ingress filter        : enable
Acceptable frame types : all
Default Vlan          : 1
Configured Vlans      : 1   2   3   20   30
```

Use the following command to display the information of the vlan mapping table：

```
Switch# show vlan mapping table
Table Name      EVC Name        Mapped VLAN Raw VLAN
=============== =============== =========== ==========================
vm              evc_c1          2           10
                evc_c2          3           30-40
                evc_c3          20          untagged
                evc_c4          30          out-of-range
```

Configuring 802.1q Tunneling （Selective 802.1Q tunneling, Add two tags for incoming untagged packet.）

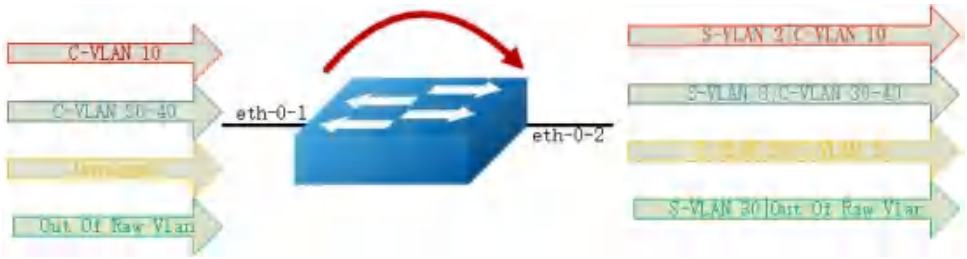


Figure 3-17 QinQ Tunnel

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2,3,10,20,30
Switch(config-vlan)# exit
```

step 3 Create evc and set dot1q mapped vlan

```
Switch(config)# ethernet evc evc_c1
Switch(config-evc)# dot1q mapped-vlan 2
Switch(config-evc)# exit
Switch(config)# ethernet evc evc_c2
Switch(config-evc)# dot1q mapped-vlan 3
Switch(config-evc)# exit
Switch(config)# ethernet evc evc_c3
```

```
Switch(config-evc)# dot1q mapped-double-vlan 10 20
Switch(config-evc)# exit
Switch(config)# ethernet evc evc_c4
Switch(config-evc)# dot1q mapped-vlan 30
Switch(config-evc)# exit
```

step 4 Create vlan mapping table and bind the vlan and evc

```
Switch(config)# vlan mapping table vm
Switch(config-vlan-mapping)# raw-vlan 10 evc evc_c1
Switch(config-vlan-mapping)# raw-vlan 30-40 evc evc_c2
Switch(config-vlan-mapping)# raw-vlan untagged evc evc_c3
Switch(config-vlan-mapping)# raw-vlan out-of-range evc evc_c4
Switch(config-vlan-mapping)# raw-vlan 10 20 egress-vlan untag
Switch(config-vlan-mapping)# exit
```

step 5 Enable vlan translation on the interface and apply the vlan mapping table

eth-0-1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# switchport dot1q-tunnel type selective
Switch(config-if)# switchport dot1q-tunnel vlan mapping table vm
Switch(config-if)# switchport dot1q-tunnel native inner-vlan 10
Switch(config-if)# switchport dot1q-tunnel allowed vlan add 2,3,20,30
Switch(config-if)# exit
```

eth-0-2:

```
Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 2,3,20,30
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

This example shows how to configure a switchport to selective dot1q-tunnel port:

```
Switch# show interface switchport interface eth-0-1
 Interface name        : eth-0-1
 Switchport mode       : dot1q-tunnel(selective)
 VLAN mapping table    : vm
 Ingress filter        : enable
 Acceptable frame types  : all
 Default Vlan          : 10
 Configured Vlans      : 1   2   3   20   30
```

Use the following command to display the information of the vlan mapping table：

```
Table Name      EVC Name        Mapped VLAN Raw VLAN
================ ================ =========== ================================
vm              evc_c1          2           10
                evc_c2          3           30-40
                evc_c3          20(10)      untagged
                evc_c4          30          out-of-range
```

3.9.3 Application cases

N/A

3.10 Configuring Link Aggregation

3.10.1 Overview

Function Introduction

Abbreviation

Abbreviation	Full Name
LACP	Link Aggregation Control Protocol
AGG Group	Aggregation Group
AGG IF	Aggregation Interface
Member IF	Member Interface

description

Link Aggregation technique is aggregating multiple physical Ethernet interfaces into one, form into an aggregation group in logical; multiple physical interfaces in the same aggregation group will be seen as one interface logically. Using link aggregation technique can reach the goal of increasing link's bandwidth by binding multiple physical interfaces into one logical interface without upgrading hardware. At the meanwhile as realizing the increased bandwidth, link aggregation uses backup link mechanism, which can improve the reliabilities between devices effectively.

Link binding is achieved by interface binding. Multiple ethernet interfaces become an aggregation group. The ethernet interfaces are member interfaces . Each aggregation group corresponding to a logical interface, which is called aggregation interface. The aggregation group and aggregation interface should use same index, for example: aggregation group 1 corresponding to aggregation interface 1.

Types of Aggregation group/interface:

- layer 2 Aggregation group/interface: All member interfaces are layer 2 interface
- layer 3 Aggregation group/interface: All member interfaces are layer 3 interface

Modes of link aggregation

There are two modes of link aggregation. The advantages for each mode are as following:

- static aggregation mode: After configured, if the member statues is up, the aggregation interface can enter up statues
- dynamic aggregation mode: Use LACP protocol to adapt the interface select/un-select statues according to the locol and remote information.

NOTE Aggregation groups with static mode called static aggregation group; Aggregation groups with dynamic mode called dynamic aggregation group

Status of member interface:

- w(waiting): in dynamic aggregation mode, this member can not forward data traffic
- s(suspend): in dynamic aggregation mode, this member can not forward data traffic
- B(bundle): this member can forward data traffic
- D(down/admin down): interface is in down status

NOTE The aggregation interface is up when at least one member is in "bundle" status.

Background

Current problem

As the network size continuously extend, users increase higher requirements about the bandwidth and reliability of key link. In traditional technology, it always changes interface boards with higher speed rate or devices that support higher speed rate of interface boards; however, this scheme will have higher cost and it is not flexible.

Advantages of link aggregation

Link Aggregation technique is aggregating multiple physical Ethernet interfaces into one, form into an aggregation group in logical; multiple physical interfaces in the same aggregation group will be seen as one interface logically. Using link aggregation technique can reach the goal of increasing link's bandwidth by binding multiple physical interfaces into one logical interface without upgrading hardware. At the meanwhile as realizing the increased bandwidth, link aggregation uses backup link mechanism, which can improve the reliabilities between devices effectively.

Advantages of link aggregation:

- Increase bandwidth: multiple physical interfaces combine into one interface logically; the bandwidth of aggregative interface is the sum of each member interface's bandwidth.
- Link redundancy protection: when there is a certain link fails, traffic can cut back to another useable member links, to increasing the reliability of link aggregation interface.
- Load balancing: within one aggregation group, it can realize that load balancing between member links

Principle Description

LACP Protocol



Figure 3-18 LACPDU Packet

Table 3-2 Fields in packet

Field	Length	Description
Destination Address	6 Bytes	Destination Address which is a multicast address (01-80-C2-00-00-02)
Source Address	6 Bytes	Source Address, which is the MAC address of the transmiting interface
Length/Type	2 Bytes	Protocal type 0x8809
Subtype	1 Bytes	sub type. 0x01 is LACP packet
Version Number	1 Bytes	Protocol version 0x01
TLV_type	1 Bytes	0x00: Terminator0x01:Actor0x02:Partner0x03:Collector
Actor_Information_Length	1 Bytes	Length of actor field. 20 Bytes
Actor_System_Priority	2 Bytes	Local system priority. Configurable. 32768(0x8000) by default
Actor_System	6 Bytes	System ID, which is same as local MAC address
Actor_key	2 Bytes	Interface Key value, which is generated by interface configuration. It is a key factor for the member to join the aggregation group. The interface configuration include AGG index and whether join the MLAG
Actor_Port_Priority	2 Bytes	Interface system priority. Configurable. 2048(0x800) by default
Actor_Port	2 Bytes	Interface index, according to the switch selected interface

Actor_State	1 Bytes	Local statues information, valid range is 0-7. 0(LACP_Activity): The passive/active statues of the interface's link control. 0 - Passive, 1 - Active.1(LACP_Timeout): timeout value. 0 - Long timeout, 1- short timeout.2(Aggregation): aggregation capacity of the interface. 1 - True, the interface is integrable. 0 - False, independent link, not integrable.3(Synchronization): if the interface is aggregated. 1 - True, the link is in IN_SYNC statues, the interface is in the correct aggregation group. 0 - False, the link is in OUT_OF_SYNC statues, the interface is not in the correct aggregation group.4(Collecting): 1 - True, receiving enabled on current interface. 0 - False, receiving disabled on current interface. 5(Distributing): 1 - True, transmitting enabled on current interface. 0 - False, transmitting disabled on current interface. 6:Defaulted： 1 - True, the partner information used by Actor is from the configured default value. 0 - False, he partner information used by Actor is from the LACPDU packets.7(Expired): 1 - True, Actor RX is timeout. 0 - False, Actor RX is not timeout.
Reserved	3 Bytes	Reserved for debug and extension
Partner_Information_Length	1 Bytes	Partner information length. Partner is the remote system/interface/statues information received by the interface, which is corresponding to actor field. At system initial, the partner field is filled with "0". After received the remote information, the partner field is updated.
Partner_System_Priority	2 Bytes	The priority of remote system
Partner_System	6 Bytes	The system index of remote system, which is same as the MAC address of remote device
Partner_key	2 Bytes	Remote interface KEY value
Partner_Port_Priority	2 Bytes	The priority of remote interface
Partner_Port	2 Bytes	Remote interface index
Partner_State	1 Bytes	Partner_State
Reserved	2 Bytes	Reserved field
Collector_Information_Length	1 Bytes	Collector information length 0x10
CollectorMaxDelay	2 Bytes	Maximum delay. 0xffff by default
Reserved	12 Bytes	Reserved field
Terminator_Length	1 Bytes	Terminator information length 0x00

Reserved	50 Bytes	Reserved field, which is filled with "0"
FCS	4 Bytes	cyclic redundancy check field

Dynamic aggregation mode is achieved via LACP protocol. The content of LACP protocol and working mechanism of dynamic aggregation mode are described as following:

LACP protocol based on standard IEEE802.3ad, is to achieved dynamic link aggregation. The devices running LACP protocol use LACPDU to exchange the information of the link aggregation.

The member interface of aggregation group can send/receive the LACPDU(Link Aggregation Control Protocol Data Unit) to advice the local information to remote end. After the remote device receive the LACPDU, it compare the information with other members and set the member statues to select/un-select.

LACP working mode supports ACTIVE/PASSIVE.

At least one device is ACTIVE mode. Only the device in ACTIVE mode can send LACPDU initiative. The device in PASSIVE mode can reply the LACPDU. If both devices are in PASSIVE mode, no LACPDU can be sent.

LACP protocol defines priority. Priority supports system priority and interface priority. The smaller number indicates higher priority.

- System priority: The device with higher priority can decide the select/un-select statues of interface.
- Interface priority: The interface with higher priority can be selected preferential.

Link Aggregation

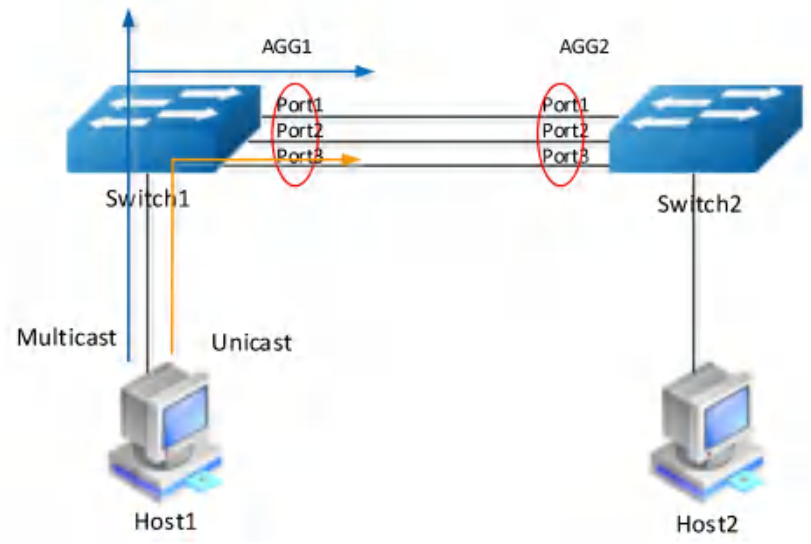


Figure 3-19 Link Aggregation

As it shows in Figure, the following explanation shows the forwarding process of link aggregation:

- Switch1's aggregation interface agg1 dock with Switch2's aggregation interface agg2, the aggregation link is formed between two switches.
- The aggregation interface is seen as an interface logically, when broadcast message forwards, there is only one broadcast on agg1, and then load balancing strategy will decide which physical interface will it forwards through.
- After the message gets to agg2, Switch2 learns MAC, the corresponding interface is agg2, rather than the physical interface which receives message.
- The aggregation link consists of three physical links in the figure, when the load traffic is extremely ideal, the link bandwidth of Switch1 and Switch2 will be the sum of the three physical links' bandwidths.

Link Redundancy

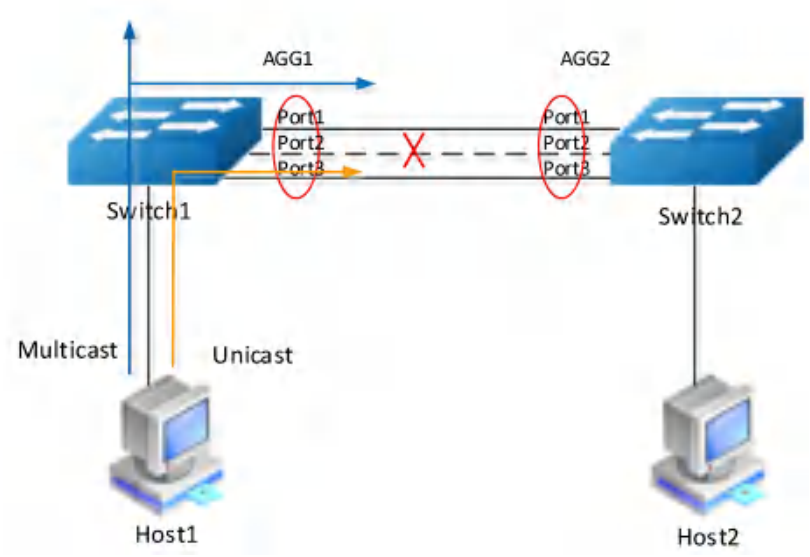


Figure 3-20 Link Redundancy

As the figure shows, when one link fails between Switch1 and Switch2, as long as other links are normal, then aggregation link can still normally forwarding network traffic; merely, the bandwidth of aggregation link decreases.

Load balance

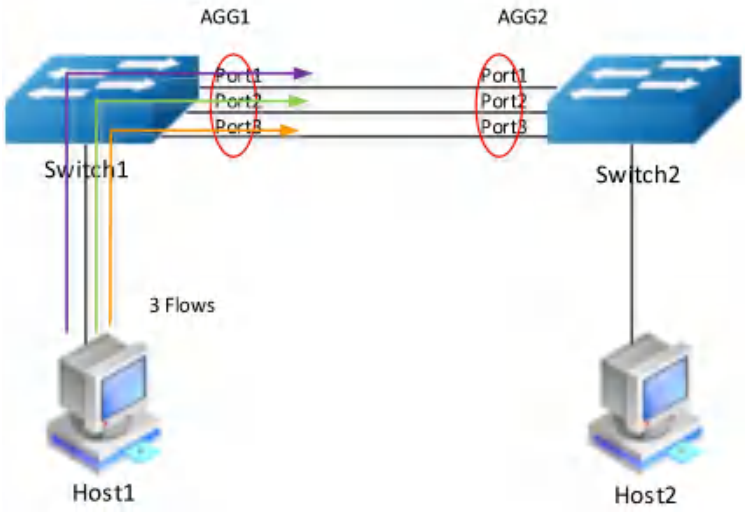


Figure 3-21 Load balance

As the figure shows, the traffic which forwards to agg1 will get away from different physical interfaces, to realize traffic's load balancing.

There are four modes of load balancing: Static(default), Dynamic, Round-robin, and resilent.

- Static: Hash load balancing mode, is to calculate a Hash value according to the Hash Key of message (the default of Hash Key is message's Dst-IP, Src-IP, Dst-MAC, Src-MAC and IP-Protocol); different Hash values match to different member link, which will realize the effects of load balancing. If the Hash Key sections of tested traffic are the same, then the Hash values calculated are the same, too. It cannot realize the effect in this situation.
- Dynamic: load balancing by flow. If port1 has fewer traffic than port2 and port3, the new flow will distribute to port1. But if multi flows enter the device at same time, these flows may be distributed to same port and lead unbalace.
- Round-robin: load balancing in round-robin mode. If the first packet is forwarded to port1, the second packet will forward to port2, the third packet will forward to port3, and so on. The problem is if the latency is different for each link, the packets may get out of order.

- Resilient: load balancing by hash. The difference with static modes is: in resilient mode, when there is link failure, system will re-calculate the hash value for the traffic on the failure link and re reload to the normal link.

High light functions

Weighted link aggregation

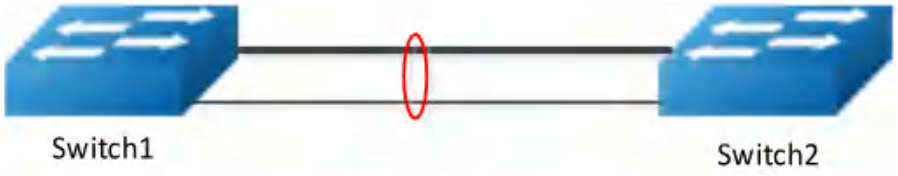


Figure 3-22 Weighted link aggregation

Switch1 and Switch2 are connected by two physical links. These two physical links constitute an aggregation link.

In some case, the two links have different bandwidth, so the different weight is needed to make sure the bandwidth can used reasonable.

Based on traditional link aggregation, the weighted link aggregation remove the limitation of member interface type, and it supports to aggregate interfaces with different speed. Using the weight configuration to ensure the traffic distribute to the member reasonable.

Interface weight configuration is valid for static AGG and LACP.

Master-slave mode link aggregation

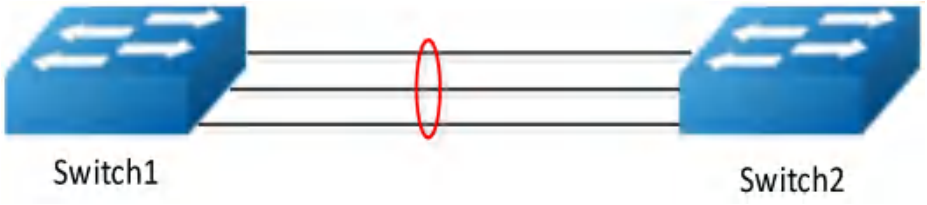


Figure 3-23 Master-slave mode link aggregation

Switch1 and Switch2 connected by aggregation link. To prevent single link failure lead network breaking or congestion, the system need a backup link.

As the figure shows, there are three physical links between Switch1 and Switch2. Two links are working link and one is backup link. If any of the working link failure occurs, the backup link can transmitting the packets instead of the working link.

Comparing with traditional link aggregation, Master-slave mode link aggregation has following advantages: in traditional link aggregation, links are the backup for each other, when failure occurs the traffic of the failed link need to distribute to other working links. When the network usage is high (for example,in full speed forwarding case), the packets loss can not be prevented.

The administrator can configure M working lins and N backup links, to make a reasonable ratio of M:N.

Master-slave mode link aggregation is only valid for LACP

3.10.2 Restrictions and Precautions

Interface Property

The aggregation interface is a logical interface, part of the attributes must be same among all interfaces. Therefore, when the member join an aggregation interface, it should follow the following rules:

Attribute	Description
interface	All member interfaces types (layer 2 / layer 3) must be same Layer 2 and layer 3 mixed is not allowedThe VLAN id of layer 2 interface should be same
DHCP	The interface with DHCP relay/DHCP snooping/DHCP server/DHCPv6 relay/DHCPv6 snooping enabled can not join the aggregation interface
Port isolate	The port isolate configuration must be same for each member interface
Private VLAN	The Private VLAN type / public VLAN / Private VLAN must be same for each member interface
Error disable	The Error disable configuration must be same for each member interface and aggregation interface
MTU	The MTU value must be same for each member interface and aggregation interface
EFD	The EFD status (enabled/disabled) must be same for each member interface and aggregation interface
MAC Learning	The MAC Learning status (enabled/disabled) must be same for each member interface and aggregation interface
Port Block	The Port Block configuration must be same for each member interface and aggregation interface

Table 3-3 Aggregation Conditions

Features mutual exclusion

Part of features are not supported on aggregation or member interface. See the following table:

Feature	Description
IP Source guard	NOT support on aggregation and member interfaces
IVI	NOT support on aggregation and member interfaces
Loop back detection	NOT support on aggregation and member interfaces
Monitor link	Interface with Monitor link enabled can NOT join the aggregation
G.8031/G.8032/ERPS	Interface with G.8031/G.8032/ERPS enabled can NOT join the aggregation.G.8031/G.8032/ERPS can be enabled on aggregation interface
PBR	Interface with PBR enabled can NOT join the aggregation.PBR can be enabled on aggregation interface
QOS policy map	Interface with QOS policy map enabled can NOT join the aggregation.QOS policy map can be enabled on aggregation interface
Port security	Interface with Port security enabled can NOT join the aggregation.Port security can be enabled on aggregation interface
Voice VLAN	Interface with Voice VLAN enabled can NOT join the aggregation.Voice VLAN can be enabled on aggregation interface

Table 3-4 Mutual exclusion

Feature	Description
IP Source guard	NOT support on aggregation and member interfaces
IVI	NOT support on aggregation and member interfaces
Loop back detection	NOT support on aggregation and member interfaces
Monitor link	Interface with Monitor link enabled can NOT join the aggregation
G.8031/G.8032/ERPS	Interface with G.8031/G.8032/ERPS enabled can NOT join the aggregation.G.8031/G.8032/ERPS can be enabled on aggregation interface
PBR	Interface with PBR enabled can NOT join the aggregation.PBR can be enabled on aggregation interface
QOS policy map	Interface with QOS policy map enabled can NOT join the aggregation.QOS policy map can be enabled on aggregation interface
Port security	Interface with Port security enabled can NOT join the aggregation.Port security can be enabled on aggregation interface
Voice VLAN	Interface with Voice VLAN enabled can NOT join the aggregation.Voice VLAN can be enabled on aggregation interface

Specification limitation

The maximum aggregation group/interface and the maximum member interface per-group are limited according to system resources

System supports 5 modes for aggregation maximum aggregation group/interface and member interface per-group. After the mode changes, it will teke effect after device reboot.

The 5 modes are described as following.

Table 3-5 Specification modes

Mode	Description
group-mode flexible	Default group-mode. Support up to 64 aggregation groups(Group 0 is reserved). Maximum member interface per-group is configurable. Support up to 16 member interfaces by default
group-mode 56	Support up to 56 aggregation groups(Group 0 is reserved).Support up to 16 member interfaces
group-mode 32	Support up to 32 aggregation groups(Group 0 is reserved).Support up to 32 member interfaces
group-mode 16	Support up to 16 aggregation groups(Group 0 is reserved).Support up to 64 member interfaces
group-mode 8	Support up to 8 aggregation groups(Group 0 is reserved).Support up to 128 member interfaces

The real specification according to different hardware condition, please reference to the .

3.10.3 Configuration

Configuring Static-channel-group

Topology
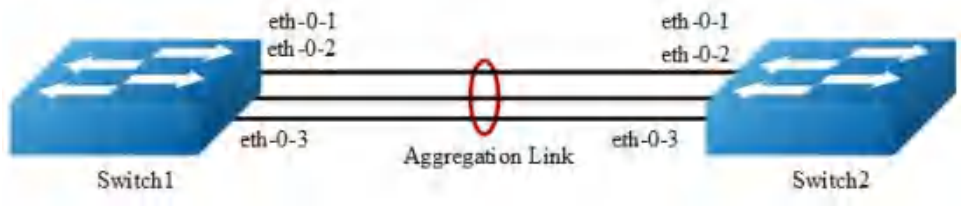


Figure 3-24 Static Agg

Configuration Steps

The configurations of Switch1 and Switch2 are as below:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and add the interface to the channel group

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# static-channel-group 1
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# static-channel-group 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# static-channel-group 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the information of the channel-group:

```
Switch1# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
      D - down/admin down   B - in Bundle
      R - Layer3            S - Layer2
      w - wait              U - in use
Mode:   SLB  - static load balance
      DLB  - dynamic load balance
      SHLB - self-healing load balance
      RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
----------------+---------+-----------+-----------------------------------------------
agg1(SU)       SLB      Static       eth-0-1(B) eth-0-2(B) eth-0-3(B)
```

Use the following command to display the information of the interface agg:

```
Switch1# show interface agg 1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia a876.6b2c.9c01)
 Bandwidth 3000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 140 bits/sec, 0 packets/sec
  0 packets input, 0 bytes
  Received 0 unicast, 0 broadcast, 0 multicast
  0 runts, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  0 input packets with dribble condition detected
  1080 packets output, 60614 bytes
  Transmitted 0 unicast, 0 broadcast, 0 multicast
0 underruns, 0 output errors, 0 pause output
```

Configure dynamic lacp
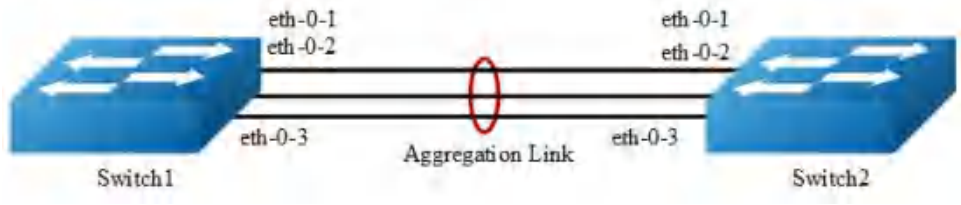


Figure 3-25 Dynamic LACP

The configurations of Switch1 and Switch2 are as below:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the global attributes of LACP

Set the dynamic lacp mode of aggregation groups.

Switch1 configuration:

```
Switch(config)# port-channel 1 lacp-mode dynamic
```

Switch2 configuration:

```
Switch(config)# port-channel 1 lacp-mode dynamic
```

step 3 Enter the interface configure mode and add the interface to the channel group

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# channel-group 1 mode active
```

```
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the information of the channel-group:

```
Switch# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
      D - down/admin down   B - in Bundle
      R - Layer3            S - Layer2
      w - wait              U - in use
Mode:   SLB  - static load balance
      DLB  - dynamic load balance
      SHLB - self-healing load balance
      RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
----------------+---------+--------------+-------------------------------------------
agg1(SU)       SLB      LACP(Dynamic)  eth-0-1(B) eth-0-2(B) eth-0-3(B)
```

Use the following command to display the information of the interface agg:

```
Switch1# show interface agg1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia cce3.33fc.330b)
 Bandwidth 3000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 2 bits/sec, 0 packets/sec
  13 packets input, 1184 bytes
  Received 0 unicast, 0 broadcast, 0 multicast
  0 runts, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  0 input packets with dribble condition detected
  20 packets output, 2526 bytes
```

```
Transmitted 0 unicast, 0 broadcast, 0 multicast
0 underruns, 0 output errors, 0 pause output
```

Configure channel-group
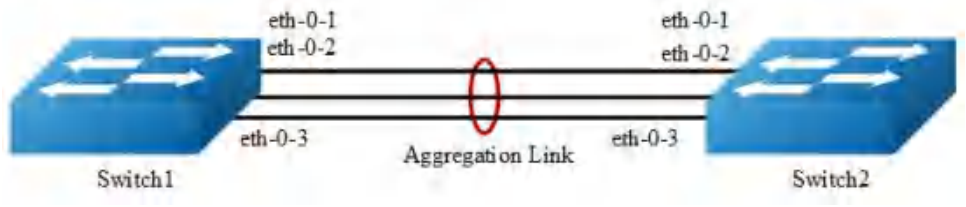


Figure 3-26 Static LACP

The configurations of Switch1 and Switch2 are as below:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the global attributes of LACP

Set the system priority of this switch. This priority is used for determining the system that is responsible for resolving conflicts in the choice of aggregation groups. A lower numerical value has a higher priority.Set the load balance mode. In this case we choose source MAC address for load balance.

Switch1 configuration:

```
Switch(config)# lacp system-priority 2000
Switch(config)# hash-field port-channel
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip disable
```

Switch2 configuration:

```
Switch(config)# lacp system-priority 1000
Switch(config)# hash-field port-channel
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip disable
```

step 3 Enter the interface configure mode and add the interface to the channel group

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the information of the channel-group:

```
Switch# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
      D - down/admin down   B - in Bundle
      R - Layer3            S - Layer2
      w - wait              U - in use
Mode:   SLB  - static load balance
      DLB  - dynamic load balance
      SHLB - self-healing load balance
      RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
---------------+---------+------------+-----------------------------------------------
agg1(SU)       SLB      LACP        eth-0-1(B) eth-0-2(B) eth-0-3(B)
```

Use the following command to display the information of the interface agg:

```
Switch1# show interface agg1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia cce3.33fc.330b)
 Bandwidth 3000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 2 bits/sec, 0 packets/sec
   13 packets input, 1184 bytes
   Received 0 unicast, 0 broadcast, 0 multicast
   0 runts, 0 giants, 0 input errors, 0 CRC
   0 frame, 0 overrun, 0 pause input
   0 input packets with dribble condition detected
   20 packets output, 2526 bytes
   Transmitted 0 unicast, 0 broadcast, 0 multicast
   0 underruns, 0 output errors, 0 pause output
```

Configuring weighted link aggregation

Topology

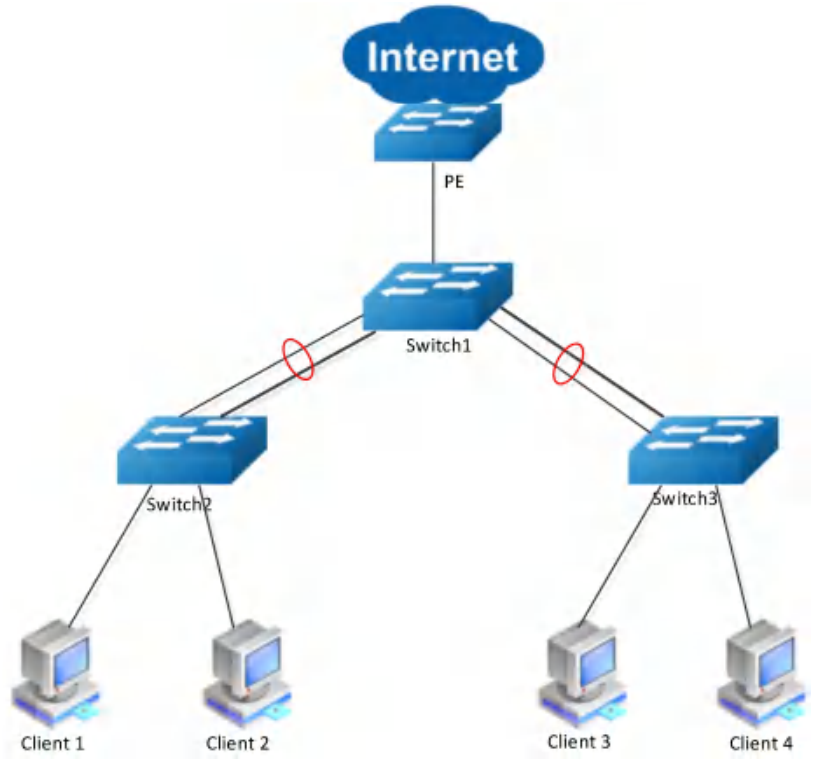


Figure 3-27 weighted link aggregation

Configuration Ideas

AS the figure shows， Switch2 and Switch3 are access devices. The uplink device is the aggregation layer Switch1. There are two links with different bandwidth between the access and aggregation devices, which become a weight aggregation link. The interface weight configuration is valid for both static AGG and LACP, the following example shows how to enable weighted link aggregation on LACP .

NOTE In this example, we suppose that the physical bandwidth of eth-0-1 is 1 Gbps, the physical bandwidth of eth-0-25 is 10 Gbps. So we can set the weight of interface eth-0-25 as 10 times of the weight of eth-0-1. The real interface type and bandwidth according to different hardware model, please reference to the product specification.

The ratio of weight and bandwidth is not mandatory. Users can choose the weight value according the network requirements.

Configuration Steps

This chapter shows how to configure Switch1 and Switch2.

The Switch3 is same as Switch1.

The configurations of Switch1 and Switch2 are as below:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# exit
Switch(config)# interface eth-0-25
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# distribute-weight 10
```

```
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

NOTE By default, the weight of member interfaces are 1.

System can only support member weight for static load balance and non self-healing mode. The total weight of all members should not exceed the maximum member count.

step 3 Validation

Show channel-group summary information:

```
Switch# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
     D - down/admin down   B - in Bundle
     R - Layer3          S - Layer2
     w - wait           U - in use
Mode:   SLB  - static load balance
     DLB  - dynamic load balance
     SHLB - self-healing load balance
     RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
----------------+---------+--------------+-----------------------------------------------
agg1(SU)       SLB      LACP(Dynamic)  eth-0-1(B) eth-0-25(B)
```

Show interface agg1 information:

```
Switch# show interface agg1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia cce3.33fc.330b)
 Bandwidth 11000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 2 bits/sec, 0 packets/sec
  13 packets input, 1184 bytes
  Received 0 unicast, 0 broadcast, 0 multicast
  0 runts, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  0 input packets with dribble condition detected
  20 packets output, 2526 bytes
  Transmitted 0 unicast, 0 broadcast, 0 multicast
  0 underruns, 0 output errors, 0 pause output
```

Configuring Master-slave mode link aggregation

Topology

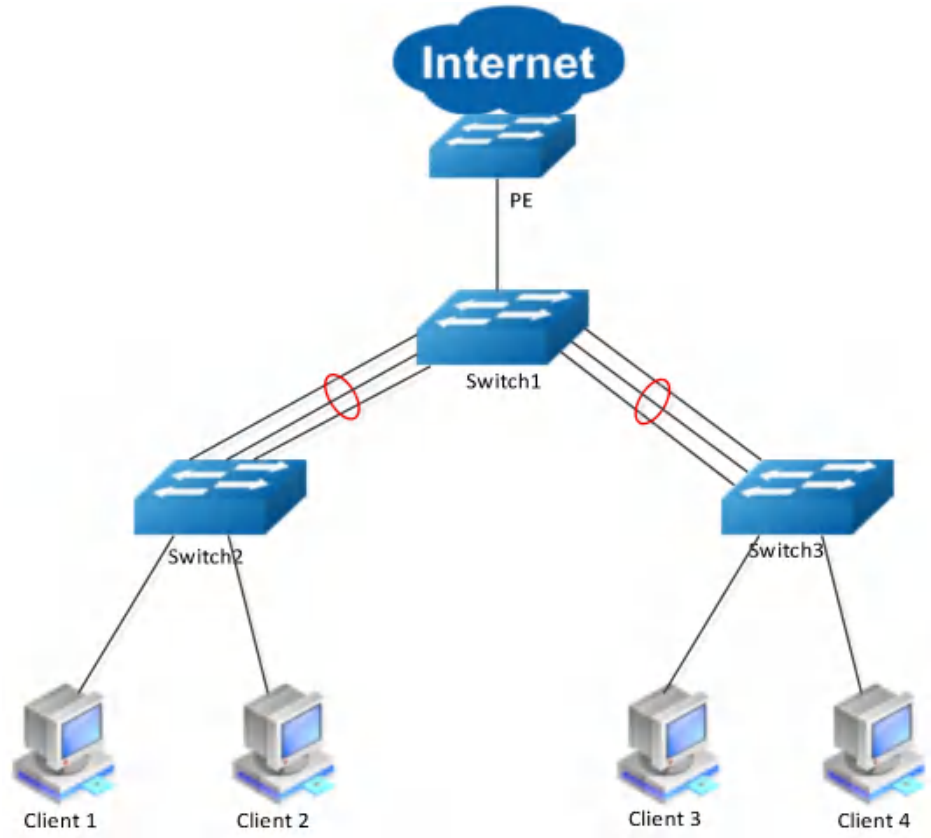


Figure 3-28 Master-slave mode link aggregation

Configuration Ideas

AS the figure shows， Switch2 and Switch3 are access devices. The uplink device is the aggregation layer Switch1. The uplink of the access device need large bandwidth to satisfy multi-clients requirements. It also need redundancy link to prevent uplink failure leading all downlink clients out of service.

In this example, 2:1 backup is configured, which means two working links and one backup link in the network.

Configuration Steps

This chapter shows how to configure Switch1 and Switch2.

The Switch3 is same as Switch1.

The configurations of Switch1 and Switch2 are as below:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface eth-0-3
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Configure the interface

```
Switch(config)# interface agg 1
Switch(config-if)# lacp max-active-linknumber 2
Switch(config-if)# lacp preempt enable
Switch(config-if)# lacp preempt delay 20
Switch(config-if)# end
```

NOTE "lacp max-active-linknumber 2" specifies at most two links in working statues. There is one link in backup statues because we have three members. When any of the working links failed, the backup link can become working link.

"lacp preempt enable" enables preempt. "lacp preempt delay 20" means the link start to preempt working link after 20 seconds. In this case after the previous working link recoverd, it wait for 20 seconds to preempt and become working link again. "preempt delay" can prevent link switch too often when link flaps. The default preempt delay value is 30. If all link has same parameter, the preempt feature can be disabled. By default the preempt feature is disabled.

Validation

Show channel-group summary information:

```
Switch# show channel-group summary
port-channel group-mode: flexible

Flags:  s - Suspend         T - Standby
      D - Down/Admin down   B - In bundle
      R - Layer3            S - Layer2
      w - Wait              U - In use

Mode:   SLB  - Static load balance
      DLB  - Dynamic load balance
      RR   - Round robin load balance
      RLB  - Resilient load balance
Aggregator Name  Mode     Protocol    Ports
----------------+---------+--------------+-------------------------------------
-------
agg1(SU)       SLB     LACP         eth-0-9(B)     eth-0-10(B)    eth-0-
11(T)
```

Show interface agg 1 information:

```
Switch# show interface agg 1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is 845b.cf07.8009 (bia 845b.cf07.8009)
 Bandwidth 80000000 kbits
 Index 2049 , Metric 1 , Encapsulation ARPA
 Speed - 80.00Gb/s , Duplex - Full  , Media type is Aggregation
 Link type is autonegotiation
 FEC config: DEFAULT
 FEC ability: NONE
 FEC status: OFF
 The Maximum Frame Size is 9600 bytes
 VRF binding: not bound
 ARP timeout 01:00:00,  ARP retry interval 1s
```

```
ARP Proxy is disabled, Local ARP Proxy is disabled
  5 minute input rate 71 bits/sec, 0 packets/sec
  5 minute output rate 68 bits/sec, 0 packets/sec
  73 packets input, 10804 bytes
  Received 0 unicast, 0 broadcast, 73 multicast
  0 runts, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  72 packets output, 10656 bytes
  Transmitted 0 unicast, 0 broadcast, 72 multicast
  0 underruns, 0 output errors, 0 pause output
  0 output discard
```

Server dual network cards binding and switches interaction

Server dual network cards binding mode	Corresponding configuration for switch side
Bond0	Use static aggregation
Bond1	No link-agg configuration required
Bond2	Use static aggregation
Bond3	Different switch's interfaces with different VLANs
Bond4	Use dynamic aggregation
Bond5	No link-agg configuration required
Bond6	No link-agg configuration required

NOTE In actual use, Bond4 is more popular, so we recommend you to use Bond4.

3.10.4 Application cases

Topology

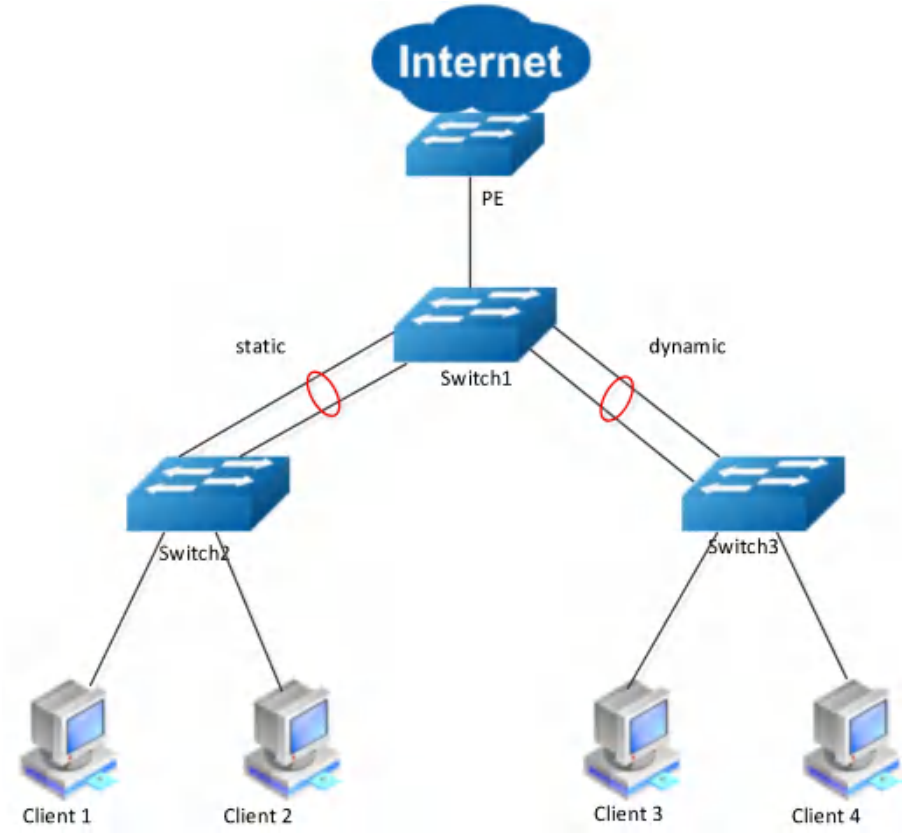


Figure 3-29 Aggregation application

Configuration Ideas

AS the figure shows， Switch2 and Switch3 are access devices. The uplink device is the aggregation layer Switch1. The uplink of the access device need large bandwidth to satisfy multi-clients requirements. It also need redundancy link to prevent uplink failure leading all downlink clients out of service.

In this case, we use link aggregation to satisfy the requirements.

- If both devices supports dynamic LACP, it is recommended to use LACP
- If any of the devices can not support dynamic LACP, the static aggregation is the only choice.

This example shows how to configure static aggregation between Switch1 and Switch2; it also shows how to configure LACP protocol between Switch1 and Switch3.

Configuration Steps for static aggregation between Switch1 and Switch2：

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and add member interface into the static aggregation interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# static-channel-group 1
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# static-channel-group 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

NOTE Before first member join the aggregation, the layer2/layer3 type of the interface must be confirmed. The aggregation's type is inherited form the first member.

If the layer2/layer3 type is different, the new member can not join the aggregation interface.

step 3 Enter the aggregation interface configure mode to set attributes

```
Switch(config)# interface agg 1
Switch(config-if)# ip address 10.10.10.1/24
```

NOTE The following attributes can NOT be modified: switch-port/no switch-port, duplex, speed.

The VLAN ID can be changed for layer2 aggregation.

Configuration Steps for LACP protocol between Switch1 and Switch3：

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable LACP globally

Method 1: Dynamic LACP

NOTE If LACP negotiation faild, each member interface can inherite the VLAN attributes and forward layer2 traffic independently. Dynamic LACP is mainly used in PXE(Preboot Execution Environmemt) scene. It is not recommended to use dynamic LACP for switches connection, because negotiation failure may lead network loop.

```
Switch(config)# port-channel 2 lacp-mode dynamic
```

Method 2: Static LACP

NOTE The default mode is static LACP. Use the following command to restore static mode：

```
Switch(config)# no port-channel 2 lacp-mode
```

step 3 Enter the interface configure mode and add member interface into the aggregation interface

```
Switch(config)# interface eth-0-3
Switch(config-if)# no shutdown
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# exit
Switch(config)# interface eth-0-4
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Validation

Switch1 and Switch2 information:

Show channel-group summary information:

```
Switch1# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
      D - down/admin down   B - in Bundle
      R - Layer3          S - Layer2
      w - wait            U - in use
Mode:   SLB  - static load balance
      DLB  - dynamic load balance
      SHLB - self-healing load balance
      RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
---------------+---------+------------+-----------------------------------------------
agg1(SU)       SLB      Static       eth-0-1(B) eth-0-2(B)
```

Show interface agg 1 information:

```
Switch1# show interface agg 1
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia a876.6b2c.9c01)
 Bandwidth 2000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 140 bits/sec, 0 packets/sec
  0 packets input, 0 bytes
  Received 0 unicast, 0 broadcast, 0 multicast
```

```
  0 runs, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  0 input packets with dribble condition detected
  1080 packets output, 60614 bytes
  Transmitted 0 unicast, 0 broadcast, 0 multicast
0 underruns, 0 output errors, 0 pause output
```

Switch1 and Switch3 information:

Show channel-group summary information:

```
Switch# show channel-group summary
port-channel load-balance hash-arithmetic: xor
port-channel load-balance hash-field-select:
   macsa
Flags:  s - suspend         T - standby
     D - down/admin down   B - in Bundle
     R - Layer3          S - Layer2
     w - wait            U - in use
Mode:  SLB  - static load balance
     DLB  - dynamic load balance
     SHLB - self-healing load balance
     RR   - round robin load balance
Aggregator Name  Mode     Protocol    Ports
----------------+---------+--------------+-----------------------------------------------
agg2(SU)       SLB      LACP(Dynamic)  eth-0-3(B) eth-0-4(B)
```

Show interface agg2 information:

```
Switch# show interface agg2
Interface agg1
 Interface current state: UP
 Hardware is AGGREGATE, address is cce3.33fc.330b (bia cce3.33fc.330b)
 Bandwidth 2000000 kbits
 Index 1025 , Metric 1 , Encapsulation ARPA
 Speed - 1000Mb/s , Duplex - Full  , Media type is Aggregation
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 ARP timeout 01:00:00,  ARP retry interval 1s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 2 bits/sec, 0 packets/sec
  13 packets input, 1184 bytes
  Received 0 unicast, 0 broadcast, 0 multicast
  0 runs, 0 giants, 0 input errors, 0 CRC
  0 frame, 0 overrun, 0 pause input
  0 input packets with dribble condition detected
  20 packets output, 2526 bytes
  Transmitted 0 unicast, 0 broadcast, 0 multicast
  0 underruns, 0 output errors, 0 pause output
```

3.11 Configuring Flow Control

3.11.1 Overview

Function Introduction

Flow control enables connected Ethernet ports to control traffic rates during congestion by allowing congested nodes to pause link operation at the other end. If one port experiences congestion and cannot receive any more traffic, it notifies the other port to stop sending until the condition clears. When the local device detects any congestion at its end, it can notify the link partner or the remote device of the congestion by sending a pause frame. You can use the flowcontrol interface configuration command to set the interface's ability to receive and send pause frames to on, off. The default state for ports is receive off and send off. In auto-negotiation link, local device's flow control ability can be notified to link partner by link up/down.


Flow control send/receive on ability only works on full duplex link

Principle Description

N/A

3.11.2 Configuration

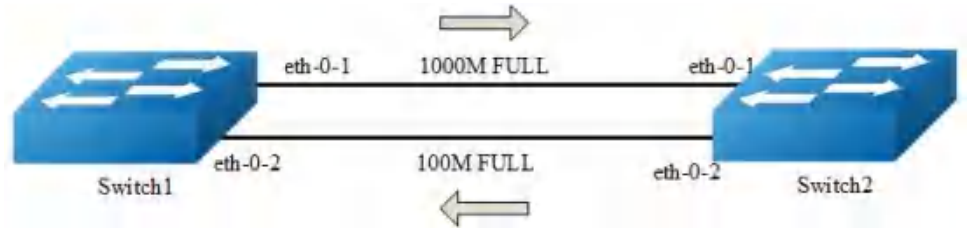


Figure 3-30 Flow control

Configuring Flow Control Send

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and enable flowcontrol send

```
Switch(config)# interface eth-0-1
Switch(config-if)# flowcontrol send on
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

Use the following command to display the information of flow control:

```
Switch# show flowcontrol
Port      Receive FlowControl  Send FlowControl   RxPause     TxPause
          admin   oper         admin   oper
--------- -------- --------   -------- --------  ----------- -----------
eth-0-1   off     off         on      on        0           0
eth-0-2   off     off         off     off       0           0
eth-0-3   off     off         off     off       0           0
```

Use the following command to display the information of flow control on specified interface:

```
Switch# show flowcontrol eth-0-1
Port      Receive FlowControl  Send FlowControl   RxPause     TxPause
          admin   oper         admin   oper
--------- -------- --------    -------- --------  ----------- -----------
eth-0-1   off     off          on      on        0           0
```

Configuring Flow Control Receive

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and enable flowcontrol send

```
Switch(config)# interface eth-0-1
Switch1(config-if)# flowcontrol receive on
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

Use the following command to display the information of flow control:

```
Switch1# show flowcontrol
Port      Receive FlowControl  Send FlowControl   RxPause     TxPause
          admin   oper         admin   oper
--------- -------- --------    -------- --------  ----------- -----------
eth-0-1   on      on           off     off       0           0
eth-0-2   off     off          off     off       0           0
eth-0-3   off     off          off     off       0           0
```

Use the following command to display the information of flow control on specified interface:

```
Switch1# show flowcontrol eth-0-1
Port      Receive FlowControl  Send FlowControl   RxPause     TxPause
          admin   oper         admin   oper
--------- -------- --------    -------- --------  ----------- -----------
eth-0-1   on      on           off     off       0           0
```

3.11.3 Application cases

N/A

3.12 Configuring Storm Control

3.12.1 Overview

Function Introduction

Storm control prevents traffic on a LAN from being disrupted by a broadcast, a multicast, or a unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance.

Storm control uses one of these methods to measure traffic activity:

- Bandwidth as a percentage of the total available bandwidth of the port (Level mode).
- Traffic rate in packets per second of the port (PPS mode).

PPS = Packets per second

Principle Description

N/A

3.12.2 Configuration

Configuring Bandwidth Percentage Storm Control

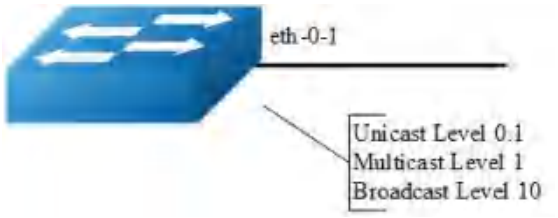


Figure 3-31 Percentage Storm Control

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, and set the storm control level

User can set different level for Unknown unicast/multicast/broad cast packets:

```
Switch(config)# interface eth-0-1
Switch(config-if)# storm-control unicast level 0.1
Switch(config-if)# storm-control multicast level 1
Switch(config-if)# storm-control broadcast level 10
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

```
Switch# show storm-control interface eth-0-1
Port     ucastMode ucastlevel   bcastMode bcastLevel   mcastMode mcastLevel
--------------------------------------------------------------------------------
eth-0-1  Level    0.10       Level    10.00      Level    1.00
```

Configuring Packets per-Second Storm Control

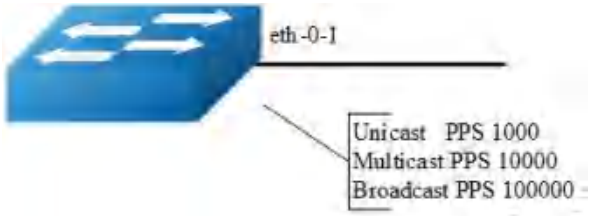


Figure 3-32 PPS Storm Control

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, and set the storm control pps

User can set different pps for Unknown unicast/multicast/broad cast packets:

```
Switch(config)# interface eth-0-1
Switch(config-if)# storm-control unicast pps 1000
Switch(config-if)# storm-control multicast pps 10000
Switch(config-if)# storm-control broadcast pps 100000
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

```
Switch# show storm-control interface eth-0-1
Port     ucastMode ucastlevel   bcastMode bcastLevel   mcastMode mcastLevel
-------------------------------------------------------------------------------
eth-0-1  PPS      1000        PPS      100000      PPS      10000
```

3.12.3 Application cases

N/A

3.13 Configuring Loopback Detection

3.13.1 Overview

Function Introduction

The loopback in the networks would cause the device continued to send broadcast, multicast and unknow unicast packets. It will waste the resource of network even paralysis the whole network. To detect the loopback in the layer 2 network rapidly and avoid to effect the whole network, system need to provide a detection function to notice the user checking the network connection and configuration, and control the error interface when the network appears loopback.

Loopback Detection can detects whether the interface of device exists loopback. When enable loopback detection on a interface, device will send detection packets from this interface by periodically. If the device receives detection packets sent from the interface, this interface is considered that there is a loop existed and the device can send alarm information to network management system. Administraitors discover loopback problem througt alarm information and resolve the problem to avoid longtime network abnormal. In addition, the device can control the specific interface and configured Trap according the requirement, and disable the interface to quickly reduce the impact in the network of loopback to the minimum.

Principle Description

N/A

3.13.2 Configuration

Enable Loopback Detect

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, and enable Loopback Detect

```
Switch(config)# interface eth-0-1
Switch(config-if)# loopback-detect enable
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

By default, loopback detection is disable. When the interface enable loopback detection, system send the detection packets to detect the loopback. Default detection packets transmission interval is 5 second.

Use the following command to display the loopback detection states：

```
Switch# show loopback-detect
Loopback detection packet interval(second): 5
Loopback detection recovery time(second):  15
Interface      Action         Status
eth-0-2        shutdown       NORMAL
```

Configuring Loopback Detect packet interval

The network is changing all the time, therefor the loopback detection is an continued process. The interface sent loopback detection packets in a certain interval of time, the packets transimission time is loopback detection packets sending period.

The device send the lopback detection packets time interval range is 1 to 300 seconds.The loopback status recover period default is 3 times of the interface send interval.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 set the packet interval of Loopback Detect

```
Switch(config)# loopback-detect packet-interval 10
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the packet interval of Loopback Detect:

```
Switch# show loopback-detect packet-interval
Loopback detection packet interval(second): 10
```

Configuring Loopback Detect action

If a loopback is detected on the interface and loopback is enabled on this interfac, the system can configure an action to send alarm, shutdown the interface, block the interface or other action.

After loopback detection is enabled on an interface, the interface sends loopback detection packets at intervals. When a loopback is detected on the interface, the system performs an action to minimize the impact on the entire network.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, and set the action of Loopback Detect

```
Switch(config)# interface eth-0-1
Switch(config-if)# loopback-detect action shutdown
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the information of Loopback Detect on the interface:

```
Switch# show loopback-detect interface eth-0-1
Interface      Action         Status
eth-0-1        shutdown       NORMAL
```

Configuring specify VLAN Loopback Detection

specify the VLAN IDs of loopback detection packets on an interface After loopback detection is enabled on an interface, system send untagged loopback detection packets by default. It means the device dosen't detect any specify vlan loopback packets. When interface is configured Tagged mode in vlan, the loopback detection packets sent by this interface will be discard on the link, and interface won't receive the loop packets which is sent by itself. So we should specify the VLAN IDs of loopback detection packets on an interface.

After the loopback-detect packet vlan command is executed on an interface, the interface sends an untagged loopback detection packet and the loopback detection packets with the specified VLAN tags. The specified VLANs exist and the interface has been added to the VLANs in tagged mode. If you run the loopback-detect packet vlan command multiple times in the same interface view, multiple VLAN IDs are specified. You can specify a maximum of eight VLAN IDs

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, and set the specify vlan of Loopback Detect

```
Switch(config)# interface eth-0-1
Switch(config-if)# loopback-detect packet vlan 20
```

step 3 Exit the configure mode

```
Switch(config-if)# end
```

step 4 Validation

Use the following command to display the configuration of Loopback Detect:

```
Switch# show running-config interface eth-0-1
Building configuration...
!
interface eth-0-1
 loopback-detect enable
 loopback-detect packet vlan 20
!
```

3.13.3 Application cases

N/A

3.14 Configuring Layer 2 Protocols Tunneling

3.14.1 Overview

Function Introduction

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure.

When Layer 2 protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a new Layer 2 header and send them across the service-provider network. Core switches in the network do not process these packets but forward them as normal packets. Layer 2 protocol packets pass the service-provider infrastructure and reach customer switches on the outbound side of the service-provider network. The new Layer 2 header will be stripped when the Layer 2 protocol packets are sent to customer switches. Layer 2 protocol tunneling can be used independently or can enhance 802.1Q tunneling.

Principle Description

N/A

3.14.2 Configuration

Tunnel Designed Layer2 Protocol Packets

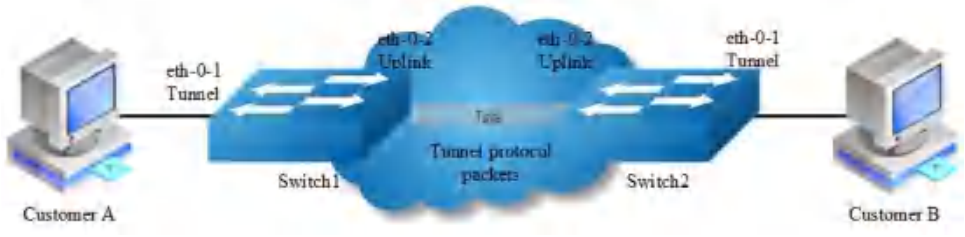


Figure 3-33 L2 protocol tunnel

The designed Layer2 protocol packets include STP BPDU, LACP slow proto, DOT1X EAPOL, CFM.

In this example, one link is between Switch1 and Switch2. Switch1 eth-0-1 and Switch2 eth-0-1 are configured tunnel port. Switch1 eth-0-2 and Switch2 eth-0-2 are configured uplink port. If protocol packets are received on port eth-0-1 of Switch1, packets should be added new Layer 2 header and sent out from uplink port. The new Layer 2 header will be as follows: MAC da should be tunnel dmac; MAC sa should be switch route-mac; VLAN ID should be tunnel vid; VLAN priority (cos) should be Layer 2 Protocol cos; Ethertype should be 0xFFEE. When the packets with new Layer 2 header are received on port eth-0-2 of Switch2, new Layer 2 header will be stripped and the packets will be sent to port eth-0-1 of Switch2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2-4
Switch(config-vlan)# exit
```

step 3 Create evc and set dot1q mapped vlan

```
Switch(config)# ethernet evc evc_c1
Switch(config-evc)# dot1q mapped-vlan 2
Switch(config-evc)# exit

Switch(config)# ethernet evc evc_c2
Switch(config-evc)# dot1q mapped-vlan 3
Switch(config-evc)# exit

Switch(config)# ethernet evc evc_c3
Switch(config-evc)# dot1q mapped-vlan 4
Switch(config-evc)# exit
```

step 4 Enable l2 protocol，set the tunnel destination mac and add l2 protocao mac address

```
Switch(config)# l2protocol enable
Switch(config)# l2protocol tunnel-dmac 0100.0CCD.CDD2
Switch(config)# l2protocol mac 3 0180.C200.0008
Switch(config)# l2protocol mac 4 0180.C200.0009
Switch(config)# l2protocol full-mac 0100.0CCC.CCCC
```

step 5 Enter the interface configure mode and set the attributes of the interfaces. Bind the l2 protocol mac and the evc

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
```

```
Switch(config-if)# switchport trunk allowed vlan add 2-4
Switch(config-if)# spanning-tree port disable
Switch(config-if)# l2protocol mac 3 tunnel evc evc_c1
Switch(config-if)# l2protocol mac 4 tunnel evc evc_c2
Switch(config-if)# l2protocol full-mac tunnel evc evc_c3
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 2-4
Switch(config-if)# l2protocol uplink enable
```

step 6 Exit the configure mode

```
Switch(config-if)# end
```

step 7 Validation

Use the following command to display the information of tunnel interface:

```
switch1# show l2protocol interface eth-0-1
Interface  PDU Address      MASK          Status   EVC
       (u)-Untagged
       (t)-Tagged
========= ================= ============== ======== ================
eth-0-1   0180.c200.0008(u)  ffff.ffff.ffff  Tunnel   evc_c1
eth-0-1   0180.c200.0008(t)  ffff.ffff.ffff  Tunnel   evc_c1
eth-0-1   0180.c200.0009(u)  ffff.ffff.ffff  Tunnel   evc_c2
eth-0-1   0180.c200.0009(t)  ffff.ffff.ffff  Tunnel   evc_c2
eth-0-1   0100.0ccc.cccc(u)  ffff.ffff.ffff  Tunnel   evc_c3
eth-0-1   0100.0ccc.cccc(t)  ffff.ffff.ffff  Tunnel   evc_c3
eth-0-1   stp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-1   stp(t)          ffff.ffff.ffff  Peer     N/A
eth-0-1   slow-proto(u)     ffff.ffff.ffff  Peer     N/A
eth-0-1   slow-proto(t)     ffff.ffff.ffff  Peer     N/A
eth-0-1   dot1x(u)        ffff.ffff.ffff  Peer     N/A
eth-0-1   dot1x(t)        ffff.ffff.ffff  Peer     N/A
eth-0-1   cfm(u)          ffff.ffff.ffff  Peer     N/A
eth-0-1   cfm(t)          ffff.ffff.ffff  Peer     N/A
eth-0-1   lldp(u)         ffff.ffff.ffff  Peer     N/A
eth-0-1   lldp(t)         ffff.ffff.ffff  Peer     N/A
eth-0-1   cdp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-1   cdp(t)          ffff.ffff.ffff  Peer     N/A
eth-0-1   vtp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-1   vtp(t)          ffff.ffff.ffff  Peer     N/A
```

Use the following command to display the information of uplink interface:

```
switch1# show l2protocol interface eth-0-2
Interface  PDU Address      MASK          Status   EVC
       (u)-Untagged
       (t)-Tagged
========= ================= ============== ======== ================
eth-0-2   0180.c200.0008(u)  ffff.ffff.ffff  Peer     N/A
eth-0-2   0180.c200.0008(t)  ffff.ffff.ffff  Peer     N/A
eth-0-2   0180.c200.0009(u)  ffff.ffff.ffff  Peer     N/A
eth-0-2   0180.c200.0009(t)  ffff.ffff.ffff  Peer     N/A
eth-0-2   0100.0ccc.cccc(u)  ffff.ffff.ffff  Peer     N/A
eth-0-2   0100.0ccc.cccc(t)  ffff.ffff.ffff  Peer     N/A
```

```
eth-0-2   stp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-2   stp(t)          ffff.ffff.ffff  Peer     N/A
eth-0-2   slow-proto(u)     ffff.ffff.ffff  Peer     N/A
eth-0-2   slow-proto(t)     ffff.ffff.ffff  Peer     N/A
eth-0-2   dot1x(u)        ffff.ffff.ffff  Peer     N/A
eth-0-2   dot1x(t)        ffff.ffff.ffff  Peer     N/A
eth-0-2   cfm(u)          ffff.ffff.ffff  Peer     N/A
eth-0-2   cfm(t)          ffff.ffff.ffff  Peer     N/A
eth-0-2   lldp(u)         ffff.ffff.ffff  Peer     N/A
eth-0-2   lldp(t)         ffff.ffff.ffff  Peer     N/A
eth-0-2   cdp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-2   cdp(t)          ffff.ffff.ffff  Peer     N/A
eth-0-2   vtp(u)          ffff.ffff.ffff  Peer     N/A
eth-0-2   vtp(t)          ffff.ffff.ffff  Peer     N/A
eth-0-2   N/A             N/A           Uplink   N/A
```

Use the following command to display the information of tunnel destination mac:

```
Switch1# show l2protocol tunnel-dmac
Layer2 protocols tunnel destination MAC address is 0100.0ccd.cdd2
```

3.14.3 Application cases

N/A

3.15 Configuring MSTP

3.15.1 Overview

Function Introduction

The MSTP (Multiple Spanning Tree Algorithm and Protocol (IEEE 802.1Q-2005)) enables multiple VLANs to be mapped to the same spanning-tree instance, thereby reducing the number of spanning-tree instances needed to support a large number of VLANs. The MSTP provides for multiple forwarding paths for data traffic and enables load balancing. It improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths). The most common initial deployment of MSTP is in the backbone and distribution layers of a Layer 2 switched network; this deployment provides the highly-available network required in a service-provider environment. When the switch is in the multiple spanning-tree (MST) modes, the Rapid Spanning Tree Protocol (RSTP), which is based on IEEE 802.1w, is automatically enabled. The RSTP provides rapid convergence of the spanning tree through explicit handshaking that eliminates the IEEE 802.1D forwarding delay and quickly transitions root ports and designated ports to the forwarding state.

Principle Description

N/A

3.15.2 Configuration

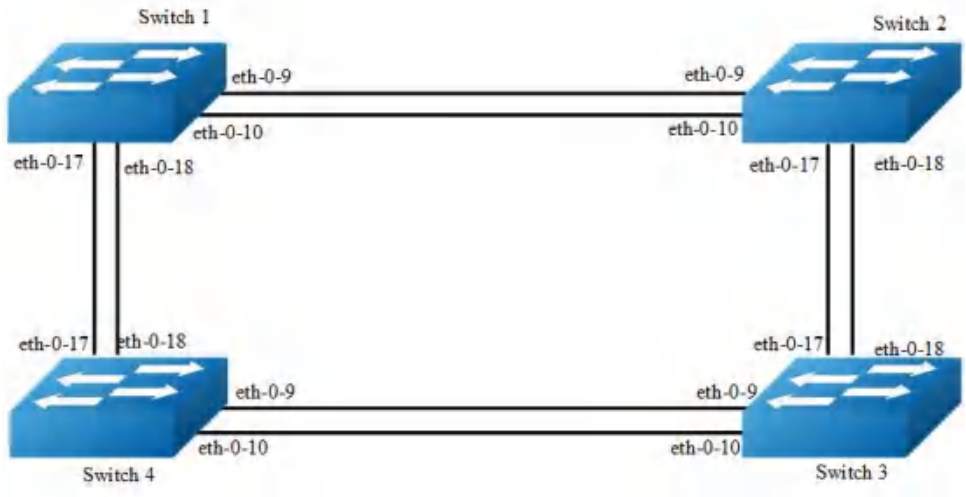


Figure 3-34 MSTP

The configurations of Switch1-Switch4 are as blow. The configurations of these 4 Switches are same if there is no special description.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the mode of STP

```
Switch(config)# spanning-tree mode mstp
```

step 3 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# vlan 20
Switch(config-vlan)# exit
```

step 4 Enter the MSTP configure mode，create region and instance. Bind the vlan to the instance.

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# region RegionName
Switch(config-mst)# instance 1 vlan 10
Switch(config-mst)# instance 2 vlan 20
Switch(config-mst)# exit
```

step 5 Enter the interface configure mode, set the attributes of the interfaces

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-18
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 6 Enable STP and set priority for each swicth

Switch1:

```
Switch# configure terminal
Switch(config)# spanning-tree priority 0
Switch(config)# spanning-tree enable
```

Switch2:

```
Switch# configure terminal
Switch(config)# spanning-tree instance 1 priority 0
Switch(config)# spanning-tree enable
```

Switch3:

```
Switch# configure terminal
Switch(config)# spanning-tree instance 2 priority 0
Switch(config)# spanning-tree enable
```

Switch4:

```
Switch# configure terminal
Switch(config)# spanning-tree enable
```

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

Use the following command to display the information of MSTP on Switch1:

```
Switch# show spanning-tree mst brief
##### MST0:  Vlans: 1
Multiple spanning tree protocol Enabled
Root ID    Priority      0 (0x0000)
           Address     2225.fa28.c900
           Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
Bridge ID  Priority      0 (0x0000)
           Address     2225.fa28.c900
           Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
           Aging Time  300 sec
Interface   Role        State         Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9     Designated    Forwarding     20000        128.9      P2p
eth-0-10    Designated    Forwarding     20000        128.10     P2p
eth-0-17    Designated    Forwarding     20000        128.17     P2p
```

```
eth-0-18   Designated    Forwarding     20000        128.18     P2p
##### MST1:  Vlans: 10
Root ID    Priority     1 (0x0001)
        Address    9c9a.7d91.9f00
Bridge ID  Priority   32769 (0x8001)
        Address    2225.fa28.c900
Interface   Role        State         Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Rootport      Forwarding     20000        128.9       P2p
eth-0-10   Alternate     Discarding     20000        128.10      P2p
eth-0-17   Designated    Forwarding     20000        128.17      P2p
eth-0-18   Designated    Forwarding     20000        128.18      P2p
##### MST2:  Vlans: 20
Root ID    Priority     2 (0x0002)
        Address    304c.275b.b200
Bridge ID  Priority   32770 (0x8002)
        Address    2225.fa28.c900
Interface   Role        State         Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Alternate     Discarding     20000        128.9       P2p
eth-0-10   Alternate     Discarding     20000        128.10      P2p
eth-0-17   Rootport      Forwarding     20000        128.17      P2p
eth-0-18   Alternate     Discarding     20000        128.18      P2p
```

Use the following command to display the information of MSTP on Switch2:

```
Switch# show spanning-tree mst brief
##### MST0:  Vlans: 1
Multiple spanning tree protocol Enabled
Root ID    Priority     0 (0x0000)
        Address    2225.fa28.c900
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
Bridge ID  Priority   32768 (0x8000)
        Address    9c9a.7d91.9f00
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
        Aging Time  300 sec
Interface   Role        State         Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Rootport      Forwarding     20000        128.9       P2p
eth-0-10   Alternate     Discarding     20000        128.10      P2p
eth-0-17   Designated    Forwarding     20000        128.17      P2p
eth-0-18   Designated    Forwarding     20000        128.18      P2p
##### MST1:  Vlans: 10
Root ID    Priority     1 (0x0001)
        Address    9c9a.7d91.9f00
Bridge ID  Priority     1 (0x0001)
        Address    9c9a.7d91.9f00
Interface   Role        State         Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Designated    Forwarding     20000        128.9       P2p
eth-0-10   Designated    Forwarding     20000        128.10      P2p
eth-0-17   Designated    Forwarding     20000        128.17      P2p
eth-0-18   Designated    Forwarding     20000        128.18      P2p
##### MST2:  Vlans: 20
Root ID    Priority     2 (0x0002)
        Address    304c.275b.b200
```

```
Bridge ID  Priority   32770 (0x8002)
        Address    9c9a.7d91.9f00
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Designated    Forwarding     20000        128.9       P2p
eth-0-10   Designated    Forwarding     20000        128.10      P2p
eth-0-17   Rootport      Forwarding     20000        128.17      P2p
eth-0-18   Alternate     Discarding     20000        128.18      P2p
```

Use the following command to display the information of MSTP on Switch3:

```
Switch# show spanning-tree mst brief
##### MST0:  Vlans: 1
Multiple spanning tree protocol Enabled
Root ID    Priority     0 (0x0000)
        Address    2225.fa28.c900
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
Bridge ID  Priority   32768 (0x8000)
        Address    304c.275b.b200
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
        Aging Time  300 sec
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Rootport      Forwarding     20000        128.9       P2p
eth-0-10   Alternate     Discarding     20000        128.10      P2p
eth-0-17   Alternate     Discarding     20000        128.17      P2p
eth-0-18   Alternate     Discarding     20000        128.18      P2p
##### MST1:  Vlans: 10
Root ID    Priority     1 (0x0001)
        Address    9c9a.7d91.9f00
Bridge ID  Priority   32769 (0x8001)
        Address    304c.275b.b200
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Designated    Forwarding     20000        128.9       P2p
eth-0-10   Designated    Forwarding     20000        128.10      P2p
eth-0-17   Rootport      Forwarding     20000        128.17      P2p
eth-0-18   Alternate     Discarding     20000        128.18      P2p
##### MST2:  Vlans: 20
Root ID    Priority     2 (0x0002)
        Address    304c.275b.b200
Bridge ID  Priority     2 (0x0002)
        Address    304c.275b.b200
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Designated    Forwarding     20000        128.9       P2p
eth-0-10   Designated    Forwarding     20000        128.10      P2p
eth-0-17   Designated    Forwarding     20000        128.17      P2p
eth-0-18   Designated    Forwarding     20000        128.18      P2p
```

Use the following command to display the information of MSTP on Switch4:

Switch# show spanning-tree mst brief

```
Switch# show spanning-tree mst brief
##### MST0:  Vlans: 1
Multiple spanning tree protocol Enabled
Root ID    Priority     0 (0x0000)
```

```
        Address    2225.fa28.c900
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
Bridge ID  Priority   32768 (0x8000)
        Address    80a4.be55.6400
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
        Aging Time  300 sec
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Designated    Forwarding     20000        128.9      P2p
eth-0-10   Designated    Forwarding     20000        128.10     P2p
eth-0-17   Rootport      Forwarding     20000        128.17     P2p
eth-0-18   Alternate     Discarding     20000        128.18     P2p
##### MST1:  Vlans: 10
Root ID    Priority     1 (0x0001)
        Address    9c9a.7d91.9f00
Bridge ID  Priority   32769 (0x8001)
        Address    80a4.be55.6400
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Alternate     Discarding     20000        128.9      P2p
eth-0-10   Alternate     Discarding     20000        128.10     P2p
eth-0-17   Rootport      Forwarding     20000        128.17     P2p
eth-0-18   Alternate     Discarding     20000        128.18     P2p
##### MST2:  Vlans: 20
Root ID    Priority     2 (0x0002)
        Address    304c.275b.b200
Bridge ID  Priority   32770 (0x8002)
        Address    80a4.be55.6400
Interface   Role       State        Cost     Priority.Number    Type
--------------------------------------------------------------------------------
eth-0-9    Rootport      Forwarding     20000        128.9      P2p
eth-0-10   Alternate     Discarding     20000        128.10     P2p
eth-0-17   Designated    Forwarding     20000        128.17     P2p
eth-0-18   Designated    Forwarding     20000        128.18     P2p
```

3.15.3 Application cases

N/A

3.16 Configuring MLAG

3.16.1 Overview

Function Introduction

In the network topology of high availability data center, typically TOR switch or server connects to two aggregative switches in order to provide redundancy protection and load sharing. In this situation, Spanning Tree Protocol (STP) can prevent frame loops by blocking half of ports on aggregative switches; however, it will reduce usage of network bandwidth in half; although using MSTP can improve bandwidth utilization to a certain extent, it increases the complexity of the network and is not conducive to operation and problem location.

Using MLAG (Multi-Chassis Link Aggregation) can solve this problem. MLAG is a virtualization technology that presents two different switches as a unique node to establish aggregation-link with the same terminal or device. Between two switches, there is one peer-link that is connected to make the two switches look like one device logically. Ports on these two devices generate aggregative ports to make all ports can participate with forwarding data traffic. Thus MLAG brings the reliability from the link level to the device level by forming an Active-Active system.

Background

Compared with stacking, although MLAG devices still need to be managed separately; MLAG device is simple enough to take lower risk of split-brain. Known unicast traffic forwarding chooses local agg port other than MLAG peer's to prevent too much traffic accross the peer-linkto avoid the bandwidth lack of the conection and lower network latency.

MLAG provides the following technical benefits:

- Provides higher bandwidth and reliability links as network traffic increases.
- MPAG provide a layer2 topology to support redundant backup without loop. It does not need to configure STP and can simplify the network and configuration.
- Supports static LAG or LACP to connect to other switch or servers instead of other protocols.
- Supports active-active Layer-2 redundancy
- Supports upgrade the devices separately. Any one device can support normal business operations.

Principle Description

Basic principle

Terminology

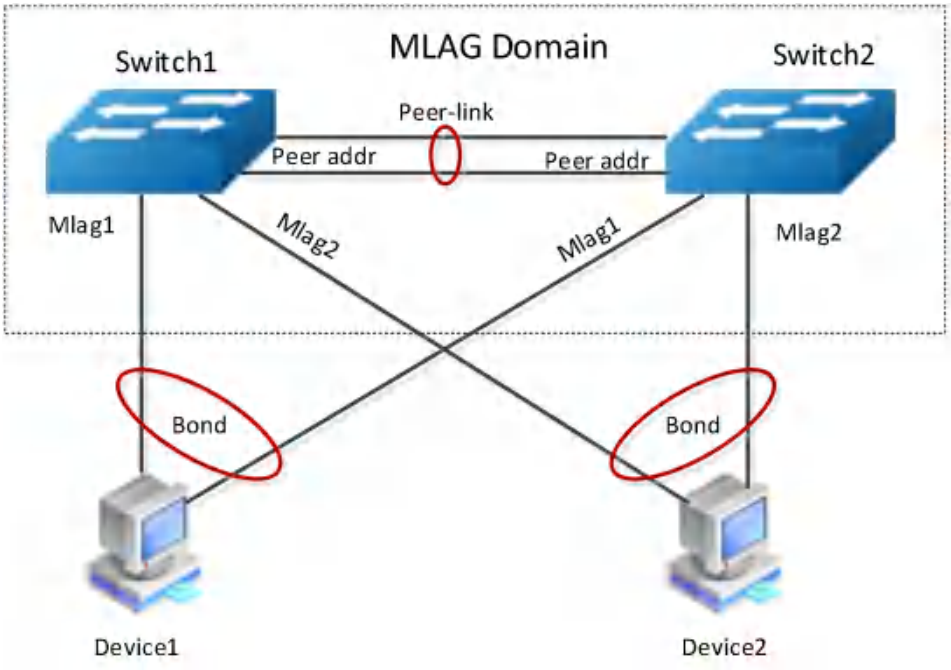


Figure 3-35 MLAG topology sketch

The figure above is the diagram that shows MLAG network. In the diagram, the MLAG domain contains two switches; the device which connects with can be a server or a switch, or another MLAG domain. Switch 1 and Switch 2 each has two ports that join into different MLAG groups; relevant terminologies are shown in the following:

- **MLAG**: Multi-Chassis Link Aggregation; two devices forms into a MLAG called MLAG Peer
- **MLAG group**: interfaces that join the same MLAG group, will be seen as the same aggregation port by external device
- **Orphan Port**: is the interfaces that are not join into MLAG group on MLAG devices
- **Peer-link**: used for interconnecting two devices which forms into a MLAG; while port from one side of MLAG group has failure, the traffic will pass through peer-link to forward by the MLAG Peer device.
- **Peer-address**: communication address of the MLAG protocol on the MLAG Peer device
- **Reload-delay**: after restarting the device, non-peer-link port will set to errdisable status for a period to prevent network looping before MLAG neighbors established; when this timer timeout, these ports will be in admin up status automatically (this timer is configurable).

- **System ID**: default is the Route-MAC address of the device. System ID is used for MLAG master and slave role election. (It is different from LACP system-id).

Forwarding principle(Normal Forwarding）

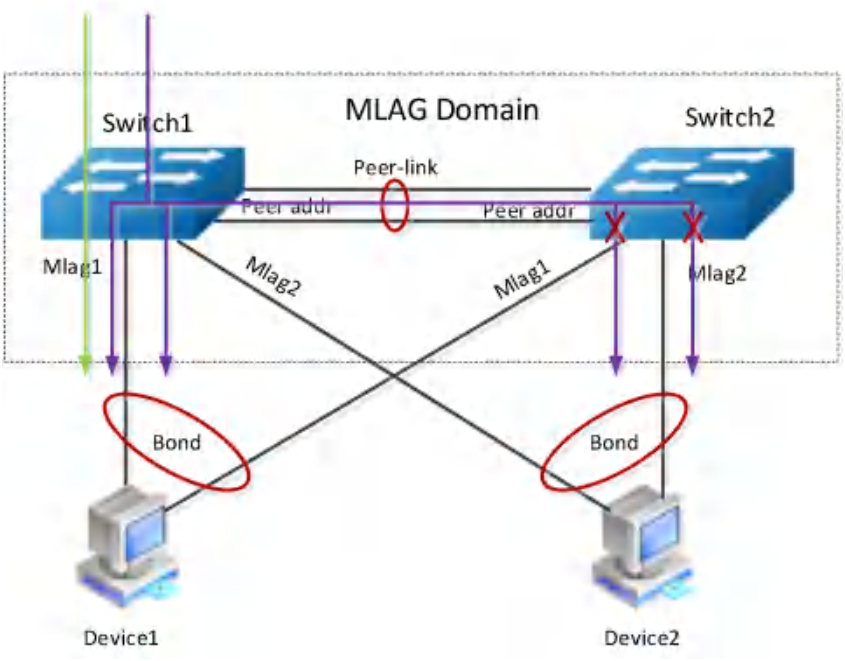


Figure 3-36 MLAG traffic forwarding sketch

As shown in Figure above, the following explains the process of establishing MLAG and its principle of forwarding:

- Switch 1 and Switch 2 create a TCP connection by peer-address, exchange MLAG protocol messages, and establish MLAG neighborhood
- Taking election of Master/Slave by comparing the size of System IDs which are carried from MLAG protocol messages, select the larger System ID as Master by default; MLAG will use Master System ID as the LACP System MAC of the whole system.
- After creating neighbors, keepalive messages are sent between devices and enable timeout mechanism; if it does not receive any keepalive messages within the hold time, then it will consider that the MLAG neighbor disconnects and the device will return to single running status.
- The corresponding aggregative ports of two devices will be binding into the same MLAG group
- When the binding aggregative port in a MLAG group learned some MAC addresses, the device will sync these MACs to peer device, so that the mac address table on both devices will maintain synchronization.
- For known unicast traffic, it will check the local MAC address table to conduct forwarding since two sides both having MAC address tables. The green lines in the figure show the traffic entering from Switch 1, looking up Switch 1's mac address table and forwarding; the traffic entering into Switch 2 is not shown in the figure, it has the same forwarding principles as Switch 1.
- For unknown unicast or broadcast traffic will need to flood, like it shows in blue line; it will pass through PEER-LINK to flood into MLAG PEER device, Switch 2 will see that MLAG 1 and 2 all have UP status member on Switch 1 according to MLAG protocol. Therefore, Switch 2 will discard this part of traffic to prevent flow loopback and double flow problems.

Link failure protection (MLAG interface failure)

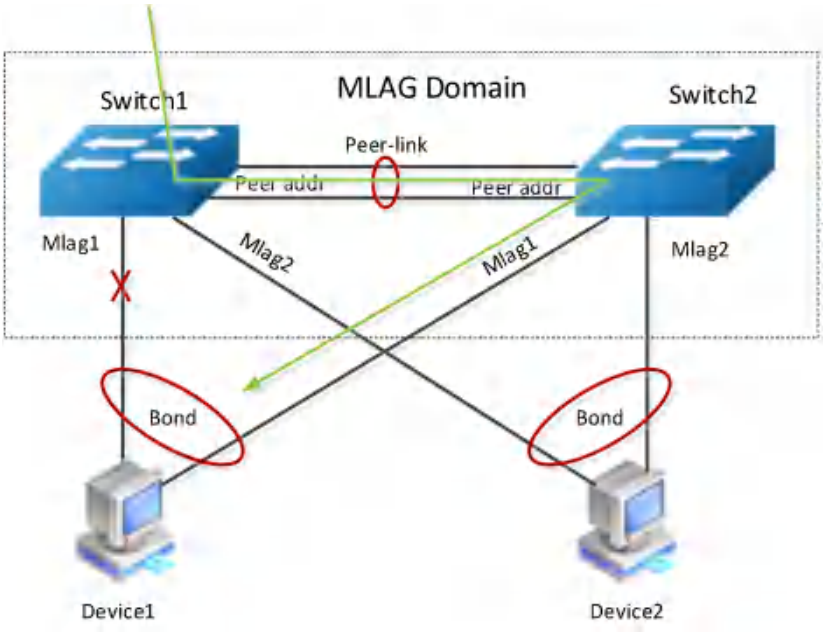


Figure 3-37 Protection switching to unicast messages from MLAG

Shown as above, when MLAG 1 interface expired on Switch 1, to protect unicast traffic forwarding, the system will do as the following:

- When all ports within one MLAG group are losing efficacy on Switch 1, conducting update operations for MAC table, leading data traffic to peer link
- Data traffic will reach to Switch 2 from peer link to continue lookup MAC table; and forwarding from its corresponding MLAG group from Switch 2
- When MLAG port which lose efficacy restores again on Switch 1, system will operate SWITCH 2ack
- This switching only affects the traffic on the expired interface, and will not affect other traffic; in the figure, it will only influence the traffic to MLAG 1 on Switch 1, the traffic to MLAG2 will forward normally.

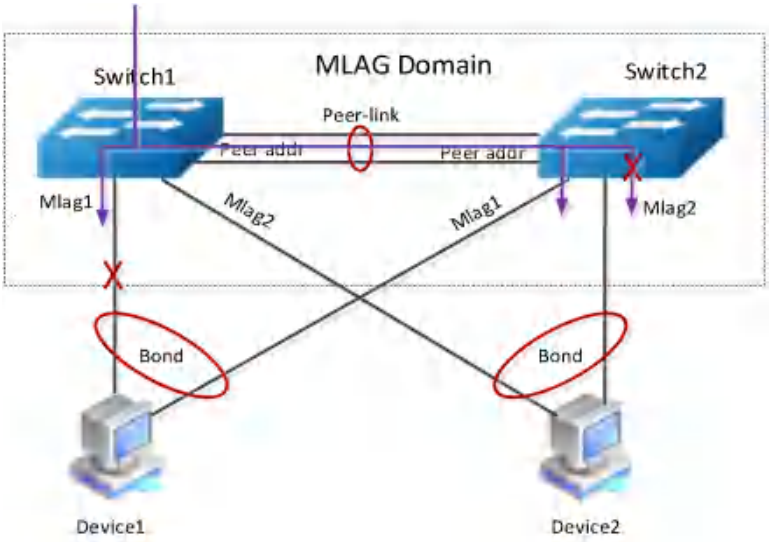


Figure 3-38 MLAG Protection of flood traffic

forwarding flood traffic, system will do the followings:

- When Switch 1 finds that local MLAG 1 interface is expired, it will notify Switch 2 by MLAG protocol; Switch 2 will not discard traffics to MLAG 1 anymore after it receiving messages from Switch 1.
- When MLAG port restore again from expired status, Switch 1 will notify Switch 2 using MLAG protocol; Switch 2 will discard traffics to MLAG 1 again.

Link failure protection (Peer link failure)

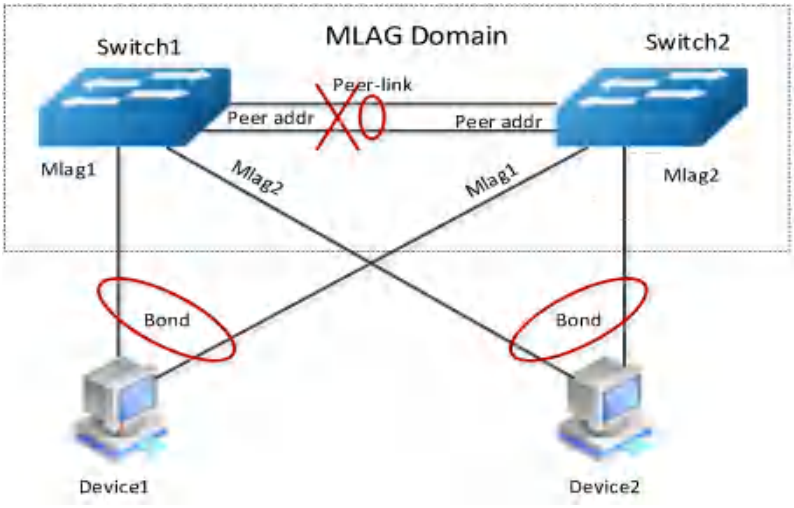


Figure 3-39 Peer-link failure protection

- When Peer link disconnect and the hold time is timed out, MLAG device will be divided into two single switches.
- If Device 1 and Device 2 are using LACP aggregative connection with MLAG device, when Peer link expired and MLAG divides, Switch 1 and Switch 2 will use different ID independently. Hence, there is only one link path is active, so there is no risk of looping.
- If Device 1 and Device 2 are connecting with MLAG device using static link aggregation, the two links are still in active; since members cannot forward traffic within the same aggregative group, it will not cause looping.

Surrounding Features

LACP with MLAG

To realize LACP for MLAG, we do the followings:

- LACP messages that are sent by two MLAG devices on MLAG interface has the same Actor System Priority and Actor System ID
- LACP messages that are sent by two MLAG devices on MLAG interface has the same Actor Key

MLAG device use the same LACP System id as the master. (LACP System id includes LACP System Priority and LACP System MAC). Master device should synchronize the LACP System id with the slave device.

MLAG slave device should use the LACP System MAC after MLAG established. If the MLAG master goes down or reboot, the MLAG session is down and the slave device should switch to its own LACP System MAC， this event will lead LACP protocol negotiate again and LACP link changes from up to down, then up again after negotiation succeed. To enhance the reliability in this scene, our device provides a command to configure MLAG LACP System MAC. After this command is configured, all MLAG interfaces can use same LACP system MAC.

LACP System id is conceps of LACP protocol, which are used for indicating a device in lacp protocol. It is recommend to use LACP mode aggregation to avoid unidirection link. Static aggregation is used only if the remote device does not support LACP.

ISSU with MLAG

MLAG device will use Error-disable mechanism to realize ISSU. When it restarts after devices finishing upgrading, the devices that have configured MLAG will make the local ports to Error-disable for a period. MLAG will re-negotiation and synchronize tables, and calculate STP during this period of time; when the timer times out, ports will restore normal. This period of time can be configured by "reload-delay" command; the default value is 300 seconds.

The port in errdisable status is similar to the port being "down" status, it will not participate in forwarding; it can restore by command "no shutdown", and using "show errdisable recovery" to check the status.

VARP with MLAG

System will provide global configuration "virtual mac" and port configuration "virtual ip", to give the uniform network addresses including ip address and mac address to opposite terminal that is connecting with MLAG device. Whatever which device received this kinds of messages of virtual ip and virtual mac, they all can handling correctly on local. Virtual ip can be in the same network section as the interface ip or it can be in different section; each interface can configure up to 15 virtual ip. A switch can only support one Virtual MAC entirely.

Virtual mac needs to setup a mac address that is nonexistent at local; the address cannot be the same as the route-mac of device or the mac address of interface.

Dual-homed Server and MLAG

To those dual-homed servers, they set the bond mode 4 usually; in this condition, the switch needs to configure lacp aggregation mode and add the agg port to mlag group, using "channel-group" command to setup lacp aggregation mode.

When the bond mode is configured as 0 or 2, the switch needs to configure static port aggregation, and the aggregated ports are added to the mlag group; it does not need to setup mlag group for other bond mode.

3.16.2 Restrictions and Precautions

- The two switches that are forming into MLAG should be the switches from our company; it cannot mix combine with switches from other companies to form into MLAG.
- Peer-address only supports directly connect segments address.
- Ports in errdisable status are same as ports in shutdown status, they will not forward traffic, they will automatically recover on timer expiry if the reason is "reload-delay"; or system administrator can check with "show errdisable recovery" command and recover them by using "shutdown/no shutdown" command.
- MLAG does not support multicast table entry synchronization and multicast data traffic will be flooded and forwarded in a multicast environment. MLAG with multicast application scenarios is not recommended. It is recommended to disable the igmp snooping function on MLAG devices. (Igmp snooping is enabled by default)
- Enable STP MLAG devices should enable e-stp.

3.16.3 Configuration

Basic configuration

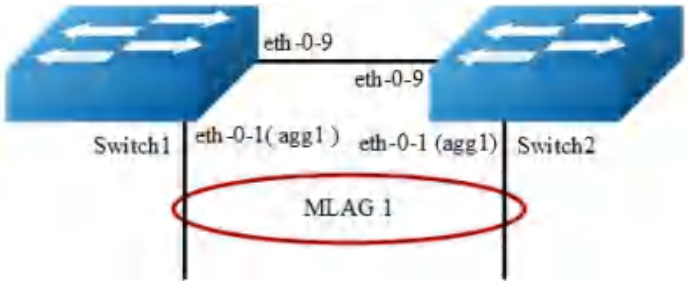


Figure 3-40 MLAG

The configurations of Switch1-Switch2 are as blow. The configurations of these 2 Switches are same if there is no special description.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,4094
Switch(config-vlan)# exit
```

step 3 Create a static agg

```
Switch(config)# interface eth-0-1
Switch(config-if)# static-channel-group 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Set the attributes of the peer link interface

interface eth-0-9 will be set as the peer link interface later

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# spanning-tree port disable
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 5 Bind the agg interface to the mlag

```
Switch(config)# interface agg1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# mlag 1
Switch(config-if)# exit
```

step 6 Set the attributes of the vlan interface

Switch1：

```
Switch(config)# interface vlan4094
Switch(config-if)# ip address 12.1.1.1/24
Switch(config-if)# exit
```

Switch2：

```
Switch(config)# interface vlan4094
Switch(config-if)# ip address 12.1.1.2/24
Switch(config-if)# exit
```

step 7 Enter the mlag configure mode and set the attributes of the mlag

Switch1：

```
Switch(config)# mlag configuration
Switch(config-mlag)# peer-link eth-0-9
Switch(config-mlag)# peer-address 12.1.1.2
Switch(config-mlag)# exit
```

Switch2：

```
Switch(config)# mlag configuration
Switch(config-mlag)# peer-link eth-0-9
Switch(config-mlag)# peer-address 12.1.1.1
Switch(config-mlag)# end
```

step 8 Validation

Use the following command to display the information of mlag on Switch1

```
Switch# show mlag
MLAG configuration:
```

```
-----------------
role      : Master
local_sysid : ea90.aecc.cc00
mlag_sysid  : ea90.aecc.cc00
peer-link   : eth-0-9
peer conf   : Yes

Switch# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1     up          up

Switch# show mlag peer
MLAG neighbor is 12.1.1.2, MLAG version 1
MLAG state = Established, up for 00:13:07
Last read 00:00:48, hold time is 240, keepalive interval is 60 seconds
Received 17 messages,Sent 19 messages
Open    : received 1, sent 2
KAlive   : received 15, sent 16
Fdb sync : received 0, sent 0
Failover : received 0, sent 0
Conf    : received 1, sent 1

Connections established 1; dropped 0
Local host: 12.1.1.1, Local port: 61000
Foreign host: 12.1.1.2, Foreign port: 46157
remote_sysid: baa7.8606.8b00



Switch# show mac address-table
       Mac Address Table
-------------------------------------------
(*) - Security Entry
Vlan   Mac Address      Type       Ports
----   -----------      --------   -----
```

Use the following command to display the information of mac address table on Switch2

```
Switch# show mlag
MLAG configuration:
-----------------
role      : Slave
local_sysid : baa7.8606.8b00
mlag_sysid  : ea90.aecc.cc00
peer-link   : eth-0-9
peer conf   : Yes

Switch# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1     up          up

Switch# show mlag peer
MLAG neighbor is 12.1.1.1, MLAG version 1
MLAG state = Established, up for 00:14:29
Last read 00:00:48, hold time is 240, keepalive interval is 60 seconds
Received 19 messages,Sent 19 messages
```

```
Open    : received 1, sent 1
KAlive  : received 17, sent 17
Fdb sync : received 0, sent 0
Failover : received 0, sent 0
Conf    : received 1, sent 1

Connections established 1; dropped 0
Local host: 12.1.1.2, Local port: 46157
Foreign host: 12.1.1.1, Foreign port: 61000
remote_sysid: ea90.aecc.cc00
```

Use the following command to display the information of mlag on Switch2:

```
Switch# show mac address-table
      Mac Address Table
-------------------------------------------
(*) - Security Entry
Vlan   Mac Address     Type      Ports
----   -----------     --------  -----
```

Enable MLAG and xSTP

topology

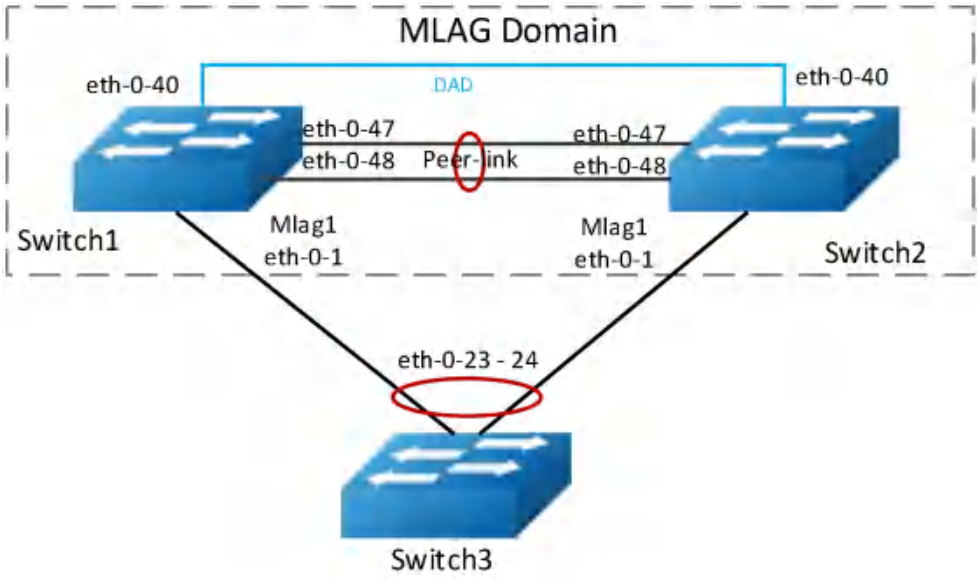


Figure 3-41 MLAG+RSTP

The MLAG domain operates as part of the network alongside other devices in the network, running RSTP to prevent network storms caused by loops in case of network abnormalities. The diagram above shows the network topology related to MLAG.

Configuration Steps

First, complete the configuration of Switch1 and Switch2.
Enter configuration mode.

```
Switch1# configure terminal
```

Enter VLAN configuration mode and create VLANs.

Configure Switch1, and configure Switch2 identically to Switch1.

```
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10,4094
Switch1(config-vlan)# exit
```

Configure the properties of the aggregated interface agg63.

Configure Switch1, and configure Switch2 identically to Switch1.

```
Switch1(config)# interface range eth-0-47 - 48
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan all
Switch1(config-if)# channel-group 63 mode active
Switch1(config-if)# exit
```

Configure the MLAG interface.

Configure Switch1, with eth-0-1 as a member port of MLAG 1. Switch2 should be configured the same as Switch1.

```
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport access vlan 10
Switch1(config-if)# channel-group 1 mode active
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
```

Add the aggregated interface to MLAG.

Configure Switch1, and configure Switch2 identically to Switch1.

```
Switch1(config)# interface agg1
Switch1(config-if)# mlag 1
Switch1(config-if)# exit
```

Configure the properties of the VLAN interface.

Configure Switch1.

```
Switch1(config)# interface vlan4094
Switch1(config-if)# ip address 172.20.1.1/24
Switch1(config-if)# exit
```

Configure Switch2.

```
Switch2(config)# interface vlan4094
Switch2(config-if)# ip address 172.20.1.2/24
Switch2(config-if)# exit
```

Configure Layer 3 interfaces for DAD.

Configure Switch1.

```
Switch1(config)# interface eth-0-40
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.77.1/24
Switch1(config-if)# exit
```

Configure Switch2.

```
Switch2(config)# interface eth-0-40
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.77.2/24
Switch2(config-if)# exit
```

Enter MLAG mode and configure the relevant properties.

Configure Switch1 as the MLAG Master and assign it a higher priority.

```
Switch1(config)# mlag configuration
Switch1(config-mlag)# priority 254
Switch1(config-mlag)# peer-link agg63
Switch1(config-mlag)# peer-address 172.20.1.2
Switch1(config-mlag)# dual-active-detection source 192.168.77.1 peer 192.168.77.2
Switch1(config-mlag)# exit
```

Configure Switch2.

```
Switch2(config)# mlag configuration
Switch2(config-mlag)# priority 200
Switch2(config-mlag)# peer-link agg63
Switch2(config-mlag)# peer-address 172.20.1.1
Switch2(config-mlag)# dual-active-detection source 192.168.77.2 peer 192.168.77.1
Switch2(config-mlag)# exit
```

Configure the MLAG LACP system ID.

Use Switch1's MAC address (0001.0203.0406) as the system ID for MLAG domain members to negotiate LACP with other devices. Configure both Switch1 and Switch2 with the same setting.

```
Switch1(config)# lacp mlag system-id 0001.0203.0406
```

Configure Spanning Tree.

The default mode is RSTP, and no changes are needed. STP priority can be configured to control root bridge selection. In this case, we will configure the MLAG devices to act as the root bridge. Apply the same configuration on both Switch1 and Switch2.

```
Switch1(config)# spanning-tree priority 0
Switch1(config)# spanning-tree e-stp enable
Switch1(config)# spanning-tree enable
```

Complete the configuration on Switch3.

```
Switch3# configure terminal
Switch3(config)# vlan database
Switch3(config-vlan)# vlan 10
Switch3(config-vlan)# exit
Switch3(config)# interface range eth-0-23 - 24
Switch3(config-if)# switchport access vlan 10
Switch3(config-if)# channel-group 1 mode active
Switch3(config-if)# exit
Switch3(config)# spanning-tree enable
```

Verification

Display the MLAG status on Switch1.

```
Switch1# show mlag
MLAG configuration:
-----------------
role               : Master
local_dev_pri      : 254
remote_dev_pri     : 200
local_sysid        : 0001.0203.0406
remote_sysid       : 0001.0203.0d02
mlag_sysid         : 0001.0203.0406
```

```
local_lacppri          : 32768
remote_lacppri         : 32768
mlag_lacppri           : 32768
peer-link              : agg63
peer conf              : Yes
reload-delay           : Auto(300s)
arp sync status
 interface vlan1       : local:unavailable(no_valid_mlag_id), remote:unavailable(no_valid_mlag_id)
Switch1#
```

Display the MLAG status on Switch2.

```
Switch2# show mlag
MLAG configuration:
-----------------
role                   : Slave
local_dev_pri          : 200
remote_dev_pri         : 254
local_sysid            : 0001.0203.0d02
remote_sysid           : 0001.0203.0406
mlag_sysid             : 0001.0203.0406
local_lacppri          : 32768
remote_lacppri         : 32768
mlag_lacppri           : 32768
peer-link              : agg63
peer conf              : Yes
reload-delay           : Auto(300s)
arp sync status
 interface vlan1       : local:unavailable(no_valid_mlag_id), remote:unavailab
le(no_valid_mlag_id)
Switch2#
```

Display the e-stp status on Switch1.

```
Switch1# show spanning-tree e-stp
 E-STP enabled
 Active Bridge Id 0000000102030406
 Peer-link agg63

 Bridge up - Spanning Tree Enabled
 Mode - Rapid spanning tree protocol
 Path Cost Standard - dot1t
 Root Path Cost 0 - Root Port 0 -  Bridge Priority 0
 Forward Delay 15 - Hello Time 2 - Max Age 20
 Tx Hold Count 6
 Root Id 0000000102030406
 Bridge Id 0000000102030406
 Last topology change Fri Jan 30 04:14:16 2026
 Edgeport bpdu-filter disabled
 Edgeport bpdu-guard disabled
 CIST Best Estp Sysmac 0001.0203.0406
 CIST Self Estp Sysmac 0001.0203.0406
 mlag 1: port 4097 - Id 8001 - Role Designated - State Forwarding

Switch1#
```

Display the e-stp status on Switch2.

```
Switch2# show spanning-tree e-stp
 E-STP enabled
 Active Bridge Id 0000000102030406
 Peer-link agg63

 Bridge up - Spanning Tree Enabled
 Mode - Rapid spanning tree protocol
 Path Cost Standard - dot1t
 Root Path Cost 0 - Root Port 4159 -  Bridge Priority 0
 Forward Delay 15 - Hello Time 2 - Max Age 20
 Tx Hold Count 6
 Root Id 0000000102030406
 Bridge Id 0000000102030406
 Last topology change Fri Jan 30 16:20:30 2026
 Edgeport bpdu-filter disabled
 Edgeport bpdu-guard disabled
 CIST Best Estp Sysmac 0001.0203.0406
 CIST Self Estp Sysmac 0001.0203.0d02
 mlag 1: port 4097 - Id 8001 - Role Designated - State Forwarding

Switch2#
```

Display the STP port status on Switch1.

```
Switch1# show spanning-tree brief
Rapid spanning tree protocol Enabled
Root ID    Priority     0 (0x0000)
        Address    0001.0203.0406
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec

Bridge ID  Priority     0 (0x0000)
        Address    0001.0203.0406
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
        Aging Time  300 sec

Interface  Role        State         Cost     Priority.PortNumber  Type
--------------------------------------------------------------------------------
mlag1     Designated    Forwarding      2000         128.4097       P2p
Switch1#
```

Display the STP port status on Switch2.

```
Switch2# show spanning-tree brief
Rapid spanning tree protocol Enabled
Root ID    Priority     0 (0x0000)
        Address    0001.0203.0406
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec

Bridge ID  Priority     0 (0x0000)
        Address    0001.0203.0406
        Hello Time  2 sec  Max Age  20 sec Forward Delay 15 sec
        Aging Time  300 sec

Interface  Role        State         Cost     Priority.PortNumber  Type
--------------------------------------------------------------------------------
mlag1     Designated    Forwarding      2000         128.4097       P2p
Switch2#
```

Note:

- When using MLAG and xSTP simultaneously, E-STP must be enabled, and STP must not be disabled on the peer-link port。
- After enabling E-STP, it is recommended to keep the global STP parameters on both MLAG devices and the STP parameters on the aggregated interfaces bound to MLAG consistent; otherwise, STP convergence may be affected. The peer-link interface and aggregated interfaces bound to MLAG should not have STP link cost, STP interface priority, or STP link type configured.
- If an interface already has STP link cost, STP interface priority, or STP link type configured, these settings will revert to default values when the interface is set as a peer-link or bound to MLAG. The system will log this event.
- Both MLAG members should be configured with the same LACP MLAG system ID to avoid brief traffic interruption caused by LACP renegotiation when the Master fails.

Layer 2 Forwarding Protection

Topology

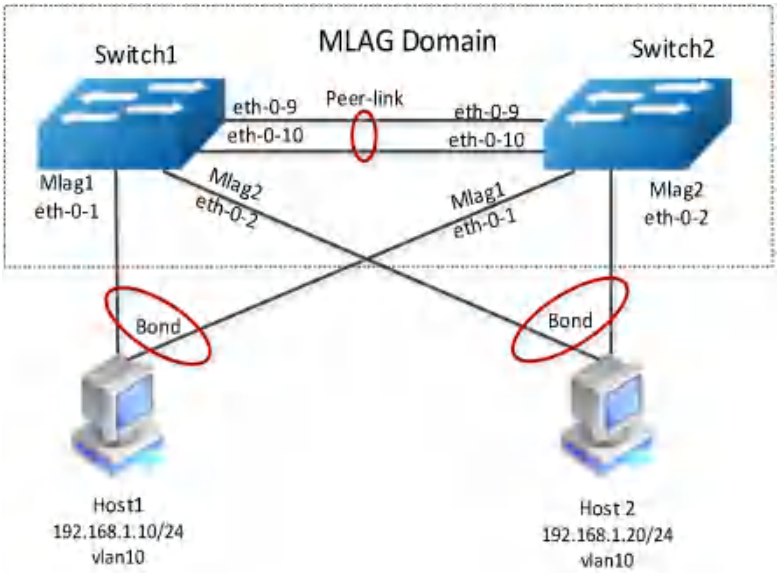


Figure 3-42 MLAG

Requirement

Host 1 and Host 2 are in the same network section, its dual network interface cards (NIC) use active-active method to connect into MLAG device, and it needs network without looping, Host 1 and Host 2 realize Layer 2 interflow. MLAG1 will use dynamic aggregation link, MLAG 2 will use static aggregation link.

Configuration Steps

Note: The channel-group 55 used to interconnect the two switches in the example configuration can be modified according to different standards of boards, use any channel-group serial number will not affect the configuration of this example.

step 1 Configure MLAG PEER Interconnect Ports

Two devices doing MLAG, choose at least two 10G links for interconnection (in the case of service port 10G)

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1 (config)# interface range eth-0-9 – 10
Switch_1 (config-if-range)# no shutdown
Switch_1 (config-if-range)# lacp timeout short
Switch_1 (config-if-range)# channel-group 55 mode active
Switch_1 (config-if-range)# exit
```

```
Switch_1 (config)# interface agg 55
Switch_1 (config-if)# spanning-tree port disable
Switch_1 (config-if)# switchport mode trunk
Switch_1 (config-if)# switchport trunk allowed vlan all
Switch_1 (config-if)#end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface range eth-0-9 – 10
Switch_2 (config-if-range)# no shutdown
Switch_2 (config-if-range)# lacp timeout short
Switch_2(config-if-range)# channel-group 55 mode active
Switch_2(config-if-range)# exit
Switch_2(config)# interface agg 55
Switch_2(config-if)# spanning-tree port disable
Switch_2(config-if)# switchport mode trunk
Switch_2(config-if)# switchport trunk allowed vlan all
Switch_2(config-if)#end
```

step 2 Configure MLAG PEER Communication Address

In the example vlan4094 is used to configure the mlag communication address, it is recommended that no service port join vlan4094, vlan4094 is for peer-link use only.

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 4094
Switch_1(config-vlan)# exit
Switch_1(config)# interface vlan 4094
Switch_1(config-if)# ip address 10.10.0.1/30
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 4094
Switch_2(config-vlan)# exit
Switch_2(config)# interface vlan 4094
Switch_2(config-if)# ip address 10.10.0.2/30
Switch_2(config-if)# end
```

step 3 Configure MLAG PEER

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# mlag configuration
Switch_1(config-mlag)# timers mlag 1 5
Switch_1(config-mlag)# peer-link agg55
Switch_1(config-mlag)# peer-address 10.10.0.2
Switch_1(config-mlag)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# mlag configuration
Switch_2(config-mlag)# timers mlag 1 5
Switch_2(config-mlag)# peer-link agg 55
Switch_2(config-mlag)# peer-address 10.10.0.1
Switch_2(config-mlag)# end
```

step 4 Configure MLAG Port

note： lacp mlag system-id HHHH.HHHH.HHHH is the user-defined LACP system id, two devices in the MLAG should use same LACP system id, for example: 0000.0000.aaaa; When cascade connect two pairs of MLAG devices, the LACP system id should be different for each MLAG pair.

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# no ip igmp snooping
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 10
Switch_1(config-vlan)# exit
Switch_1(config)# lacp mlag system-id 0000.0000.aaaa
Switch_1(config)# interface eth-0-1
Switch_1(config-if)# no shutdown
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# channel-group 1 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface eth-0-2
Switch_1(config-if)# no shutdown
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# static-channel-group 2
Switch_1(config-if)# exit
Switch_1(config)# interface agg 1
Switch_1(config-if)# mlag 1
Switch_1(config-if)# exit
Switch_1(config)# interface agg 2
Switch_1(config-if)# mlag 2
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# no ip igmp snooping
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10
Switch_2(config-vlan)# exit
Switch_2(config)# lacp mlag system-id 0000.0000.aaaa
Switch_2(config)# interface eth-0-1
Switch_2(config-if)# no shutdown
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# channel-group 1 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface eth-0-2
Switch_2(config-if)# no shutdown
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# static-channel-group 2
```

```
Switch_2(config-if)# exit
Switch_2(config)# interface agg 1
Switch_2(config-if)# mlag 1
Switch_2(config-if)# exit
Switch_2(config)# interface agg 2
Switch_2(config-if)# mlag 2
Switch_2(config-if)# end
```

step 5 Verify Configuration Results

Check MLAG neighbor status, MLAG in established state after configuration.

```
Switch_1# show mlag peer
MLAG neighbor is 10.10.0.2, MLAG version 1
MLAG state = Established, up for 00:00:01
Last read 00:00:01, hold time is 240, keepalive interval is 60 seconds
Received 4 messages,Sent 4 messages
Open    : received 1, sent 1
KAlive   : received 1, sent 1
Fdb sync  : received 0, sent 0
Failover  : received 0, sent 0
Conf    : received 0, sent 0
Syspri   : received 1, sent 1
Peer fdb : received 1, sent 1

Connections established 1; dropped 0
Local host: 10.10.0.1, Local port: 50040
Foreign host: 10.10.0.2, Foreign port: 61000
remote_sysid: 1a53.71e9.c000
```

Check MLAG device status, one devices is in Master and another is in Slave state.

```
Switch_1# show mlag
MLAG configuration:
------------------
role       : Master
local_sysid  : 8e79.b120.2e00
remote_sysid : 1a53.71e9.c000
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300

Switch_2# show mlag
MLAG configuration:
------------------
role       : Slave
local_sysid  : 1a53.71e9.c000
remote_sysid : 8e79.b120.2e00
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
```

```
peer conf   : Yes
reload-delay : 300
```

Check MLAG group status, all interfaces should be in UP.

```
Switch_1# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1    up        up
2     agg2    up        up
```

MLAG Device as Layer 3 Network (each VLAN has only one address)

Topology

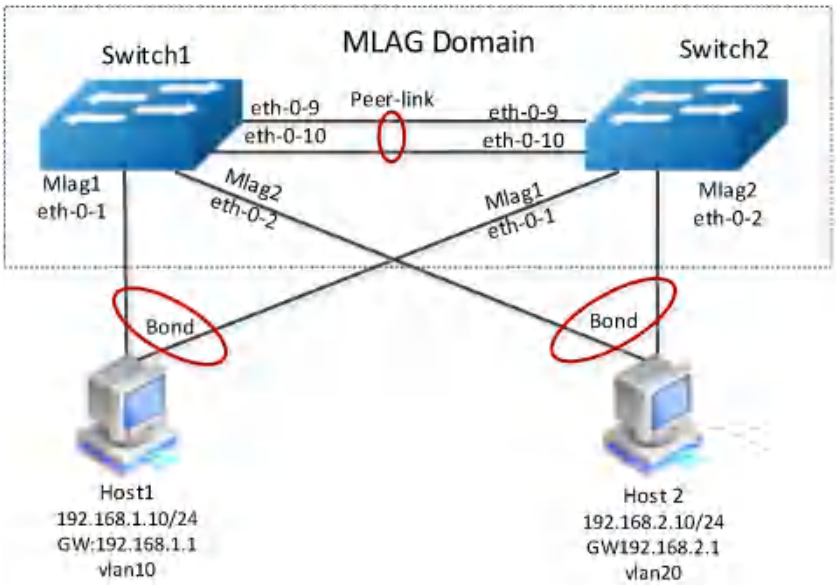


Figure 3-43 Network gateway diagram of MLAG Layer 3 - Single address

Requirement

Host 1 and Host 2 use dual network cards with active-active method to connect to MLAG device, networking across network section; their gateway is on MLAG device, requires gateway active-active and it does not use VRRP.

Configuration Steps

Note: The channel-group 55 used to interconnect the two switches in the example configuration can be modified according to different standards of boards, use any channel-group serial number will not affect the configuration of this example.

step 1 Configure MLAG PEER Interconnect Ports

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1 (config)# interface range eth-0-9 – 10
Switch_1 (config-if-range)# no shutdown
Switch_1 (config-if-range)# lacp timeout short
Switch_1 (config-if-range)# channel-group 55 mode active
Switch_1 (config-if-range)# exit
Switch_1 (config)# interface agg 55
Switch_1 (config-if)# spanning-tree port disable
Switch_1 (config-if)# switchport mode trunk
```

```
Switch_1 (config-if)# switchport trunk allowed vlan all
Switch_1 (config-if)#end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface range eth-0-9 – 10
Switch_2 (config-if-range)# no shutdown
Switch_2 (config-if-range)# lacp timeout short
Switch_2(config-if-range)# channel-group 55 mode active
Switch_2(config-if-range)# exit
Switch_2(config)# interface agg 55
Switch_2(config-if)# spanning-tree port disable
Switch_2(config-if)# switchport mode trunk
Switch_2(config-if)# switchport trunk allowed vlan all
Switch_2(config-if)#end
```

step 2 Configure MLAG PEER Communication Address

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 4094
Switch_1(config-vlan)# exit
Switch_1(config)# interface vlan 4094
Switch_1(config-if)# ip address 10.10.0.1/30
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 4094
Switch_2(config-vlan)# exit
Switch_2(config)# interface vlan 4094
Switch_2(config-if)# ip address 10.10.0.2/30
Switch_2(config-if)# end
```

step 3 Configure MLAG PEER

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# mlag configuration
Switch_1(config-mlag)# timers mlag 1 5
Switch_1(config-mlag)# peer-link agg55
Switch_1(config-mlag)# peer-address 10.10.0.2
Switch_1(config-mlag)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# mlag configuration
Switch_2(config-mlag)# timers mlag 1 5
```

```
Switch_2(config-mlag)# peer-link agg 55
Switch_2(config-mlag)# peer-address 10.10.0.1
Switch_2(config-mlag)# end
```

step 4 Configure MLAG Interface

In a Layer 3 scenario, it is recommended that both the switch service port and the peer port be set to lacp short timeout mode with the command "lacp timeout short".

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# no ip igmp snooping
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 10,20
Switch_1(config-vlan)# exit
Switch_1(config)# interface eth-0-1
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# channel-group 1 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface eth-0-2
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 20
Switch_1(config-if)# channel-group 2 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface agg 1
Switch_1(config-if)# mlag 1
Switch_1(config-if)# exit
Switch_1(config)# interface agg 2
Switch_1(config-if)# mlag 2
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# no ip igmp snooping
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10,20
Switch_2(config-vlan)# exit
Switch_2(config)# interface eth-0-1
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# channel-group 1 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface eth-0-2
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 20
Switch_2(config-if)# channel-group 2 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface agg 1
Switch_2(config-if)# mlag 1
```

```
Switch_2(config-if)# exit
Switch_2(config)# interface agg 2
Switch_2(config-if)# mlag 2
Switch_2(config-if)# end
```

step 5 Configure VARP as a Gateway to Host

note： lacp mlag system-id HHHH.HHHH.HHHH is the user-defined LACP system id, two devices in the MLAG should use same LACP system id, for example: 0000.0000.aaaa; When cascade connect two pairs of MLAG devices, the LACP system id should be different for each MLAG pair.

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# interface vlan 10
Switch_1(config-if)# ip address 192.168.1.253/24
Switch_1(config-if)# ip virtual-router address 192.168.1.1
Switch_1(config-if)# exit
Switch_1(config)# interface vlan 20
Switch_1(config-if)# ip address 192.168.2.253/24
Switch_1(config-if)# ip virtual-router address 192.168.2.1
Switch_1(config-if)# exit
Switch_1(config)# ip virtual-router mac 0000.0000.aaaa
Switch_1(config)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface vlan 10
Switch_2(config-if)# ip address 192.168.1.254/24
Switch_2(config-if)# ip virtual-router address 192.168.1.1
Switch_2(config-if)# exit
Switch_2(config)# interface vlan 20
Switch_2(config-if)# ip address 192.168.2.254/24
Switch_2(config-if)# ip virtual-router address 192.168.2.1
Switch_2(config-if)# exit
Switch_2(config)# ip virtual-router mac 0000.0000.aaaa
Switch_2(config)# end
```

step 6 Verify Configuration Results

Check the status of MLAG neighbor; MLAG will be Established status after configurations.

```
Switch_1# show mlag peer
MLAG neighbor is 10.10.0.2, MLAG version 1
MLAG state = Established, up for 00:00:01
Last read 00:00:01, hold time is 240, keepalive interval is 60 seconds
Received 4 messages,Sent 4 messages
Open    : received 1, sent 1
KAlive   : received 1, sent 1
Fdb sync  : received 0, sent 0
Failover  : received 0, sent 0
Conf    : received 0, sent 0
Syspri   : received 1, sent 1
Peer fdb : received 1, sent 1

Connections established 1; dropped 0
```

```
Local host: 10.10.0.1, Local port: 50040
Foreign host: 10.10.0.2, Foreign port: 61000
remote_sysid: 1a53.71e9.c000
```

Check the status of MLAG device, the two devices will be Master/Slave status.

```
Switch_1# show mlag
MLAG configuration:
-----------------
role       : Master
local_sysid  : 8e79.b120.2e00
remote_sysid : 1a53.71e9.c000
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300

Switch_2# show mlag
MLAG configuration:
-----------------
role       : Slave
local_sysid  : 1a53.71e9.c000
remote_sysid : 8e79.b120.2e00
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300
```

Check the status of MLAG group, all interfaces should be UP status.

```
Switch_1# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1    up          up
2     agg2    up          up
```

Check the status of VARP

```
Switch_1# show ip arp
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   10.10.0.1           -  8e79.b120.2e00  vlan4094
 Internet   10.10.0.2           0  1a53.71e9.c000  vlan4094
 Internet   192.168.1.1         -  0000.0000.0001  vlan10
 Internet   192.168.1.253       -  8e79.b120.2e00  vlan10
 Internet   192.168.2.1         -  0000.0000.0001  vlan20
 Internet   192.168.2.253       -  8e79.b120.2e00  vlan20
```

MLAG Device as Layer 3 Network (each VLAN has multiple addresses)

Topology

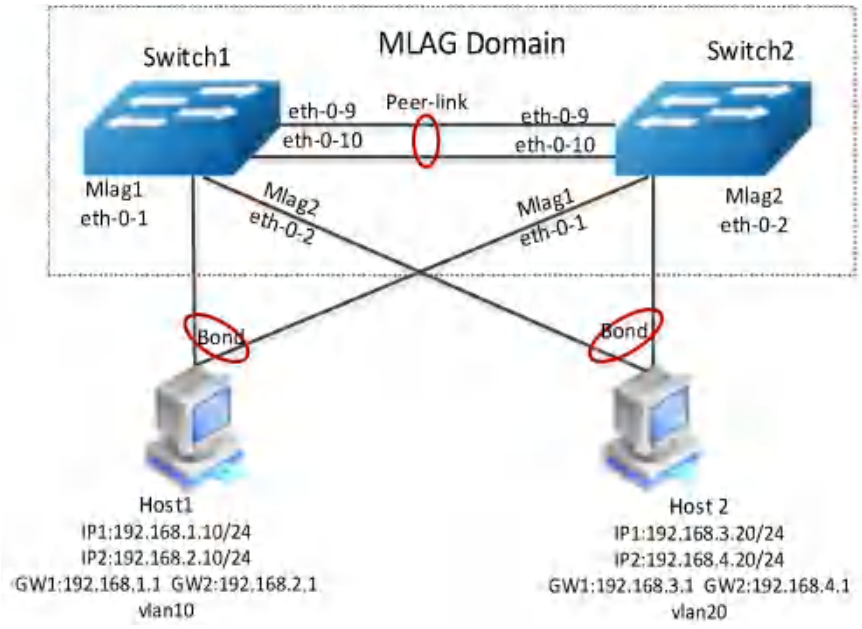


Figure 3-44 Network gateway diagram of MLAG Layer 3 - multiple addresses)

Requirement

Host 1 and Host 2 use dual network cards with active-active method to connect to MLAG device, networking across network section; their network gateway deployment is on MLAG device, and it has multiple ip addresses on the same vlan and switch is the network gateway of these networks.

Configuration Steps

Note: The channel-group 55 used to interconnect the two switches in the example configuration can be modified according to different standards of boards, use any channel-group serial number will not affect the configuration of this example.

step 1 Configure MLAG PEER Interconnect Ports

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1 (config)# interface range eth-0-9 – 10
Switch_1 (config-if-range)# no shutdown
Switch_1 (config-if-range)# lacp timeout short
Switch_1 (config-if-range)# channel-group 55 mode active
Switch_1 (config-if-range)# exit
Switch_1 (config)# interface agg 55
Switch_1 (config-if)# spanning-tree port disable
Switch_1 (config-if)# switchport mode trunk
Switch_1 (config-if)# switchport trunk allowed vlan all
Switch_1 (config-if)#end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface range eth-0-9 – 10
Switch_2 (config-if-range)# no shutdown
Switch_2 (config-if-range)# lacp timeout short
Switch_2(config-if-range)# channel-group 55 mode active
Switch_2(config-if-range)# exit
```

```
Switch_2(config)# interface agg 55
Switch_2(config-if)# spanning-tree port disable
Switch_2(config-if)# switchport mode trunk
Switch_2(config-if)# switchport trunk allowed vlan all
Switch_2(config-if)#end
```

step 2 Configure MLAG PEER Communication Address

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 4094
Switch_1(config-vlan)# exit
Switch_1(config)# interface vlan 4094
Switch_1(config-if)# ip address 10.10.0.1/30
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 4094
Switch_2(config-vlan)# exit
Switch_2(config)# interface vlan 4094
Switch_2(config-if)# ip address 10.10.0.2/30
Switch_2(config-if)# end
```

step 3 Configure MLAG PEER

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# mlag configuration
Switch_1(config-mlag)# timers mlag 1 5
Switch_1(config-mlag)# peer-link agg55
Switch_1(config-mlag)# peer-address 10.10.0.2
Switch_1(config-mlag)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# mlag configuration
Switch_2(config-mlag)# timers mlag 1 5
Switch_2(config-mlag)# peer-link agg 55
Switch_2(config-mlag)# peer-address 10.10.0.1
Switch_2(config-mlag)# end
```

step 4 Configure MLAG Port

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# no ip igmp snooping
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 10,20
```

```
Switch_1(config-vlan)# exit
Switch_1(config)# interface eth-0-1
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# channel-group 1 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface eth-0-2
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 20
Switch_1(config-if)# channel-group 2 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface agg 1
Switch_1(config-if)# mlag 1
Switch_1(config-if)# exit
Switch_1(config)# interface agg 2
Switch_1(config-if)# mlag 2
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# no ip igmp snooping
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10,20
Switch_2(config-vlan)# exit
Switch_2(config)# interface eth-0-1
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# channel-group 1 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface eth-0-2
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 20
Switch_2(config-if)# channel-group 2 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface agg 1
Switch_2(config-if)# mlag 1
Switch_2(config-if)# exit
Switch_2(config)# interface agg 2
Switch_2(config-if)# mlag 2
Switch_2(config-if)# end
```

step 5 Configure VARP as a Gateway to Host

configure SWITCH 1

IP address of the interface vlan should be different from ip virtual-router address.

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# interface vlan 10
Switch_1(config-if)# ip address 192.168.10.253/24
Switch_1(config-if)# ip virtual-router address 192.168.1.1/24
```

```
Switch_1(config-if)# ip virtual-router address 192.168.2.1/24
Switch_1(config-if)# exit
Switch_1(config)# interface vlan 20
Switch_1(config-if)# ip address 192.168.20.253/24
Switch_1(config-if)# ip virtual-router address 192.168.3.1/24
Switch_1(config-if)# ip virtual-router address 192.168.4.1/24
Switch_1(config-if)# exit
Switch_1(config)# ip virtual-router mac 0.0.1
Switch_1(config)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface vlan 10
Switch_2(config-if)# ip address 192.168.10.254/24
Switch_2(config-if)# ip virtual-router address 192.168.1.1/24
Switch_2(config-if)# ip virtual-router address 192.168.2.1/24
Switch_2(config-if)# exit
Switch_2(config)# interface vlan 20
Switch_2(config-if)# ip address 192.168.20.254/24
Switch_2(config-if)# ip virtual-router address 192.168.3.1/24
Switch_2(config-if)# ip virtual-router address 192.168.4.1/24
Switch_2(config-if)# exit
Switch_2(config)# ip virtual-router mac 0.0.1
Switch_2(config)# end
```

step 6 Verify Configuration Results

Check the status of MLAG neighbor, MLAG should be Established after configuring MLAG.

```
Switch_1# show mlag peer
MLAG neighbor is 10.10.0.2, MLAG version 1
MLAG state = Established, up for 00:00:01
Last read 00:00:01, hold time is 240, keepalive interval is 60 seconds
Received 4 messages,Sent 4 messages
Open    : received 1, sent 1
KAlive   : received 1, sent 1
Fdb sync  : received 0, sent 0
Failover  : received 0, sent 0
Conf    : received 0, sent 0
Syspri   : received 1, sent 1
Peer fdb : received 1, sent 1

Connections established 1; dropped 0
Local host: 10.10.0.1, Local port: 50040
Foreign host: 10.10.0.2, Foreign port: 61000
remote_sysid: 1a53.71e9.c000
```

Check the status of MLAG device, the two devices are in Master/Slave status.

```
Switch_1# show mlag
MLAG configuration:
-----------------
role       : Master
local_sysid  : 8e79.b120.2e00
remote_sysid : 1a53.71e9.c000
mlag_sysid   : 8e79.b120.2e00
```

```
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300

Switch_2# show mlag
MLAG configuration:
-----------------
role       : Slave
local_sysid  : 1a53.71e9.c000
remote_sysid : 8e79.b120.2e00
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300
```

Check the status of MLAG group, all interfaces should be UP.

```
Switch_1# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1     up          up
2     agg2     up          up
```

Check the status of VARP

```
Switch_1# show ip arp
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   10.10.0.1            -  8e79.b120.2e00  vlan4094
 Internet   10.10.0.2            0  1a53.71e9.c000  vlan4094
 Internet   192.168.1.1          -  0000.0000.0001  vlan10
 Internet   192.168.2.1          -  0000.0000.0001  vlan10
 Internet   192.168.10.253       -  8e79.b120.2e00  vlan10
 Internet   192.168.3.1          -  0000.0000.0001  vlan20
 Internet   192.168.4.1          -  0000.0000.0001  vlan20
 Internet   192.168.20.253       -  8e79.b120.2e00  vlan20
```

MLAG on orphan port

Topology

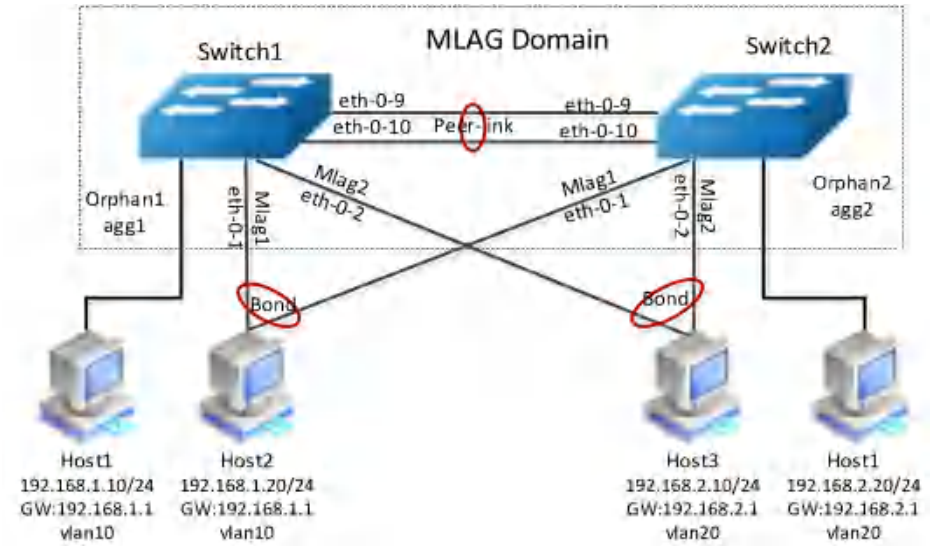


Figure 3-45 MLAG orphan port

Requirement

Host 1 and Host 4 use single network card to connect to MLAG device, Host 2 and Host 3 use dual network cards with active-active method to connect to MLAG device, networking across network section; its network gateway deployment is on MLAG device, requires gateway active-active and it does not use VRRP.

Configuration Steps

Note: The channel-group 55 used to interconnect the two switches in the example configuration can be modified according to different standards of boards, use any channel-group serial number will not affect the configuration of this example.

step 1 Configure MLAG PEER Interconnect Ports

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1 (config)# interface range eth-0-9 – 10
Switch_1 (config-if-range)# no shutdown
Switch_1 (config-if-range)# lacp timeout short
Switch_1 (config-if-range)# channel-group 55 mode active
Switch_1 (config-if-range)# exit
Switch_1 (config)# interface agg 55
Switch_1 (config-if)# spanning-tree port disable
Switch_1 (config-if)# switchport mode trunk
Switch_1 (config-if)# switchport trunk allowed vlan all
Switch_1 (config-if)#end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface range eth-0-9 – 10
Switch_2 (config-if-range)# no shutdown
Switch_2 (config-if-range)# lacp timeout short
Switch_2(config-if-range)# channel-group 55 mode active
Switch_2(config-if-range)# exit
Switch_2(config)# interface agg 55
Switch_2(config-if)# spanning-tree port disable
```

```
Switch_2(config-if)# switchport mode trunk
Switch_2(config-if)# switchport trunk allowed vlan all
Switch_2(config-if)#end
```

step 2 Configure MLAG PEER Communication Address

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 4094
Switch_1(config-vlan)# exit
Switch_1(config)# interface vlan 4094
Switch_1(config-if)# ip address 10.10.0.1/30
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 4094
Switch_2(config-vlan)# exit
Switch_2(config)# interface vlan 4094
Switch_2(config-if)# ip address 10.10.0.2/30
Switch_2(config-if)# end
```

step 3 Configure MLAG PEER

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# mlag configuration
Switch_1(config-mlag)# timers mlag 1 5
Switch_1(config-mlag)# peer-link agg55
Switch_1(config-mlag)# peer-address 10.10.0.2
Switch_1(config-mlag)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# mlag configuration
Switch_2(config-mlag)# timers mlag 1 5
Switch_2(config-mlag)# peer-link agg 55
Switch_2(config-mlag)# peer-address 10.10.0.1
Switch_2(config-mlag)# end
```

step 4 Configure MLAG Interface and orphan Port

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# no ip igmp snooping
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 10,20
Switch_1(config-vlan)# exit
Switch_1(config)# interface eth-0-20
```

```
Switch_1(config-if)# no shutdown
Switch_1(config-if)# switchport access vlan 10
Switch_2(config-if)# exit
Switch_1(config)# interface eth-0-21
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# channel-group 1 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface eth-0-22
Switch_1(config-if)# no shutdown
Switch_1(config-if)# lacp timeout short
Switch_1(config-if)# switchport access vlan 20
Switch_1(config-if)# channel-group 2 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface agg 1
Switch_1(config-if)# mlag 1
Switch_1(config-if)# exit
Switch_1(config)# interface agg 2
Switch_1(config-if)# mlag 2
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# no ip igmp snooping
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10,20
Switch_2(config-vlan)# exit
Switch_2(config)# interface eth-0-21
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# channel-group 1 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface eth-0-22
Switch_2(config-if)# no shutdown
Switch_2(config-if)# lacp timeout short
Switch_2(config-if)# switchport access vlan 20
Switch_2(config-if)# channel-group 2 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface agg 1
Switch_2(config-if)# mlag 1
Switch_2(config-if)# exit
Switch_2(config)# interface agg 2
Switch_2(config-if)# mlag 2
Switch_2(config)# interface eth-0-23
Switch_2(config-if)# no shutdown
Switch_2(config-if)# switchport access vlan 20
Switch_2(config-if)# exit
Switch_2(config-if)# end
```

step 5 Configure VARP as a Gateway to Host

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# interface vlan 10
Switch_1(config-if)# ip address 192.168.1.253/24
Switch_1(config-if)# ip virtual-router address 192.168.1.1
Switch_1(config-if)# exit
Switch_1(config)# interface vlan 20
Switch_1(config-if)# ip address 192.168.2.253/24
Switch_1(config-if)# ip virtual-router address 192.168.2.1
Switch_1(config-if)# exit
Switch_1(config)# ip virtual-router mac 0.0.1
Switch_1(config)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface vlan 10
Switch_2(config-if)# ip address 192.168.1.254/24
Switch_2(config-if)# ip virtual-router address 192.168.1.1
Switch_2(config-if)# exit
Switch_2(config)# interface vlan 20
Switch_2(config-if)# ip address 192.168.2.254/24
Switch_2(config-if)# ip virtual-router address 192.168.2.1
Switch_2(config-if)# exit
Switch_2(config)# ip virtual-router mac 0.0.1
Switch_2(config)# end
```

Multiple MLAG Domain Cascading

Topology

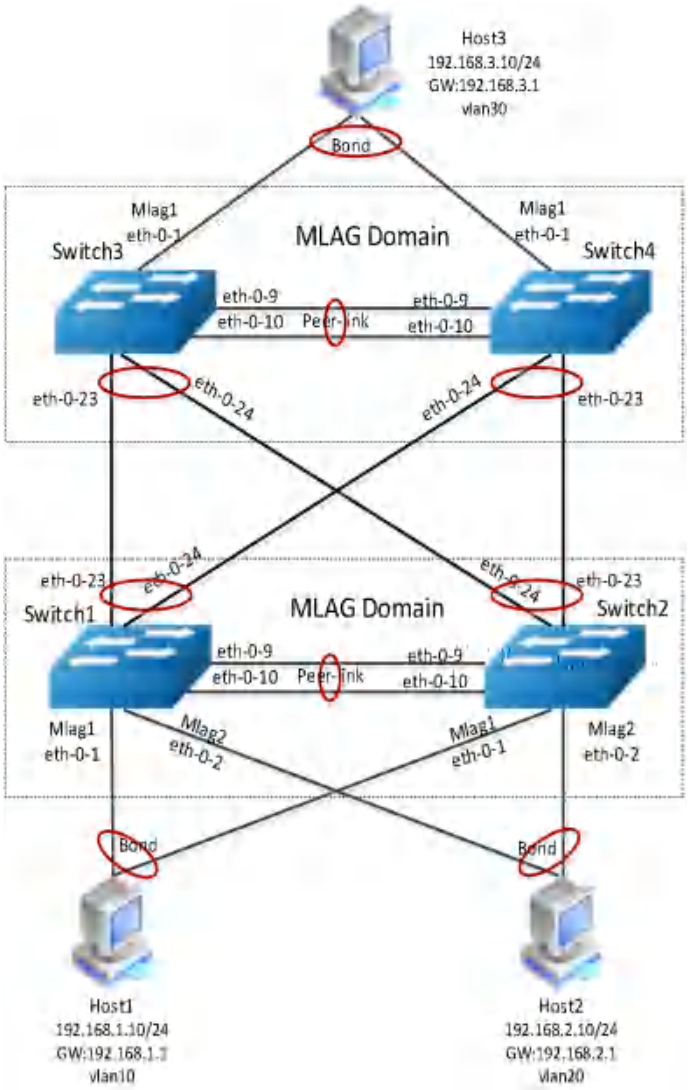


Figure 3-46 Network diagram of MLAG domain cascading

Requirement

While the size of networking increasing, sometime it requires multiple MLAG cascading, asking for connections between Host 1/2/3 and deploying gateway on SWITCH 3/4, and the messages are sent by HOST 1/2 should pass through Layer 2 to forward to SWITCH 3/4.

Configuration Steps

Note: The channel-group 55 used to interconnect the two switches in the example configuration can be modified according to different standards of boards, use any channel-group serial number will not affect the configuration of this example.

step 1 Configure MLAG PEER Interconnect Ports

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1 (config)# interface range eth-0-9 – 10
Switch_1 (config-if-range)# no shutdown
Switch_1 (config-if-range)# lacp timeout short
Switch_1 (config-if-range)# channel-group 55 mode active
Switch_1 (config-if-range)# exit
```

```
Switch_1 (config)# interface agg 55
Switch_1 (config-if)# spanning-tree port disable
Switch_1 (config-if)# switchport mode trunk
Switch_1 (config-if)# switchport trunk allowed vlan all
Switch_1 (config-if)#end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# interface range eth-0-9 – 10
Switch_2 (config-if-range)# no shutdown
Switch_2 (config-if-range)# lacp timeout short
Switch_2(config-if-range)# channel-group 55 mode active
Switch_2(config-if-range)# exit
Switch_2(config)# interface agg 55
Switch_2(config-if)# spanning-tree port disable
Switch_2(config-if)# switchport mode trunk
Switch_2(config-if)# switchport trunk allowed vlan all
Switch_2(config-if)#end
```

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3 (config)# interface range eth-0-9 – 10
Switch_3 (config-if-range)# no shutdown
Switch_3 (config-if-range)# lacp timeout short
Switch_3(config-if-range)# channel-group 55 mode active
Switch_3 (config-if-range)# exit
Switch_3 (config)# interface agg 55
Switch_3 (config-if)# spanning-tree port disable
Switch_3 (config-if-range)# switchport mode trunk
Switch_3 (config-if-range)# switchport trunk allowed vlan all
Switch_3 (config-if)#exit
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# interface range eth-0-9 – 10
Switch_4 (config-if-range)# no shutdown
Switch_4 (config-if-range)# lacp timeout short
Switch_4(config-if-range)# static-channel-group 55
Switch_4(config-if-range)# exit
Switch_4(config)# interface agg 55
Switch_4(config-if)# spanning-tree port disable
Switch_4(config-if-range)# switchport mode trunk
Switch_4(config-if-range)# switchport trunk allowed vlan all
Switch_4(config-if)#exit
```

step 2 Configure MLAG PEER Communication Address

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 4094
```

```
Switch_1(config-vlan)# exit
Switch_1(config)# interface vlan 4094
Switch_1(config-if)# ip address 10.10.0.1/30
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 4094
Switch_2(config-vlan)# exit
Switch_2(config)# interface vlan 4094
Switch_2(config-if)# ip address 10.10.0.2/30
Switch_2(config-if)# end
```

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3(config)# vlan database
Switch_3(config-vlan)# vlan 4094
Switch_3(config-vlan)# exit
Switch_3(config)# interface vlan 4094
Switch_3(config-if)# ip address 10.10.0.5/30
Switch_3(config-if)# end
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# vlan database
Switch_4(config-vlan)# vlan 4094
Switch_4(config-vlan)# exit
Switch_4(config)# interface vlan 4094
Switch_4(config-if)# ip address 10.10.0.6/30
Switch_4(config-if)# end
```

step 3 Configure MLAG PEER

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# mlag configuration
Switch_1(config-mlag)# timers mlag 1 5
Switch_1(config-mlag)# peer-link agg55
Switch_1(config-mlag)# peer-address 10.10.0.2
Switch_1(config-mlag)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# mlag configuration
Switch_2(config-mlag)# timers mlag 1 5
Switch_2(config-mlag)# peer-link agg 55
Switch_2(config-mlag)# peer-address 10.10.0.1
Switch_2(config-mlag)# end
```

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3(config)# mlag configuration
Switch_3(config-mlag)# timers mlag 1 5
Switch_3(config-mlag)# peer-link agg55
Switch_3(config-mlag)# peer-address 10.10.0.6
Switch_3(config-mlag)# end
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# mlag configuration
Switch_4(config-mlag)# timers mlag 1 5
Switch_4(config-mlag)# peer-link agg 55
Switch_4(config-mlag)# peer-address 10.10.0.5
Switch_4(config-mlag)# end
```

step 4 Configure the Interconnection Port Between 1/2 and 3/4

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config)# no ip igmp snooping
Switch_1(config-vlan)# vlan 10,20
Switch_1(config-vlan)# exit
Switch_1(config)# interface range eth-0-23 – 24
Switch_1(config-if-range)# no shutdown
Switch_1(config-if-range)# lacp timeout short
Switch_1(config-if-range)# switchport mode trunk
Switch_1(config-if-range)# switchport trunk allowed vlan add 10,20
Switch_1(config-if-range)# channel-group 54 mode active
Switch_1(config-if-range)# exit
Switch_1(config)# interface agg 54
Switch_1(config-if)# mlag 54
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# no ip igmp snooping
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10,20
Switch_2(config-vlan)# exit
Switch_2(config)# interface range eth-0-23 – 24
Switch_2(config-if-range)# no shutdown
Switch_2(config-if-range)# lacp timeout short
Switch_2(config-if-range)# switchport mode trunk
Switch_2(config-if-range)# switchport trunk allowed vlan add 10,20
Switch_2(config-if-range)# channel-group 54 mode active
Switch_2(config-if-range)# exit
Switch_2(config)# interface agg 54
Switch_2(config-if)# mlag 54
Switch_2(config-if)# end
```

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3(config)# no ip igmp snooping
Switch_3(config)# vlan database
Switch_3(config-vlan)# vlan 10,20
Switch_3(config-vlan)# exit
Switch_3(config)# interface range eth-0-23 – 24
Switch_3(config-if-range)# no shutdown
Switch_3(config-if-range)# lacp timeout short
Switch_3(config-if-range)# switchport mode trunk
Switch_3(config-if-range)# switchport trunk allowed vlan add 10,20
Switch_3(config-if-range)# channel-group 54 mode active
Switch_3(config-if-range)# exit
Switch_3(config)# interface agg 54
Switch_3(config-if)# mlag 54
Switch_3(config-if)# end
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# no ip igmp snooping
Switch_4(config)# vlan database
Switch_4(config-vlan)# vlan 10,20
Switch_4(config-vlan)# exit
Switch_4(config)# interface range eth-0-23 – 24
Switch_4(config-if-range)# no shutdown
Switch_4(config-if-range)# lacp timeout short
Switch_4(config-if-range)# switchport mode trunk
Switch_4(config-if-range)# switchport trunk allowed vlan add 10,20
Switch_4(config-if-range)# channel-group 54 mode active
Switch_4(config-if-range)# exit
Switch_4(config)# interface agg 54
Switch_4(config-if)# mlag 54
Switch_4(config-if)# end
```

step 5 Configure MALG Group for HOST Connection

configure SWITCH 1

```
Switch_1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_1(config)# vlan database
Switch_1(config-vlan)# vlan 10,20
Switch_1(config-vlan)# exit
Switch_1(config)# mlag lacp system-id 0000.0000.aabb
Switch_1(config)# interface eth-0-1
Switch_1(config-if)# no shutdown
Switch_1(config-if)# switchport access vlan 10
Switch_1(config-if)# channel-group 1 mode active
Switch_1(config-if)# exit
Switch_1(config)# interface eth-0-2
Switch_1(config-if)# no shutdown
Switch_1(config-if)# switchport access vlan 20
Switch_1(config-if)# channel-group 2 mode active
Switch_1(config-if)# exit
```

```
Switch_1(config)# interface agg 1
Switch_1(config-if)# mlag 1
Switch_1(config-if)# exit
Switch_1(config)# interface agg 2
Switch_1(config-if)# mlag 2
Switch_1(config-if)# end
```

configure SWITCH 2

```
Switch_2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_2(config)# vlan database
Switch_2(config-vlan)# vlan 10,20
Switch_2(config-vlan)# exit
Switch_2(config)# mlag lacp system-id 0000.0000.aabb
Switch_2(config)# interface eth-0-1
Switch_2(config-if)# no shutdown
Switch_2(config-if)# switchport access vlan 10
Switch_2(config-if)# channel-group 1 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface eth-0-2
Switch_2(config-if)# no shutdown
Switch_2(config-if)# switchport access vlan 20
Switch_2(config-if)# channel-group 2 mode active
Switch_2(config-if)# exit
Switch_2(config)# interface agg 1
Switch_2(config-if)# mlag 1
Switch_2(config-if)# exit
Switch_2(config)# interface agg 2
Switch_2(config-if)# mlag 2
Switch_2(config-if)# end
```

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3(config)# vlan database
Switch_3(config-vlan)# vlan 30
Switch_3(config-vlan)# exit
Switch_3(config)# interface eth-0-1
Switch_3(config-if)# no shutdown
Switch_3(config-if)# lacp timeout short
Switch_3(config-if)# switchport access vlan 30
Switch_3(config-if)# channel-group 1 mode active
Switch_3(config-if)# exit
Switch_3(config)# interface agg 1
Switch_3(config-if)# mlag 1
Switch_3(config-if)# end
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# vlan database
Switch_4(config-vlan)# vlan 30
Switch_4(config-vlan)# exit
Switch_4(config)# interface eth-0-1
Switch_4(config-if)# no shutdown
```

```
Switch_4(config-if)# lacp timeout short
Switch_4(config-if)# switchport access vlan 30
Switch_4(config-if)# channel-group 1 mode active
Switch_4(config-if)# exit
Switch_4(config)# interface agg 1
Switch_4(config-if)# mlag 1
Switch_4(config-if)# end
```

step 6 Configure VARP as a Gateway to Host

The gateway is deployed on SWITCH 3/4, SWITCH 1/2 does not need to be configured.

configure SWITCH 3

```
Switch_3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_3(config)# interface vlan 10
Switch_3(config-if)# ip address 192.168.1.253/24
Switch_3(config-if)# ip virtual-router address 192.168.1.1
Switch_3(config-if)# exit
Switch_3(config)# interface vlan 20
Switch_3(config-if)# ip address 192.168.2.253/24
Switch_3(config-if)# ip virtual-router address 192.168.2.1
Switch_3(config-if)# exit
Switch_3(config)# interface vlan 30
Switch_3(config-if)# ip address 192.168.3.253/24
Switch_3(config-if)# ip virtual-router address 192.168.3.1
Switch_3(config-if)# exit
Switch_3(config)# ip virtual-router mac 0.0.1
Switch_3(config)# end
```

configure SWITCH 4

```
Switch_4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch_4(config)# interface vlan 10
Switch_4(config-if)# ip address 192.168.1.254/24
Switch_4(config-if)# ip virtual-router address 192.168.1.1
Switch_4(config-if)# exit
Switch_4(config)# interface vlan 20
Switch_4(config-if)# ip address 192.168.2.254/24
Switch_4(config-if)# ip virtual-router address 192.168.2.1
Switch_4(config-if)# exit
Switch_4(config)# interface vlan 30
Switch_4(config-if)# ip address 192.168.3.254/24
Switch_4(config-if)# ip virtual-router address 192.168.3.1
Switch_4(config-if)# exit
Switch_4(config)# ip virtual-router mac 0.0.1
Switch_4(config)# end
```

step 7 Verify Configuration Results

Check the status of MLAG neighbor, MLAG should be Established after configuring MLAG.

```
Switch_1# show mlag peer
MLAG neighbor is 10.10.0.2, MLAG version 1
MLAG state = Established, up for 00:00:01
Last read 00:00:01, hold time is 240, keepalive interval is 60 seconds
Received 4 messages,Sent 4 messages
```

```
Open    : received 1, sent 1
KAlive   : received 1, sent 1
Fdb sync  : received 0, sent 0
Failover  : received 0, sent 0
Conf    : received 0, sent 0
Syspri   : received 1, sent 1
Peer fdb : received 1, sent 1

Connections established 1; dropped 0
Local host: 10.10.0.1, Local port: 50040
Foreign host: 10.10.0.2, Foreign port: 61000
remote_sysid: 1a53.71e9.c000
```

Check the status of MLAG device, the two devices are in Master/Slave status

```
Switch_1# show mlag
MLAG configuration:
------------------
role       : Master
local_sysid  : 8e79.b120.2e00
remote_sysid : 1a53.71e9.c000
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300

Switch_2# show mlag
MLAG configuration:
------------------
role       : Slave
local_sysid  : 1a53.71e9.c000
remote_sysid : 8e79.b120.2e00
mlag_sysid   : 8e79.b120.2e00
local_syspri : 32768
remote_syspri: 32768
mlag_syspri  : 32768
peer-link    : agg55
peer conf    : Yes
reload-delay : 300
```

Check the status of MLAG group, all interfaces should be UP.

```
Switch_1# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1     up          up
2     agg2     up          up
54     agg54    up          up
```

Check the status of VARP.

```
Switch_3# show ip arp
 Protocol   Address        Age (min)  Hardware Addr   Interface
 Internet   10.10.0.5           -   50bd.ac96.f800  vlan4094
 Internet   10.10.0.6           0   fef0.6b89.5800  vlan4094
 Internet   192.168.1.1         -   0000.0000.0001  vlan10
```

```
Internet  192.168.1.253       -  50bd.ac96.f800  vlan10
Internet  192.168.2.1         -  0000.0000.0001  vlan20
Internet  192.168.2.253       -  50bd.ac96.f800  vlan20
Internet  192.168.3.1         -  0000.0000.0001  vlan30
Internet  192.168.3.253       -  50bd.ac96.f800  vlan30
```

After configuring HOST, it should use ping to pass through gateway. It can also use ping between HOSTs.

3.17 Configuring Hash Load-balance

3.17.1 Configuring Linkagg Hash

Overview

Linkagg can aggregate several physical interface to be a logical channel to enhance proformance and redundancy.When use linkagg transmit packets,it could be cause the same data stream transmitting on different physical interfaces.Because of that,the opposite equipment can receive packet disordering. In order to avoid this phennomenon,linkagg can accrod packets property to get a hash value,then it chooses appropriate physical interface to transmit packets.Besides this,it also can improve linkagg load balancing result.

Configuring Linkagg Hash Globally

The follow steps show how to set unicast and non-unicast linkagg hash on packets output interface globally and the configurations has the lowest priority.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value global

```
Switch(config)# hash-value global
Switch(config-hash-value-global)# port-channel select user
Switch(config-hash-value-global)# end
```

step 4 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------
 ipv6 address compress          xor
 hash seed                   user set (0)
 hash arithmetic               xor
 hash symmetry                 disable
 ip                         enable
 ipv6                       enable
 mpls                       enable
------------------------------------------------------------------------
 hash field select
 Packet           HashField
------------------------------------------------------------------------
```

```
l2:              macsa

ip:              ipsa

ipv6:            ipsa            ipda
                 l4-sourceport   l4-destport
                 ip-protocol

gre:             ipsa            ipda
                 gre-key

vxlan:           vni             outer-l4-sourceport
                 outer-ipda      outer-ipsa

nvgre:           vsid            outer-ipda
                 outer-ipsa

mpls:            top-label       2nd-label

vpws:            top-label       2nd-label

vpls(inner-l2):  inner-macda     inner-macsa

vpls(inner-l3):  inner-ipda      inner-ipsa

l3vpn:           inner-ipsa      inner-ipda
                 inner-ip-protocol  inner-l4-sourceport
                 inner-l4-destport
```

Use the following command to display the information of hash value global:

```
Switch# show hash-value global
LBT:load balance type      LBM :load balance mode
PT :packet type            HF  :hash field
HA :hash arithmetic
hash-value global
 LBT        LBM        PT         HF         HA
--------------------------------------------------------------------------------
 port-channel  -          all        user       xor
 ecmp       -          all        ecmp       xor
 ecmp       flow id    all        ecmp       xor
 entropy    -          all        ecmp       xor
--------------------------------------------------------------------------------
 Efd hash field select:
  macsa            macda
  ipsa             ipda
  sourceport       destport
  ip-protocol
```

Configuring Linkagg Hash Input

The follow steps show how to set unicast linkagg hash on input interface and the configuration priority is higher than output. When the hash value is applied to in the input of linkagg port, the hash value will apply to the member port of linkagg port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value aaa
Switch(config-hash-value)# port-channel unicast select user
Switch(config-hash-value)# exit
```

step 4 Set hash value to interface

```
Switch(config)# interface range eth-0-1 - 2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# static-channel-group 1
Switch(config-if-range)# exit
Switch(config)# interface agg 1
Switch(config-if)# load-balance hash-value aaa input
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# load-balance hash-value aaa input
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                    user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                         enable
ipv6                       enable
mpls                       enable
------------------------------------------------------------------------
hash field select
Packet           HashField
------------------------------------------------------------------------
l2:              macsa

ip:              ipsa

ipv6:            ipsa            ipda
                 l4-sourceport       l4-destport
                 ip-protocol

gre:             ipsa            ipda
                 gre-key

vxlan:           vni             outer-l4-sourceport
                 outer-ipda          outer-ipsa
```

```
nvgre:              vsid              outer-ipda
                outer-ipsa

mpls:               top-label         2nd-label

vpws:               top-label         2nd-label

vpls(inner-l2):     inner-macda       inner-macsa

vpls(inner-l3):     inner-ipda        inner-ipsa

l3vpn:              inner-ipsa        inner-ipda
                inner-ip-protocol   inner-l4-sourceport
                inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value aaa
LBT:load balance type       LBM:load balance mode
PT :packet type             HF :hash field
HA :hash arithmetic
hash-value name: aaa
 LBT          LBM          PT           HF           HA
--------------------------------------------------------------------------------
 port-channel   unicast      all          user         xor
 port-channel   non-unicast   all          NOCFG        NOCFG
 ecmp         -            all          NOCFG        NOCFG
 ecmp         flow id      all          NOCFG        NOCFG
```

Use the following command to display the application of hash value on port:

```
Switch# show hash-value interface-applied
eth-0-3
 hash-value aaa input
agg1
 hash-value aaa input
```

Configuring Linkagg Hash output

The follow steps show how to set unicast linkagg hash on output interface and the configuration priority is lower than input. It only can be applied on linkagg port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value aaa
Switch(config-hash-value)# port-channel unicast select user
Switch(config-hash-value)# exit
```

step 4 Set hash value to interface

```
Switch(config)# interface range eth-0-1 - 2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# static-channel-group 1
Switch(config-if-range)# exit
Switch(config)# interface agg 1
Switch(config-if)# load-balance hash-value aaa output
Switch(config-if)# exit
```

step 5 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                    user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                          enable
ipv6                         enable
mpls                         enable
------------------------------------------------------------------------
hash field select
Packet            HashField
------------------------------------------------------------------------
l2:               macsa

ip:               ipsa

ipv6:             ipsa              ipda
                  l4-sourceport     l4-destport
                  ip-protocol

gre:              ipsa              ipda
                  gre-key

vxlan:            vni               outer-l4-sourceport
                  outer-ipda        outer-ipsa

nvgre:            vsid              outer-ipda
                  outer-ipsa

mpls:             top-label         2nd-label

vpws:             top-label         2nd-label

vpls(inner-l2):   inner-macda       inner-macsa

vpls(inner-l3):   inner-ipda        inner-ipsa

l3vpn:            inner-ipsa        inner-ipda
                  inner-ip-protocol inner-l4-sourceport
                  inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value aaa
LBT:load balance type       LBM:load balance mode
PT :packet type             HF :hash field
HA :hash arithmetic
hash-value name: aaa
 LBT          LBM          PT           HF           HA
--------------------------------------------------------------------------
 port-channel  unicast      all          user         xor
 port-channel  non-unicast  all          NOCFG        NOCFG
 ecmp          -            all          NOCFG        NOCFG
 ecmp          flow id      all          NOCFG        NOCFG
```

Use the following command to display the application of hash value on port:

```
Switch# show hash-value interface-applied
agg1
 hash-value aaa output
```

Configuring Linkagg Hash ACL

The follow steps show how to make linkagg hash configurations to be a ACL action and the configurations have the highest priority.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value aaa
Switch(config-hash-value)# port-channel unicast select user
Switch(config-hash-value)# exit
```

step 4 Add acl action to interface and set hash value to interface

```
Switch(config)# mac access-list mac
Switch(config-mac-acl)# permit src-mac host 0.0.1 dest-mac any
Switch(config-mac-acl)# exit
Switch(config)# class-map cmap1
Switch(config-cmap)# match access-group mac
Switch(config-cmap)# exit
Switch(config)# policy-map pmap1
Switch(config-pmap)# class cmap1
Switch(config-pmap-c)# load-balance hash-value aaa
Switch(config-pmap-c)# port-channel load-balance round-robin disable
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# no shutdown
Switch(config-if)# service-policy input pmap1
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                   user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                         enable
ipv6                        enable
mpls                        enable
------------------------------------------------------------------------------
hash field select
Packet            HashField
------------------------------------------------------------------------------
l2:               macsa

ip:               ipsa

ipv6:             ipsa              ipda
                  l4-sourceport     l4-destport
                  ip-protocol

gre:              ipsa              ipda
                  gre-key

vxlan:            vni               outer-l4-sourceport
                  outer-ipda        outer-ipsa

nvgre:            vsid              outer-ipda
                  outer-ipsa

mpls:             top-label         2nd-label

vpws:             top-label         2nd-label

vpls(inner-l2):   inner-macda       inner-macsa

vpls(inner-l3):   inner-ipda        inner-ipsa

l3vpn:            inner-ipsa        inner-ipda
                  inner-ip-protocol inner-l4-sourceport
                  inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value aaa
LBT:load balance type      LBM:load balance mode
PT :packet type            HF :hash field
HA :hash arithmetic
hash-value name: aaa
 LBT        LBM        PT         HF         HA
------------------------------------------------------------------------------
```

```
port-channel  unicast     all        user       xor
port-channel  non-unicast  all       NOCFG      NOCFG
ecmp       -          all       NOCFG      NOCFG
ecmp       flow id    all       NOCFG      NOCFG
```

Use the following command to display the information of ACL:

```
Switch# show running-config
mac access-list mac
 10 permit src-mac host 0000.0000.0001 dest-mac any
!
hash-field user
 l2 macsa
 ip ipsa
!
hash-value aaa
 port-channel unicast select user
!
class-map match-any cmap1
 match access-group mac
!
policy-map pmap1
 class cmap1
  port-channel load-balance round-robin disable
  load-balance hash-value aaa
!
interface eth-0-3
 service-policy input pmap1
!
interface null0
!
```

Configuring Non-unicast Linkagg Hash

The follow steps show how to set non-unicast linkagg hash on input interface and the configuration does not support on output. When the hash value is applied to in the input of linkagg port, the hash value will apply to the member port of linkagg port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value aaa
Switch(config-hash-value)# port-channel non-unicast select user
Switch(config-hash-value)# exit
```

step 4 Set hash value to interface

```
Switch(config)# interface range eth-0-1 - 2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# static-channel-group 1
Switch(config-if-range)# exit
Switch(config)# interface agg 1
Switch(config-if)# load-balance hash-value aaa input
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# load-balance hash-value aaa input
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                   user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                         enable
ipv6                        enable
mpls                        enable
------------------------------------------------------------------------
hash field select
Packet            HashField
------------------------------------------------------------------------
l2:               macsa

ip:               ipsa

ipv6:             ipsa              ipda
                  l4-sourceport     l4-destport
                  ip-protocol

gre:              ipsa              ipda
                  gre-key

vxlan:            vni               outer-l4-sourceport
                  outer-ipda        outer-ipsa

nvgre:            vsid              outer-ipda
                  outer-ipsa

mpls:             top-label         2nd-label

vpws:             top-label         2nd-label

vpls(inner-l2):   inner-macda       inner-macsa

vpls(inner-l3):   inner-ipda        inner-ipsa
```

```
l3vpn:          inner-ipsa          inner-ipda
                inner-ip-protocol   inner-l4-sourceport
                inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value aaa
LBT:load balance type       LBM:load balance mode
PT :packet type             HF :hash field
HA :hash arithmetic
hash-value name: aaa
 LBT          LBM          PT           HF           HA
--------------------------------------------------------------------------------
 port-channel  unicast      all          NOCFG        NOCFG
 port-channel  non-unicast  all          user         xor
 ecmp          -            all          NOCFG        NOCFG
 ecmp          flow id      all          NOCFG        NOCFG
```

Use the following command to display the application of hash value on port:

Use the following command to display the application of hash value on port:

```
Switch# show hash-value interface-applied
eth-0-3
 hash-value aaa input
agg1
 hash-value aaa input
```

3.17.2 Configuring ECMP Hash

Overview

Equal-cost multi-path routing is a routing strategy where next-hop packet forwarding to a single destination can occur over multiple “best paths” which tie for top place in routing metric calculations.Multi-path routing cam be used in conjunction with most routing protocols,because it is a per-hop decision limited to a single router.It can substantially increase bandwidth by load-balancing traffic over multiple paths.Ecmp hash is used to do load balance.

Configuring ECMP Hash Globally

The follow steps show how to set ecmp hash globally and the configurations has the lowest priority.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value global

```
Switch(config)# hash-value global
Switch(config-hash-value-global)# ecmp select user
Switch(config-hash-value-global)# end
```

step 4 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                      Control type
------------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                    user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                          enable
ipv6                         enable
mpls                         enable
------------------------------------------------------------------------------
hash field select
Packet            HashField
------------------------------------------------------------------------------
l2:               macsa

ip:               ipsa

ipv6:             ipsa              ipda
                  l4-sourceport     l4-destport
                  ip-protocol

gre:              ipsa              ipda
                  gre-key

vxlan:            vni               outer-l4-sourceport
                  outer-ipda        outer-ipsa

nvgre:            vsid              outer-ipda
                  outer-ipsa

mpls:             top-label         2nd-label

vpws:             top-label         2nd-label

vpls(inner-l2):   inner-macda       inner-macsa

vpls(inner-l3):   inner-ipda        inner-ipsa

l3vpn:            inner-ipsa        inner-ipda
                  inner-ip-protocol inner-l4-sourceport
                  inner-l4-destport
```

Use the following command to display the information of hash value global:

```
Switch# show hash-value global
LBT:load balance type      LBM :load balance mode
PT :packet type            HF  :hash field
HA :hash arithmetic
hash-value global
 LBT        LBM        PT         HF         HA
------------------------------------------------------------------------------
 port-channel  -          all        port-channel  xor
 ecmp       -          all        user       xor
 ecmp       flow id    all        user       xor
```

```
entropy     -          all        ecmp       xor
------------------------------------------------------------------------
Efd hash field select:
 macsa              macda
 ipsa             ipda
 sourceport         destport
 ip-protocol
```

Configuring ECMP Hash Input

The follow steps show how to set ECMP hash on input interface and the configuration priority is higher than global configuration.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value bbb
Switch(config-hash-value)# ecmp select user
Switch(config-hash-value)# exit
```

step 4 Set hash value to interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# load-balance hash-value bbb input
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of hash field user:

```
  Switch# show hash-field user
hash-field name: user
Option                      Control type
------------------------------------------------------------------------
ipv6 address compress          xor
hash seed                   user set (0)
hash arithmetic               xor
hash symmetry                 disable
ip                         enable
ipv6                        enable
mpls                        enable
------------------------------------------------------------------------
hash field select
Packet           HashField
------------------------------------------------------------------------
l2:              macsa

ip:              ipsa
```

```
ipv6:            ipsa            ipda
                 l4-sourceport   l4-destport
                 ip-protocol

gre:             ipsa            ipda
                 gre-key

vxlan:           vni             outer-l4-sourceport
                 outer-ipda      outer-ipsa

nvgre:           vsid            outer-ipda
                 outer-ipsa

mpls:            top-label       2nd-label

vpws:            top-label       2nd-label

vpls(inner-l2):  inner-macda     inner-macsa

vpls(inner-l3):  inner-ipda      inner-ipsa

l3vpn:           inner-ipsa      inner-ipda
                 inner-ip-protocol  inner-l4-sourceport
                 inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value bbb
LBT:load balance type     LBM:load balance mode
PT :packet type           HF :hash field
HA :hash arithmetic
hash-value name: bbb
 LBT          LBM          PT           HF           HA
--------------------------------------------------------------------------------
 port-channel  unicast      all          NOCFG        NOCFG
 port-channel  non-unicast  all          NOCFG        NOCFG
 ecmp          -            all          user         xor
 ecmp          flow id      all          NOCFG        NOCFG
```

Use the following command to display the application of hash value on port:

```
Switch# show hash-value interface-applied
eth-0-1
 hash-value bbb input
```

Configuring ECMP Hash input

The follow steps show how to make ECMP hash configurations to be a ACL action and the configurations have the highest priority.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash field

```
Switch(config)# hash-field user
Switch(config-hash-field)# l2 macsa
Switch(config-hash-field)# ip ipsa
Switch(config-hash-field)# exit
```

step 3 Set hash value

```
Switch(config)# hash-value bbb
Switch(config-hash-value)# ecmp select user
Switch(config-hash-value)# exit
```

step 4 Add acl action to interface and set hash value to interface

```
Switch(config)# mac access-list mac
Switch(config-mac-acl)# permit src-mac host 0.0.1 dest-mac any
Switch(config-mac-acl)# exit
Switch(config)# class-map cmap1
Switch(config-cmap)# match access-group mac
Switch(config-cmap)# exit
Switch(config)# policy-map pmap1
Switch(config-pmap)# class cmap1
Switch(config-pmap-c)# load-balance hash-value bbb
Switch(config-pmap-c)# ecmp load-balance round-robin disable
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# service-policy input pmap1
Switch(config-if)# end
```

step 5 Validation

Use the following command to display the information of hash field user:

```
Switch# show hash-field user
hash-field name: user
 Option                        Control type
------------------------------------------------------------------------------
ipv6 address compress           xor
hash seed                     user set (0)
hash arithmetic                 xor
hash symmetry                   disable
ip                            enable
ipv6                          enable
mpls                          enable
------------------------------------------------------------------------------
hash field select
Packet            HashField
------------------------------------------------------------------------------
l2:               macsa

ip:               ipsa

ipv6:             ipsa              ipda
                  l4-sourceport     l4-destport
                  ip-protocol

gre:              ipsa              ipda
                  gre-key

vxlan:            vni               outer-l4-sourceport
                  outer-ipda        outer-ipsa
```

```
nvgre:              vsid              outer-ipda
                outer-ipsa

mpls:               top-label         2nd-label

vpws:               top-label         2nd-label

vpls(inner-l2):     inner-macda       inner-macsa

vpls(inner-l3):     inner-ipda        inner-ipsa

l3vpn:              inner-ipsa        inner-ipda
                inner-ip-protocol   inner-l4-sourceport
                inner-l4-destport
```

Use the following command to display the information of hash value:

```
Switch# show hash-value bbb
LBT:load balance type       LBM:load balance mode
PT :packet type             HF :hash field
HA :hash arithmetic
hash-value name: bbb
 LBT          LBM          PT           HF           HA
--------------------------------------------------------------------------------
 port-channel   unicast      all          NOCFG        NOCFG
 port-channel   non-unicast  all          NOCFG        NOCFG
 ecmp         -            all          user         xor
 ecmp         flow id      all          NOCFG        NOCFG
```

Use the following command to display the information of ACL:

```
mac access-list mac
 10 permit src-mac host 0000.0000.0001 dest-mac any
!
hash-field user
 l2 macsa
 ip ipsa
!
hash-value bbb
 ecmp select user
!
class-map match-any cmap1
 match access-group mac
!
policy-map pmap1
 class cmap1
  ecmp load-balance round-robin disable
  load-balance hash-value bbb
!
interface eth-0-1
 service-policy input pmap1
!
interface null0
!
```

3.17.3 Configuring EFD Hash

Overview

Elephant Flow Detect(EFD). According to the academic institutions of the actual network of the study found that more than 80% of the bandwidth is occupied by elephant flow, the bandwidth and transmission cache of these flow is large, but not sensitive to delay, which is sensitive to delay The flow caused a great impact.EFD hash is used to detect elephant flow by recognising packet features.

Configuring EFD Hash Globally

The follow steps show how to select packet features for EFD hash globally.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set hash global

```
Switch(config)# hash-value global
Switch(config-hash-value-global)# efd select ipsa macsa
Switch(config-hash-value-global)# end
```

step 3 Validation

Use the following command to display the information of hash value global:

```
Switch# show hash-value global
LBT:load balance type       LBM :load balance mode
PT :packet type             HF  :hash field
HA :hash arithmetic
hash-value global
 LBT         LBM          PT          HF          HA
------------------------------------------------------------------------
 port-channel   -            all          port-channel   xor
 ecmp         -            all          ecmp         xor
 ecmp         flow id      all          ecmp         xor
 entropy      -            all          ecmp         xor
------------------------------------------------------------------------
 Efd hash field select:
  macsa              ipsa
```

3.18 Configuring PORT-XCONNECT

3.18.1 Overview

Function Introduction

This feature can forward the packet directly according to the destination-interface configured without looking up any table items and forwarding.

Only physical and aggregate port are currently supported.

Principle Description

N/A

3.18.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface mode and no shutdown

```
Switch(config)# interface range eth-0-1 , eth-0-2
Switch(config-if-range)# no shutdown
```

step 3 Set eth-0-1 port-xconnect destination interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# port-xconnect destination-interface eth-0-2
Switch(config-if)# end
```

step 4 Display configuration

```
Switch# show running-config
Building configuration...
version 5.3.9.18
!
no service password-encryption
!
!
!
!
!
!
!
temperature 0 0 0
!
vlan database
!

interface eth-0-1
port-xconnect destination-interface eth-0-2
!
interface eth-0-2
!
interface eth-0-3
Switch#
```

3.18.3 Application cases

N/A

4 IP Service Configuration Guide

4.1 Configuring Arp

4.1.1 Overview

Function Introduction

The Address Resolution Protocol (ARP) is a protocol used to dynamically map between Internet host addresses and Ethernet addresses. ARP caches Internet-Ethernet address mappings. When an interface requests a mapping for an address not in the cache, ARP queues the message, which requires the mapping, and broadcasts a message on the associated network requesting the address mapping. If a response is provided, the new mapping is cached and any pending message is transmitted. ARP will queue at most one packet while waiting for a response to a mapping request; only the most recently transmitted packet is kept. If the target host does not respond after 3 requests, the host is considered to be down, allowing an error to be returned to transmission attempts during this interval. If a target host does not send message for a period (normally one hour), the host is considered to be uncertainty, and several requests (normally 6, 3 unicast and 3 broadcast) will send to the host before delete the ARP entry. ARP entries may be added, deleted or changed manually. Manually added entries may be temporary or permanent.

Principle Description

N/A

4.1.2 Configuration

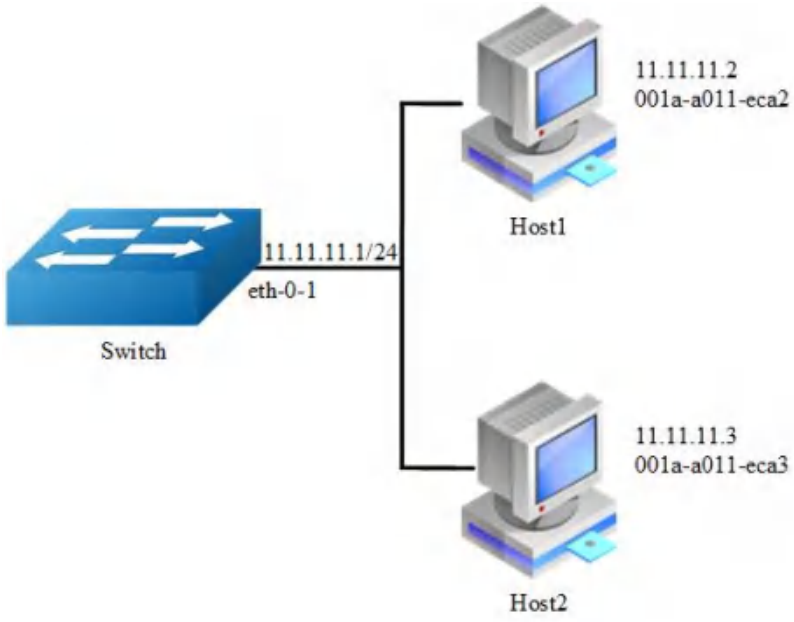


Figure 4-1 arp

In this configuration example, interface eth-0-1 assigned with address 11.11.11.1/24, on subnet 11.11.11.0/24, there are two hosts, and their IP addresses are 11.11.11.2, 11.11.11.3, MAC address are 001a-a011-eca2, 001a-a011-eca3. ARP entry of host 11.11.11.2 is added manually, the entry of host 11.11.11.3 is added dynamically. Time-out period of ARP entries for interface eth-0-1 configure to 20 minutes, ARP request retry delay on interface eth-0-1 configure to 2 seconds.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the layer 3 interface and set the ip address

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 11.11.11.1/24
```

step 3 Configure arp aging timeout value and the arp retry interval value

```
Switch(config-if)# arp timeout 1200
Switch(config-if)# arp retry-interval 2
Switch(config-if)# exit
```

step 4 Add a static arp entry

```
Switch(config)# arp 11.11.11.2 1a.a011.eca2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the information of the arp entry:

```
Switch# show ip arp
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   11.11.11.2           -   001a.a011.eca2    eth-0-1

Switch# show ip arp summary
 1 IP ARP entries, with 0 of them incomplete
(Static:0,  Dyamic:0, Interface:1)
 ARP Pkt Received is: 0
 ARP Pkt Send number is: 0
 ARP Pkt Dicard number is: 0
```

Use the following command to display the information of the arp configurations on the interface:

```
Switch# show interface eth-0-1
Interface eth-0-1
 Interface current state: Administratively DOWN
 Hardware is Ethernet, address is 6c02.530c.2300 (bia 6c02.530c.2300)
 Bandwidth 1000000 kbits
 Index 1 , Metric 1 , Encapsulation ARPA
 Speed - Auto , Duplex - Auto  , Media type is 1000BASE_T
 Link speed type is autonegotiation, Link duplex type is autonegotiation
 Input flow-control is off, output flow-control is off
 The Maximum Frame Size is 1534 bytes
 VRF binding: not bound
 Label switching is disabled
 No virtual circuit configured
 VRRP master of :  VRRP is not configured on this interface
 ARP timeout 00:20:00,  ARP retry interval 2s
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes
   Received 0 unicast, 0 broadcast, 0 multicast
   0 runts, 0 giants, 0 input errors, 0 CRC
   0 frame, 0 overrun, 0 pause input
   0 input packets with dribble condition detected
   0 packets output, 0 bytes
```

```
 Transmitted 0 unicast, 0 broadcast, 0 multicast
0 underruns, 0 output errors, 0 pause output
```

4.1.3 Application cases

N/A

4.2 Configuring Arp proxy

4.2.1 Overview

Function Introduction

Proxy ARP, the most common method for learning about other routes, enables an Ethernet host with no routing information to communicate with hosts on other networks or subnets. The host assumes that all hosts are on the same local Ethernet and that they can use ARP to determine their MAC addresses. If a switch receives an ARP request for a host that is not on the same network as the sender, the switch evaluates whether it has the best route to that host. If it does, it sends an ARP reply packet with its own Ethernet MAC address, and the host that sent the request sends the packet to the switch, which forwards it to the intended host. Proxy ARP treats all networks as if they are local and performs ARP requests for every IP address. Proxy ARP can be separated to 2 parts: Proxy ARP and local Proxy ARP. Local Proxy ARP is always used in the topology where the Device is enabled port isolate but still need to do communicating via routing. Internet Control Message Protocol (ICMP) redirects are disabled on interfaces where the local proxy ARP feature is enabled.

Principle Description

N/A

4.2.2 Configuration

Configuring ARP Proxy

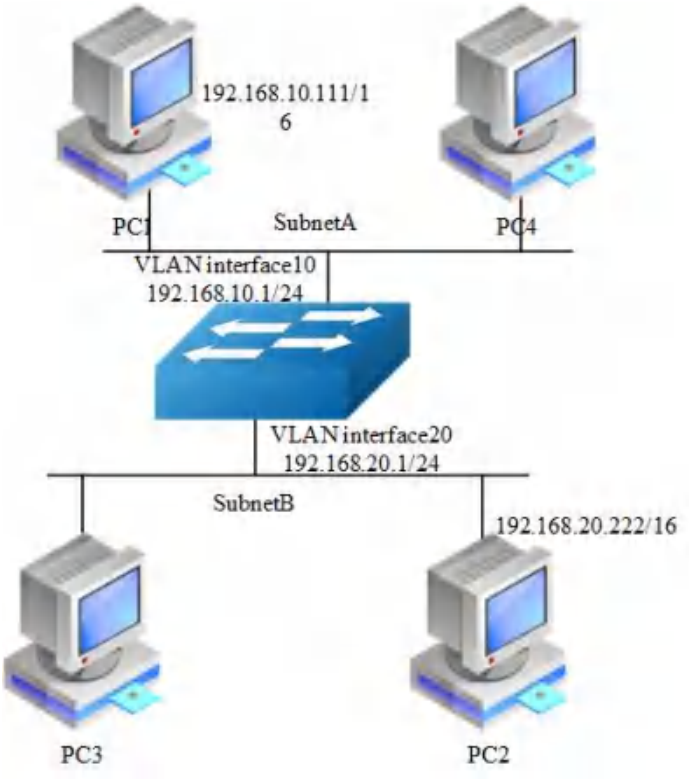


Figure 4-2 arp proxy

As seen in the above topology, PC1 is belonged to VLAN10 and PC2 is belonged to VLAN20. If ARP proxy feature is not enabled, then PC1 and PC2 can not communicate with each other. As following, these steps are shown to enable ARP proxy feature for both VLAN interface 10 and VLAN interface 20.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode, set the switch port mode and bind to the vlan

```
Switch(config)# interface eth-0-22
Switch(config-if)# switchport access vlan 10
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-23
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Create the vlan interface, configure the ip address, and enable arp proxy

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.10.1/24
Switch(config-if)# proxy-arp enable
Switch(config-if)# exit

Switch(config)# interface vlan 20
Switch(config-if)# ip address 192.168.20.1/24
Switch(config-if)# proxy-arp enable
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the information of the arp proxy configuration on the switch:

```
Switch# show ip interface vlan 10
Interface vlan10
 Interface current state: UP
 Internet address(es):
  192.168.10.1/24 broadcast 192.168.10.255
 Joined group address(es):
  224.0.0.1
 The maximum transmit unit is 1500 bytes
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ICMP unreachables are always sent
 ICMP mask replies are always sent
 ARP timeout 01:00:00,  ARP retry interval 1s
 ARP Proxy is enabled, Local ARP Proxy is disabled
```

```
 VRRP master of :  VRRP is not configured on this interface

Switch# show ip interface vlan 20
Interface vlan20
 Interface current state: UP
 Internet address(es):
  192.168.20.1/24 broadcast 192.168.20.255
 Joined group address(es):
  224.0.0.1
 The maximum transmit unit is 1500 bytes
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ICMP unreachables are always sent
 ICMP mask replies are always sent
 ARP timeout 01:00:00,  ARP retry interval 1s
 ARP Proxy is enabled, Local ARP Proxy is disabled
 VRRP master of :  VRRP is not configured on this interface
```

Use the following command to display the information of the arp entry on the switch:

```
Switch# show ip arp
 Protocol   Address        Age (min)  Hardware Addr   Interface
 Internet   192.168.10.1          -  7cc3.11f1.aa00  vlan10
 Internet   192.168.10.111        5  0cf9.11b6.6e2e  vlan10
 Internet   192.168.20.1          -  7cc3.11f1.aa00  vlan20
 Internet   192.168.20.222        6  5a94.031f.2357  vlan20
```

Use the following command to display the information on PC1:

```
[Host:~]$ ifconfig eth0
eth0     Link encap:Ethernet  HWaddr 0C:F9:11:B6:6E:2E
         inet addr:192.168.10.111  Bcast:192.168.255.255  Mask:255.255.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
         RX packets:11 errors:0 dropped:0 overruns:0 frame:0
         TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:588 (588.0 b)  TX bytes:700 (700.0 b)
         Interrupt:5

[Host:~]$ arp –a
? (192.168.20.222) at 7c:c3:11:f1:aa:00 [ether] on eth0
[Host: ~]$ route -v
Kernel IP routing table
Destination    Gateway       Genmask       Flags Metric Ref    Use Iface
192.168.0.0    *             255.255.0.0    U    0      0        0 eth0

[Host:~]$ ping 192.168.20.222
PING 192.168.20.222 (192.168.20.222) 56(84) bytes of data.
64 bytes from 192.168.20.222: icmp_seq=0 ttl=63 time=189 ms
64 bytes from 192.168.20.222: icmp_seq=1 ttl=63 time=65.2 ms
--- 192.168.20.222 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 65.209/127.226/189.244/62.018 ms, pipe 2
```

Use the following command to display the information on PC2:

```
[Host:~]$ ifconfig eth0
eth0     Link encap:Ethernet  HWaddr 5A:94:03:1F:23:57
```

```
      inet addr:192.168.20.222  Bcast:192.168.255.255  Mask:255.255.0.0
      UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
      RX packets:14 errors:0 dropped:0 overruns:0 frame:0
      TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:784 (784.0 b)  TX bytes:1174 (1.1 KiB)
      Interrupt:5

[Host:~]$ arp -a
? (192.168.10.111) at 7c:c3:11:f1:aa:00 [ether] on eth0

[Host: ~]$ route -v
Kernel IP routing table
Destination    Gateway       Genmask       Flags Metric Ref    Use Iface
192.168.0.0    *             255.255.0.0    U    0      0        0 eth0

[Host: ~]$ ping 192.168.10.111
PING 192.168.10.111 (192.168.10.111) 56(84) bytes of data.
64 bytes from 192.168.10.111: icmp_seq=0 ttl=63 time=53.8 ms
64 bytes from 192.168.10.111: icmp_seq=1 ttl=63 time=65.8 ms
--- 192.168.10.111 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 53.832/59.842/65.852/6.010 ms, pipe 2
```

Configuring Local ARP Proxy

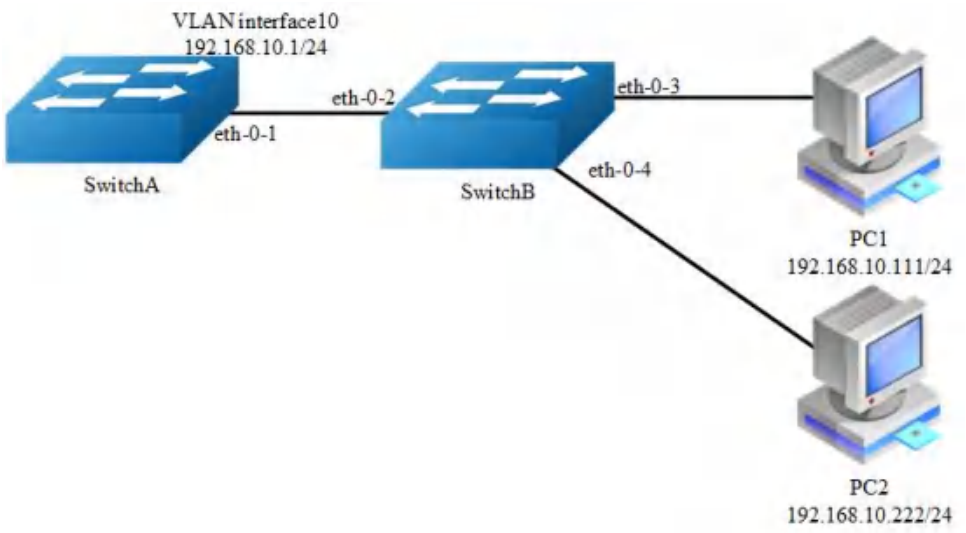


Figure 4-3 local arp proxy

As the above topology, eth-0-2, eth-0-3 and eth-0-4 are belonging to VLAN 10. eth-0-3 and eth-0-4 are both in port isolate group 1, and eth-0-2 is in port isolate group 3, so packets received in eth-0-3 can not flood to eth-0-4, but packets received in eth-0-2 can flood to both eth-0-3 and eth-0-4. PC1 is connecting with port eth-0-3 and PC2 is connecting with port eth-0-4.Configure as the following step for communicating with PC1 and PC2.

The configurations of switch A and switch B are same if there is no special description.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode, set the switch port mode and bind to the vlan

Switch A configuration:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Switch B configuration:

```
Switch(config)# interface range eth-0-2 - 4
Switch(config-if-range# switchport access vlan 10
Switch(config-if-range# no shutdown
Switch(config-if-range# exit
```

step 4 Create the vlan interface, configure the ip address, and enable local arp proxy

Switch A configuration:

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.10.1/24
Switch(config-if)# local-proxy-arp enable
Switch(config-if)# exit
```

step 5 Configuring port isolation(optional)

Switch B configuration:

After configuring port isolation as blow, eth-0-3 and eth-0-4 on swichB are isolated in layer 2 network.

```
Switch(config)# port-isolate mode l2
Switch(config)# interface eth-0-3 - 4
Switch(config-if-range# port-isolate group 1
Switch(config-if-range# exit

Switch(config)# interface eth-0-2
Switch(config-if)# port-isolate group 3
Switch(config-if)# exit
```

step 6 Validation

Use the following command to display the information of the arp entry on switchA:

```
Switch# show ip arp
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   192.168.10.1          -   eeb4.2a8d.6c00  vlan10
 Internet   192.168.10.111        0   34b0.b279.5f67  vlan10
 Internet   192.168.10.222        0   2a65.9618.57fa  vlan10
```

Use the following command to display the information of the arp configurations on the interface of switchA:

```
Switch# show ip interface vlan 10
Interface vlan10
 Interface current state: UP
 Internet address(es):
  192.168.10.1/24 broadcast 192.168.10.255
 Joined group address(es):
```

```
 224.0.0.1
The maximum transmit unit is 1500 bytes
ICMP error messages limited to one every 1000 milliseconds
ICMP redirects are never sent
ICMP unreachables are always sent
ICMP mask replies are always sent
ARP timeout 01:00:00,  ARP retry interval 1s
ARP Proxy is disabled, Local ARP Proxy is enabled
VRRP master of :  VRRP is not configured on this interface
```

Use the following command to display the information on PC1:

```
[Host: ~]$ ifconfig eth0
eth0     Link encap:Ethernet  HWaddr 34:B0:B2:79:5F:67
         inet addr:192.168.10.111  Bcast:192.168.10.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
         RX packets:22 errors:0 dropped:0 overruns:0 frame:0
         TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1344 (1.3 KiB)  TX bytes:2240 (2.1 KiB)
         Interrupt:5

[Host: ~]$ arp -a
? (192.168.10.222) at ee:b4:2a:8d:6c:00 [ether] on eth0

[Host: ~]$ ping 192.168.10.222
PING 192.168.10.222 (192.168.10.222) 56(84) bytes of data.
64 bytes from 192.168.10.222: icmp_seq=0 ttl=63 time=131 ms
64 bytes from 192.168.10.222: icmp_seq=1 ttl=63 time=159 ms
--- 192.168.10.222 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 131.078/145.266/159.454/14.188 ms, pipe 2
```

Use the following command to display the information on PC2:

```
[Host:~]$ ifconfig eth0
eth0     Link encap:Ethernet  HWaddr 2A:65:96:18:57:FA
         inet addr:192.168.10.222  Bcast:192.168.10.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
         RX packets:19 errors:0 dropped:0 overruns:0 frame:0
         TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1148 (1.1 KiB)  TX bytes:1524 (1.4 KiB)
         Interrupt:5

[Host:~]$ arp -a
? (192.168.10.111) at ee:b4:2a:8d:6c:00 [ether] on eth0

[Host: ~]$ ping 192.168.10.111
PING 192.168.10.111 (192.168.10.111) 56(84) bytes of data.
64 bytes from 192.168.10.111: icmp_seq=0 ttl=63 time=198 ms
64 bytes from 192.168.10.111: icmp_seq=1 ttl=63 time=140 ms
64 bytes from 192.168.10.111: icmp_seq=2 ttl=63 time=146 ms
--- 192.168.10.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 140.196/161.959/198.912/26.267 ms, pipe 2
```

4.2.3 Application cases

N/A

4.3 ConfiguringARP host-route

4.3.1 Overview

Brief Introduction

The ARP host-route function is to convert the ARP entries on the device and redisitribute them to route protocols. The function is enabled under interfaces, and it will redisitribute out all available ARP entries. In route protocols, host-routes generated by this function will be seen as connected routes.

4.3.2 Configuration Examples

Configure ARP host-route

1. Topology

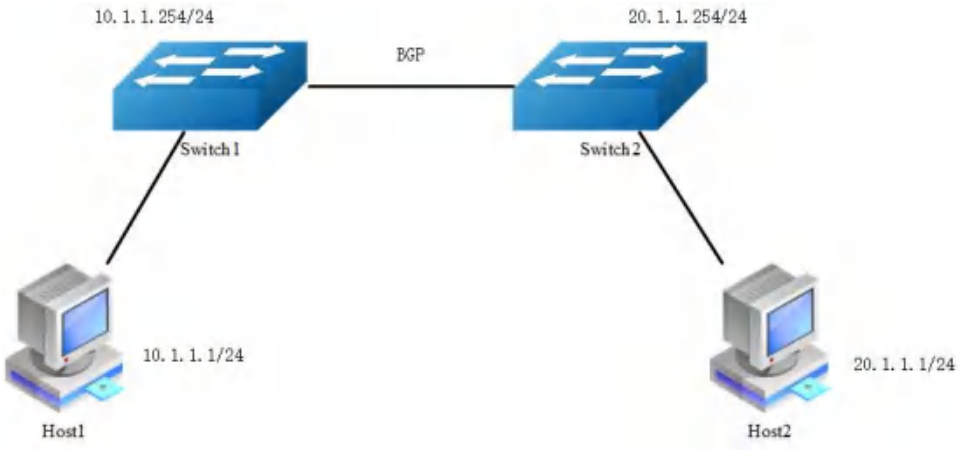


Figure 4-4 ARP host-route

In this configuration example, host1 and host2 connected to switch1 and switch2 individually through interface eth-0-1. Switch1 and switch2 built an ebgp connect. After enabling arp host-route funtion on interfaces connected to host, the switch can convert ARP entries to host-routes and redisitribute them to route protocols. With route-map, switch can purely advertise host route without network route, reducing ineffective flows due to addvertising network route.

2. Configuration Steps

3. step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the layer 3 interface and set the ip address

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.1.254/24
```

step 3 Enable arp host-route function under the interface

```
Switch(config-if)# arp host-route enable
Switch(config-if)# exit
```

step 4 Add a static arp entry

```
Switch(config)# arp 10.1.1.2 1.1.1
```

step 5 Enable BGP redistributing connected routes

```
Switch(config)# router bgp 100
Switch(config-router)# redistribute connected
```

step 6 Exiting the cofigure mode

```
Switch(config-router)# end
```

step 7 Validation

Use the following command to display the route information in BGP

```
Switch# show ip bgp
BGP table version is 1, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
l - labeled
        S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network       Next Hop        Metric LocPrf Weight Path
*> 10.1.1.0/24   0.0.0.0                  32768 ?
*> 10.1.1.2/32   0.0.0.0                  32768 ?

Total number of prefixes 2
```

4.4 Configuring DHCP Client

4.4.1 Overview

Function Introduction

Dynamic Host Configuration Protocol(DHCP) client can acquire IP address and configuration dynamically from DHCP server by DHCP. If client and server is on the same physical subnet, client can communicate with server directly, otherwise they need DHCP relay agent which is used to forward DHCP messages. DHCP client can request IP address from DHCP server by broadcasting DHCP messages. After received IP address and lease correspond to it, client will configure itself and set the expired time. When half past the lease, client will sent DHCP messages for a new lease to use the IP address continually. If it success, DHCP client will renew the lease. DHCP client can send option request to server, which may be one or several of router, static-route, classless-static-route, classless-static-route-ms, tftp-server-address, dns-nameserver , domain-name, netbios-nameserver and vendor-specific. By default, options include router, static-route, classless-static-route, classless-static-route-ms, tftp-server-address will be requested from server. We can cancel one or several of these option requests by command.

Principle Description

N/A

4.4.2 Configuration



Figure 4-5 dhcp client

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
```

step 3 disable static-route and enable DHCP client

```
Switch(config-if)# no dhcp client request static-route
Switch(config-if)# ip address dhcp
```

step 4 Exit the configure mode

```
Switch(config-if)# end
```

step 5 Validation

Check interface configuration:

```
Switch# show running-config interface eth-0-1
Building configuration...
!
interface eth-0-1
 no switchport
 ip address dhcp
 no dhcp client request static-route
!
```

Check all DHCP client status:

```
Switch# show dhcp client verbose
DHCP client informations:
============================================================
eth-0-1 DHCP client information:
 Current state: BOUND
 Allocated IP: 4.4.4.199 255.255.255.0
 Lease/renewal/rebinding: 1187/517/1037 seconds
 Lease from 2011-11-18 05:59:59 to 2011-11-18 06:19:59
 Will Renewal in 0 days 0 hours 8 minutes 37 seconds
 DHCP server: 4.4.4.1
 Transaction ID: 0x68857f54
 Client ID: switch-7e39.3457.b700-eth-0-1
```

Show DHCP client statistics:

```
Switch# show dhcp client statistics
DHCP client packet statistics:
============================================================
DHCP OFFERS    received: 1
DHCP ACKs      received: 2
DHCP NAKs      received: 0
DHCP Others    received: 0
DHCP DISCOVER      sent: 1
DHCP DECLINE       sent: 0
```

```
DHCP RELEASE      sent: 0
DHCP REQUEST      sent: 2
DHCP packet send failed: 0
```

4.4.3 Application cases

N/A

4.5 Configuring DHCP Relay

4.5.1 Overview

Function Introduction

DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router, where IP datagram are switched between networks somewhat transparently. By contrast, relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface. The relay agent sets the gateway address (girder field of the DHCP packet) and, if configured, adds the relay agent information option (option82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing option 82.

Principle Description

N/A

4.5.2 Configuration

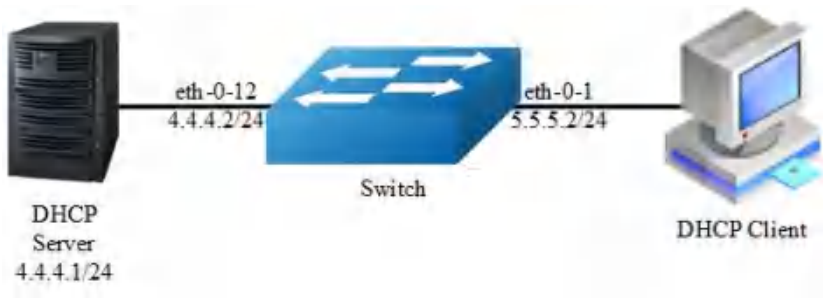


Figure 4-6 DHCP relay

This figure is the networking topology for testing DHCP relay functions. We need two Linux boxes and one Switch to construct the test bed.

Computer A is used as DHCP server.

Computer B is used as DHCP client.

Switch is used as DHCP relay agent.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode，set the attributes and ip address

```
Switch(config)# interface eth-0-12
Switch(config-if)# no switchport
Switch(config-if)# ip address 4.4.4.2/24
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-1
```

```
Switch(config-if)# no switchport
Switch(config-if)# ip address 5.5.5.2/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create a dhcp server

```
Switch(config)# dhcp-server 1 4.4.4.1
```

step 4 Enable DHCP server and option82 for the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# dhcp relay information trusted
Switch(config-if)# dhcp-server 1
Switch(config-if)# exit
```

step 5 Enable DHCP server and DHCP relay globally

```
Switch(config)# service dhcp enable
Switch(config)# dhcp relay
```

step 6 Validation

Check the interface configuration

```
Switch# show running-config interface eth-0-12
!
interface eth-0-12
 no switchport
 ip address 4.4.4.2/24
!
Switch# show running-config interface eth-0-1
!
interface eth-0-1
 no switchport
 dhcp relay information trusted
 dhcp-server 1
 ip address 5.5.5.2/24
!
```

Check the dhcp service status

```
Switch# show services
Networking services configuration:
Service Name       Status
============================================================
dhcp               enable
```

Check the dhcp server group configuration

```
Switch# show dhcp-server
DHCP server group information:
============================================================
group 1 ip address list:
[1] 4.4.4.1
```

Check the dhcp relay statistics

```
Switch# show dhcp relay statistics
DHCP relay packet statistics:
============================================================
```

```
Client relayed packets: 20
Server relayed packets: 20
Client error packets:   20
Server error packets:   0
Bogus GIADDR drops:     0
Bad circuit ID packets: 0
Corrupted agent options: 0
Missing agent options:  0
Missing circuit IDs:    0
```

Check your computer ip address from DHCP server

```
Ipconfig /all
Dhcp Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . . . . . . . . : 5.5.5.1
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 5.5.5.2
DHCP Server . . . . . . . . . . . : 4.4.4.1
DNS Servers . . . . . . . . . . . : 4.4.4.1
```

4.5.3 Application cases

N/A

4.6 Configuring DHCP server

4.6.1 Overview

Function Introduction

A DHCP server is an Internet host that returns configuration parameters to DHCP clients 。 DHCP server can provide IP address and network configuration for DHCP client by DHCP. For provide DHCP service， DHCP server need to be configured first. For example, IP address pool need be create , default gateway should be set in a pool, and some network parameters for DHCP client should be set before DHCP working. After DHCP server start to work, it will find a valid IP address from pool for DHCP client when receiving client's request. Meantime it also send network configuration parameters to client. The IP address assigned by DHCP server have a period of validity(lease), so DHCP client need to renew its lease before the lease expired for reserving current IP address by sending DHCP REQUEST message.

If DHCP server was in the same subnet with client,it can normal work after connect to subnet. Otherwise DHCP relay was needed for server providing DHCP service ,which can help to forward DHCP message between server and client.

Main options supported by DHCP server include bootfile-name, dns-server, domain-name, gateway, netbios-name-server, netbios-node-type, tftp-server-address. Besides these, some raw options were also be supported . Options with specified command line or options that the dhcp server does not need to support are not supported to be configure by the "option" CLI. These unsupported options are 3, 6, 15, 44, 46, 50, 51, 52, 53, 54, 55, 57, 58, 59, 61, 67, 82 and 150.

Principle Description

N/A

4.6.2 Configuration

Configuring DHCP server

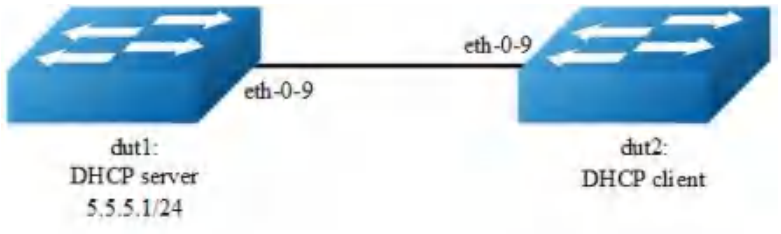


Figure 4-7 DHCP server

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable DHCP server globally， configure the ip address pool

Configure on DUT1：

```
Switch(config)#service dhcp enable
Switch(config)#dhcp server
Switch(config)#dhcp pool pool5
Switch(dhcp-config)#network 5.5.5.0/24
Switch(dhcp-config)#gateway 5.5.5.1
Switch(dhcp-config)#exit
```

step 3 Enter the interface configure mode，set the attributes and ip address

Configure on DUT1：

```
Switch(config)#interface eth-0-9
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address 5.5.5.1/24
Switch (config-if)# dhcp server enable
Switch (config-if)#exit
```

Configure on DUT2：

```
Switch#configure terminal
Switch(config)#interface eth-0-9
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address dhcp
Switch (config-if)#exit
```

step 4 Validation

Check DHCP Server(dut1) configuration：

```
Switch# show running-config
!
service dhcp enable
!
interface eth-0-9
 no switchport
 dhcp server enable
 ip address 5.5.5.1/24!
!
dhcp server
```

```
dhcp pool pool5
 network 5.5.5.0/24
 gateway 5.5.5.1
```

Check DHCP client status on DHCP Client(dut2):

```
Switch# show dhcp client verbose
DHCP client informations:
============================================================
eth-0-9 DHCP client information:
 Current state: BOUND
 Allocated IP: 5.5.5.2 255.255.255.0
 Lease/renewal/rebinding: 1194/546/1044 seconds
 Lease from 2012-02-04 07:40:12 to 2012-02-04 08:00:12
 Will Renewal in 0 days 0 hours 9 minutes 6 seconds
 DHCP server: 5.5.5.1
 Transaction ID: 0x45b0b27b
 Default router: 5.5.5.1
 Classless static route:
  Destination: 5.5.4.0, mask: 255.255.255.0, Nexthop: 5.5.5.1
 TFTP server addresses: 5.5.5.3
Client ID: switch-6e6e.361f.8400-eth-0-9
```

Check DHCP server statistics on DHCP Server(dut1):

```
Switch# show dhcp server statistics
DHCP server packet statistics:
============================================================
Message Received:
BOOTREQUEST:  0
DHCPDISCOVER:  1
DHCPREQUEST:  1
DHCPDECLINE:  0
DHCPRELEASE:  0
DHCPINFORM:  0
Message Sent:
BOOTREPLY:  0
DHCPOFFER:  1
DHCPACK:  1
DHCPNAK:  0
```

Check DHCP server addresses and interfaces on DHCP Server(dut1):

```
Switch# show dhcp server binding all
 IP address     Client-ID/          Lease expiration          Type
            Hardware address
 5.5.5.2        6e:6e:36:1f:84:00    Sat 2012.02.04 08:00:12     Dynamic
Switch# show dhcp server interfaces
List of DHCP server enabled interface(s):
DHCP server service status: enabled
Interface Name
============================================================
eth-0-9
```

Configuring DHCP server with relay



Figure 4-8 DHCP relay

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable DHCP server globally， configure the ip address pool and DHCP relay

Configure on DUT1：

```
Switch(config)#service dhcp enable
Switch(config)#dhcp server
Switch(dhcp-config)#dhcp pool pool4
Switch(dhcp-config)#network 4.4.4.0/24
Switch(dhcp-config)#gateway 4.4.4.1
Switch(dhcp-config)#exit
```

Configure on DUT2：

```
Switch(config)#service dhcp enable
Switch(config)#dhcp relay
Switch(config)#dhcp-server 1 5.5.5.1
```

step 2 Add a ip route

Configure on DUT1：

```
Switch(config)#ip route 4.4.4.0/24 5.5.5.2
```

step 4 Enter the interface configure mode，set the attributes and ip address

Configure on DUT1：

```
Switch(config)#interface eth-0-9
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address 5.5.5.1/24
Switch (config-if)# dhcp server enable
Switch (config-if)#exit
```

Configure on DUT2：

```
Switch(config)#interface eth-0-17
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address 4.4.4.1/24
Switch (config-if)# dhcp-server 1

Switch (config-if)#interface eth-0-9
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address 5.5.5.2/24
Switch (config-if)#exit
```

Configure on DUT3：

```
Switch(config)#interface eth-0-17
Switch (config-if)#no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ip address dhcp
Switch (config-if)#exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Check DHCP Server(dut1) configuration:

```
Switch# show running-config
!
service dhcp enable
!
interface eth-0-9
 no switchport
 dhcp server enable
 ip address 5.5.5.1/24!
!
ip route 4.4.4.0/24 5.5.5.2
!
dhcp server
dhcp pool pool4
 network 4.4.4.0/24
 gateway 4.4.4.1
```

Check DHCP client status on DHCP Server(dut1):

```
Switch# show dhcp client verbose
DHCP client informations:
============================================================
eth-0-17 DHCP client information:
 Current state: BOUND
 Allocated IP: 4.4.4.5 255.255.255.0
 Lease/renewal/rebinding: 1199/517/1049 seconds
 Lease from 2012-02-06 05:23:09 to 2012-02-06 05:43:09
 Will Renewal in 0 days 0 hours 8 minutes 37 seconds
 DHCP server: 5.5.5.1
 Transaction ID: 0x192a4f7d
 Default router: 4.4.4.1
 Classless static route:
  Destination: 5.5.4.0, mask: 255.255.255.0, Nexthop: 4.4.4.1
 TFTP server addresses: 5.5.5.3
 Client ID: switch-3c9a.b29a.ba00-eth-0-17
```

Check DHCP server statistics on DHCP Server(dut1):

```
Switch# show dhcp server statistics
DHCP server packet statistics:
============================================================
Message Received:
BOOTREQUEST:  0
DHCPDISCOVER:  1
```

```
DHCPREQUEST: 1
DHCPDECLINE: 0
DHCPRELEASE: 0
DHCPINFORM: 0
Message Sent:
BOOTREPLY: 0
DHCPOFFER: 1
DHCPACK: 1
DHCPNAK: 0
```

Check DHCP server addresses and interfaces on DHCP Server(dut1):

```
Switch# show dhcp server binding all
IP address     Client-ID/          Lease expiration          Type
               Hardware address
 4.4.4.5       3c:9a:b2:9a:ba:00   Mon 2012.02.06 05:43:09   Dynamic
Switch# show dhcp server interfaces
List of DHCP server enabled interface(s):
DHCP server service status: enabled
Interface Name
============================================================
eth-0-9
```

4.6.3 Application cases

N/A

4.7 Configuring DNS

4.7.1 Overview

Function Introduction

The DNS protocol controls the Domain Name System (DNS), a distributed database with which you can map hostnames to IP addresses. When you configure DNS on your switch, you can substitute the hostname for the IP address with all IP commands, such as ping, telnet, connect, and related Telnet support operations. IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. To keep track of domain names, IP has defined the concept of a domain name server, which holds a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the hostnames, specify the name server that is present on your network, and enable the DNS.

Principle Description

N/A

4.7.2 Configuration

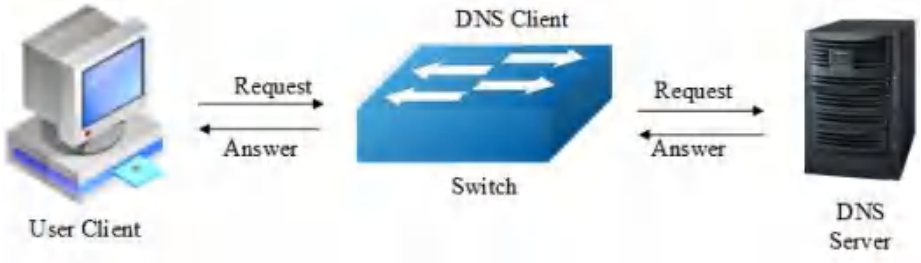


Figure 4-9 DNS

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the dns domain name and dns server address

```
Switch(config)#dns domain server1
Switch(config)#dns server 202.100.10.20
```

step 3 Set static hostname-to-address mappings (optional)

```
Switch(config)# ip host www.example1.com 192.0.2.141
```

step 4 Validation

```
Switch# show dns server
Current DNS name server configuration:
   Server             IP Address
------------------------------------------------------------
1    nameserver          202.100.10.20
```

4.7.3 Application cases

N/A

5 IP Routing Configuration Guide

5.1 Configuring IP Unicast-Routing

5.1.1 Overview

Function Introduction

Static routing is a concept describing one way of configuring path selection of routers in computer networks. It is the type of routing characterized by the absence of communication between routers regarding the current topology of the network. This is achieved by manually adding routes to the routing table. The opposite of static routing is dynamic routing, sometimes also referred to as adaptive routing.

In these systems, routes through a data network are described by fixed paths (statically). These routes are usually entered into the router by the system administrator. An entire network can be configured using static routes, but this type of configuration is not fault tolerant. When there is a change in the network or a failure occurs between two statically defined nodes, traffic will not be rerouted. This means that anything that wishes to take an affected path will either have to wait for the failure to be repaired or the static route to be updated by the administrator before restarting its journey. Most requests will time out (ultimately failing) before these repairs can be made. There are, however, times when static routes can improve the performance of a network. Some of these include stub networks and default routes.

Principle Description

N/A

5.1.2 Configuration

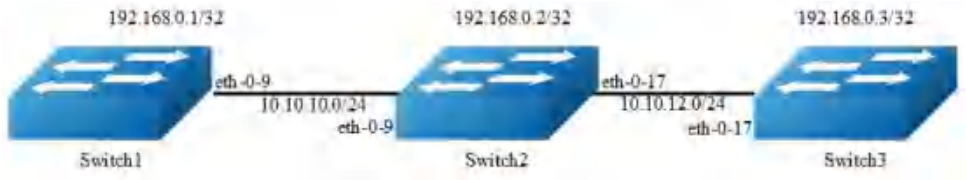


Figure 5-1 ip unicast routing

This example shows how to enable static route in a simple network topology.

There are 3 static routes on Switch1， one is to achieve remote network 10.10.12.0/24, the other two are to achieve the loopback addresses on Switch2 and Switch3. There is a default static route on Switch3, that is, static routes use same gateway or nexthop address. There are 2 static routes on swithc2, both of them are to achieve the remote switch's loopback address.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode，set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.1/24
Switch(config-if)# exit

Switch(config)# interface loopback 0
Switch(config-if)# ip address 192.168.0.1/32
Switch(config-if)# exit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.2/24
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.12.2/24
Switch(config-if)# exit

Switch(config)# interface loopback 0
Switch(config-if)# ip address 192.168.0.2/32
Switch(config-if)# exit
```

Configure on Switch3:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.12.3/24
Switch(config-if)# exit

Switch(config)# interface loopback 0
Switch(config-if)# ip add 192.168.0.3/32
Switch(config-if)# exit
```

step 3 Configuring static route

Configure on Switch1:

Note:Specify the destination prefix and mask for the network for which a gateway is required, for example, 10.10.12.0/24. Add a gateway for each of them (in this case 10.10.10.2 for all). Since R2 is the only next hop available, you can configure a default route instead of configuring the same static route for individual addresses.

```
Switch(config)# ip route 10.10.12.0/24 10.10.10.2
Switch(config)# ip route 192.168.0.2/32 10.10.10.2
Switch(config)# ip route 192.168.0.3/32 10.10.10.2
```

Configure on Switch2:

```
Switch(config)# ip route 192.168.0.1/32 10.10.10.1
Switch(config)# ip route 192.168.0.3/32 10.10.12.3
```

Configure on Switch3:

Note:Specify 10.10.12.2 as a default gateway to reach any network. Since 10.10.12.2 is the only route available you can specify it as the default gateway instead of specifying it as the gateway for individual network or host addresses.

```
Switch(config)# ip route 0.0.0.0/0 10.10.12.2
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the route information on Switch1:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
```

```
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
C     10.10.10.0/24 is directly connected, eth-0-9
C     10.10.10.1/32 is in local loopback, eth-0-9
S     10.10.12.0/24 [1/0] via 10.10.10.2, eth-0-9
C     192.168.0.1/32 is directly connected, loopback0
S     192.168.0.2/32 [1/0] via 10.10.10.2, eth-0-9
S     192.168.0.3/32 [1/0] via 10.10.10.2, eth-0-9
```

Use the following command to display the route information on Switch2:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
C     10.10.10.0/24 is directly connected, eth-0-9
C     10.10.10.2/32 is in local loopback, eth-0-9
C     10.10.12.0/24 is directly connected, eth-0-17
C     10.10.12.2/32 is in local loopback, eth-0-17
S     192.168.0.1/32 [1/0] via 10.10.10.1, eth-0-9
C     192.168.0.2/32 is directly connected, loopback0
S     192.168.0.3/32 [1/0] via 10.10.12.3, eth-0-17
```

Use the following command to display the route information on Switch3:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
Gateway of last resort is 10.10.12.2 to network 0.0.0.0
S*    0.0.0.0/0 [1/0] via 10.10.12.2, eth-0-17
C     10.10.12.0/24 is directly connected, eth-0-17
C     10.10.12.3/32 is in local loopback, eth-0-17
C     192.168.0.3/32 is directly connected, loopback0
```

5.1.3 Application cases

N/A

5.2 Configuring RIP

5.2.1 Overview

Function Introduction

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing distance) to measure the cost of a given route. The cost is a distance vector because the cost is often equivalent to the number of router hops between the source and the destination networks. RIP can receive multiple paths to a destination. The system evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If RIP receives a RIP update from another router that contains a path with fewer hops than the path stored in the route table, the system replaces the older route with the newer one. The system then includes the new path in the updates it sends to other RIP routers. RIP routers also can modify a route's cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes. A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

This chapter contains basic RIP configuration examples. To see details on the commands used in these examples, or to see the outputs of the Validation commands, refer to the RIP Command Reference. To avoid repetition, some Common commands, like configure terminal, have not been listed under the Commands Used section.

Principle Description

Reference to RFC 2453

5.2.2 Configuration

Enabling RIP

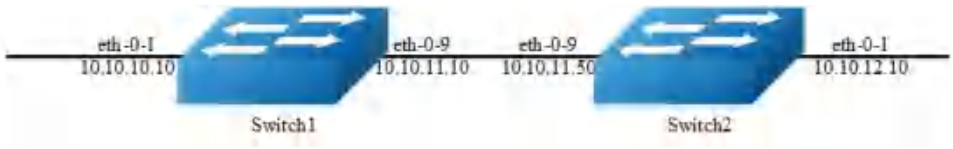


Figure 5-2 enable rip

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.10/24
Switch(config-if)# exit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.12.10/24
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
```

```
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.50/24
Switch(config-if)# exit
```

step 3 Enable RIP routing process and associate networks

Configure on Switch1:

```
Switch(config)# router rip
Switch(config-router)#network 10.10.10.0/24
Switch(config-router)#network 10.10.11.0/24
Switch(config-router)# exit
```

Configure on Switch2:

```
Switch(config)# router rip
Switch(config-router)#network 10.10.11.0/24
Switch(config-router)#network 10.10.12.0/24
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the database of rip on Switch1:

```
Switch# show ip rip database
Codes： R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
    C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
  Network         Next Hop      Metric From         If     Time
Rc 10.10.10.0/24                   1              eth-0-1
Rc 10.10.11.0/24                   1              eth-0-9
R  10.10.12.0/24    10.10.11.50    2 10.10.11.50   eth-0-9 00：02：52
```

Use the following command to display the protocol state of rip process on Switch1:

```
Switch# show ip protocols rip
Routing protocol is "rip"
 Sending updates every 30 seconds with +/-5 seconds, next due in 17 seconds
 Timeout after 180 seconds, Garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Default redistribution metric is 1
 Redistributing：
 Default version control： send version 2, receive version 2
  Interface      Send       Recv   Key-chain
  eth-0-1        2          2
  eth-0-9        2          2
 Routing for Networks：
  10.10.10.0/24
  10.10.11.0/24
 Routing Information Sources：
  Gateway         Distance  Last Update  Bad Packets  Bad Routes
  10.10.11.50          120  00：00：22            0         0
 Number of routes (including connected)： 3
 Distance： (default is 120)
```

Use the following command to display the interface of rip on Switch1:

```
Switch# show ip rip interface
eth-0-1 is up, line protocol is up
 Routing Protocol: RIP
  Receive RIP packets
  Send RIP packets
  Passive interface: Disabled
  Split horizon: Enabled with Poisoned Reversed
  IP interface address:
   10.10.10.10/24
eth-0-9 is up, line protocol is up
 Routing Protocol: RIP
  Receive RIP packets
  Send RIP packets
  Passive interface: Disabled
  Split horizon: Enabled with Poisoned Reversed
  IP interface address:
   10.10.11.10/24
```

Use the following command to display routes on Switch1:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
C    10.10.10.0/24 is directly connected, eth-0-1
C    10.10.10.10/32 is in local loopback, eth-0-1
C    10.10.11.0/24 is directly connected, eth-0-9
C    10.10.11.10/32 is in local loopback, eth-0-9
R    10.10.12.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:25:50
```

Configuring The RIP Version



Figure 5-3 rip version

Configure the receive and send specific versions of packets on an interface .

In this example, Switch2 is configured to receive and send RIP version 1 and 2 on eth-0-9 and eth-0-20.

step 1 Enter the configure mode

The following commands operate on Switch2:

```
Switch# configure terminal
```

step 2 Enable RIP routing process

```
Switch(config)# router rip
Switch(config-router)# exit
```

step 3 Enter the interface configure mode and set the version for sending and receiving rip packets

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip rip send version 1 2
Switch(config-if)# ip rip receive version 1 2
Switch(config-if)# quit

Switch(config)# interface eth-0-20
Switch(config-if)# ip rip send version 1 2
Switch(config-if)# ip rip receive version 1 2
Switch(config-if)# quit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the configuration on Switch1:

```
Switch# show running-config
interface eth-0-9
 no switchport
 ip address 10.10.11.10/24
!
router rip
 network 10.10.11.0/24
```

Use the following command to display the database of rip on Switch2:

```
Switch# show ip rip database
Codes： R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
     C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
  Network          Next Hop       Metric From          If      Time
R  10.0.0.0/8                        1             eth-0-9
Rc 10.10.11.0/24                       1               eth-0-9
Rc 10.10.12.0/24                       1               eth-0-20
```

Use the following command to display the protocol state of rip process on Switch2:

```
Switch# show ip protocols rip
Routing protocol is "rip"
 Sending updates every 30 seconds with +/-5 seconds, next due in 1 seconds
 Timeout after 180 seconds, Garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Default redistribution metric is 1
 Redistributing：
 Default version control： send version 2, receive version 2
   Interface       Send        Recv   Key-chain
   eth-0-9         1 2         1 2
   eth-0-20        1 2         1 2
 Routing for Networks：
   10.10.11.0/24
   10.10.12.0/24
 Routing Information Sources：
   Gateway         Distance  Last Update  Bad Packets  Bad Routes
   10.10.11.10          120  00：00：22            0         0
   10.10.12.50          120  00：00：27            0         0
 Number of routes (including connected)： 3
 Distance： (default is 120)
```

Use the following command to display the interface of rip on Switch2:

```
Switch# show ip rip interface
eth-0-9 is up, line protocol is up
 Routing Protocol： RIP
   Receive RIPv1 and RIPv2 packets
   Send RIPv1 and RIPv2 packets
   Passive interface： Disabled
   Split horizon： Enabled with Poisoned Reversed
   IP interface address：
    10.10.11.50/24
eth-0-20 is up, line protocol is up
 Routing Protocol： RIP
   Receive RIPv1 and RIPv2 packets
   Send RIPv1 and RIPv2 packets
   Passive interface： Disabled
   Split horizon： Enabled with Poisoned Reversed
   IP interface address：
    10.10.12.10/24
```

Use the following command to display the configuration on Switch2:

```
Switch# show run
interface eth-0-9
 no switchport
 ip address 10.10.11.50/24
 ip rip send version 1 2
 ip rip receive version 1 2
!
interface eth-0-20
 no switchport
 ip address 10.10.12.10/24
 ip rip send version 1 2
 ip rip receive version 1 2
!
router rip
 network 10.10.11.0/24
 network 10.10.12.0/24
```

Use the following command to display the configuration on Switch3:

```
Switch# show running-config
interface eth-0-20
 no switchport
 ip address 10.10.12.50/24
!
router rip
 network 10.10.12.0/24
```

Configuring Metric Parameters

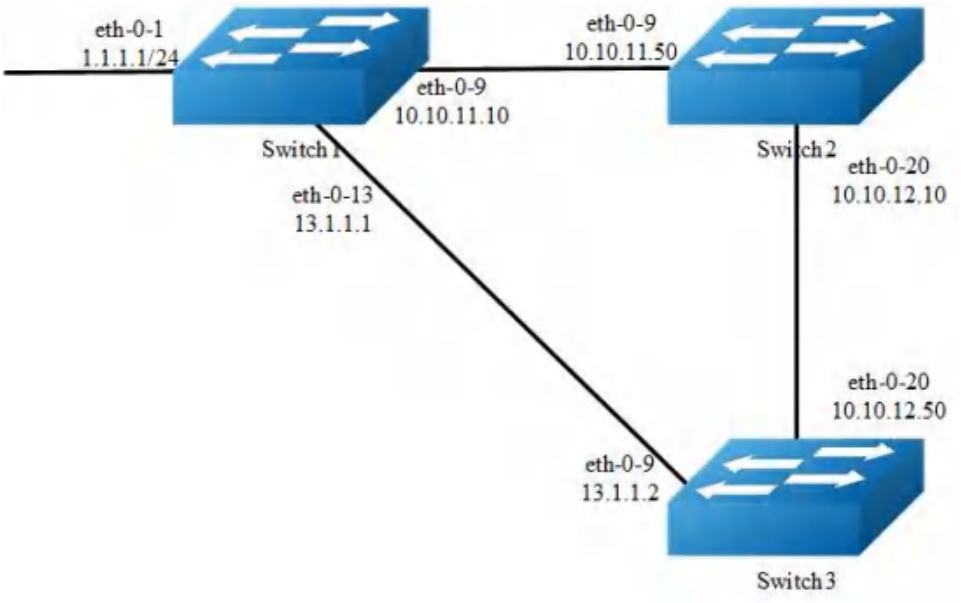


Figure 5-4 rip metric

A RIP offset list allows you to add to the metric of specific inbound or outbound routes learned or advertised by RIP. RIP offset lists provide a simple method for adding to the cost of specific routes and therefore biasing the router's route selection away from those routes. An offset list consists of the following parameters:

- An ACL that specifies the routes to which to add the metric. The direction:
- In: applies to routes the router learns from RIP neighbors.
- Out: applies to routes the router is advertising to its RIP neighbors.
- The offset value that will be added to the routing metric of the routes that match the ACL.
- The interface that the offset list applies (optional).

If a route matches both a global offset list (without specified interface) and an interface-based offset list, the interface-based offset list takes precedence. The interface-based offset list's metric is added to the route in this case.

This example Switch1 will advertise route 1.1.1.0 out of int eth-0-13 with metric 3.

step 1 precondition

Switch1

```
interface eth-0-1
no switchport
ip address 1.1.1.1/24
!
interface eth-0-9
no switchport
ip address 10.10.11.10/24
!
interface eth-0-13
no switchport
ip address 13.1.1.1/24
!
router rip
network 1.1.1.0/24
network 10.10.11.0/24
network 13.1.1.0/24
```

Switch2

```
interface eth-0-9
no switchport
ip address 10.10.11.50/24
!
interface eth-0-20
no switchport
ip address 10.10.12.10/24
!
router rip
network 10.10.11.0/24
network 10.10.12.0/24
```

Switch3

```
interface eth-0-13
no switchport
ip address 13.1.1.2/24
!
interface eth-0-20
no switchport
ip address 10.10.12.50/24
!
router rip
network 10.10.12.0/24
network 13.1.1.0/24
```

Display the routes on Switch3:

```
Switch# show ip route rip
R     1.1.1.0/24 [120/2] via 13.1.1.1, eth-0-13, 00：07：46
R     10.10.11.0/24 [120/2] via 13.1.1.1, eth-0-13, 00：07：39
            [120/2] via 10.10.12.10, eth-0-20, 00：07：39
Change router 1.1.1.0/24 via 10.10.12.10
```

step 2 Enter the configure mode

The following commands operate on Switch1:

```
Switch# configure terminal
```

step 3 Configuring access list

```
Switch(config)#ip access-list ripoffset
Switch(config-ip-acl)#permit any 1.1.1.0 0.0.0.255 any
```

step 4 Enable RIP routing process and set offset list and offset value for an interface

```
Switch(config-ip-acl)# router rip
Switch(config-router)# offset-list ripoffset out 3 eth-0-13
```

step 5 Exit the configure mode

```
Switch(config-router)# end
```

step 6 Validation

Display the routes on Switch3. The metric for the route which distributed by Switch1 is 3 now.

```
Switch# show ip route rip
R     1.1.1.0/24 [120/3] via 10.10.12.10, eth-0-20, 00：00：02
```

```
R     10.10.11.0/24 [120/2] via 13.1.1.1, eth-0-13, 00：11：40
            [120/2] via 10.10.12.10, eth-0-20, 00：11：40
```

Configuring the Administrative Distance



Figure 5-5 rip distance

By default, RIP assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the router selects the route with the lower distance. You can change the administrative distance for RIP routes.

This example all Switches have two router protocols, RIP and OSPF, OSPF route has higher priority, Switch3 will change route 1.1.1.0 with administrative distance 100.

step 1 precondition

Switch1

```
interface eth-0-1
no switchport
ip address 1.1.1.1/24
!
interface eth-0-9
no switchport
ip address 10.10.11.10/24
!
router ospf
network 1.1.1.0/24 area 0
network 10.10.11.0/24 area 0
!
router rip
network 1.1.1.0/24
network 10.10.11.0/24
```

Switch2

```
interface eth-0-9
no switchport
ip address 10.10.11.50/24
!
interface eth-0-20
no switchport
ip address 10.10.12.10/24
!
router ospf
network 10.10.11.0/24 area 0
network 10.10.12.0/24 area 0
!
router rip
network 10.10.11.0/24
network 10.10.12.0/24
```

Switch3

```
interface eth-0-20
no switchport
ip address 10.10.12.50/24
!
router ospf
network 10.10.12.0/24 area 0
!
router rip
network 10.10.12.0/24
```

Display the routes on Switch3:

```
Switch# show ip route
Codes： K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
O     1.1.1.0/24 [110/3] via 10.10.12.10, eth-0-20, 01：05：49
O     10.10.11.0/24 [110/2] via 10.10.12.10, eth-0-20, 01：05：49
C     10.10.12.0/24 is directly connected, eth-0-20
C     10.10.12.50/32 is in local loopback, eth-0-20
```

step 2 Enter the configure mode

The following commands operate on Switch3:

```
Switch# configure terminal
```

step 3 Configuring access list

```
Switch(config)#ip access-list ripdistancelist
Switch(config-ip-acl)#permit any 1.1.1.0 0.0.0.255 any
```

step 4 Enable RIP routing process and set administrative distance

```
Switch(config-ip-acl)# router rip
Switch(config-router)# distance 100 0.0.0.0/0 ripdistancelist
```

step 5 Exit the configure mode

```
Switch(config-router)# end
```

step 6 Validation

Display the routes on Switch3. The distance for the rip route is 100 now.

```
Switch# show ip route
Codes： K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
R     1.1.1.0/24 [100/3] via 10.10.12.10, eth-0-20, 00：00：02
O     10.10.11.0/24 [110/2] via 10.10.12.10, eth-0-20, 01：10：42
```

```
C    10.10.12.0/24 is directly connected, eth-0-20
C    10.10.12.50/32 is in local loopback, eth-0-20
```

Configuring Redistribution

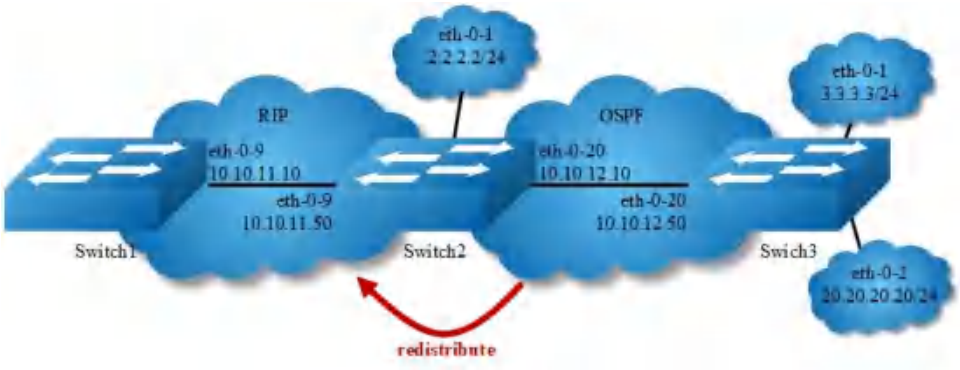


Figure 5-6 rip redistribute

You can configure the router to redistribute static routes, direct connected routes or routes learned through Open Shortest Path First (OSPF) into RIP. When you redistribute a route from one of these other protocols into RIP, the router can use RIP to advertise the route to its RIP neighbors.

Change the default redistribution metric (optional). The router assigns a RIP metric of 1 to each redistributed route by default. You can change the default metric to a value up to 16.

Enable specified routes to redistribute with default or specified metric. This example the router will set the default metric to 2 for redistributed routes and redistributes static routes and direct connected routes to RIP with default metric 2, redistributes OSPF routes with specified metric 5.

step 1 precondition

Switch1

```
interface eth-0-9
no switchport
ip address 10.10.11.10/24
!
router rip
network 10.10.11.0/24
```

Switch2

```
interface eth-0-1
no switchport
ip address 2.2.2.2/24
!
interface eth-0-9
no switchport
ip address 10.10.11.50/24
!
interface eth-0-20
no switchport
ip address 10.10.12.10/24
!
router ospf
network 10.10.12.0/24 area 0
!
router rip
```

```
network 10.10.11.0/24
!
ip route 20.20.20.0/24 10.10.12.50
```

Switch3

```
interface eth-0-1
no switchport
ip address 3.3.3.3/24
!
interface eth-0-2
no switchport
ip address 20.20.20.20/24
!
interface eth-0-20
no switchport
ip address 10.10.12.50/24
!
router ospf
network 3.3.3.0/24 area 0
network 10.10.12.0/24 area 0
```

Display the routes on Switch1:

```
Switch# show ip route
Codes： K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
C    10.10.11.0/24 is directly connected, eth-0-9
C    10.10.11.10/32 is in local loopback, eth-0-9
```

Display the routes on Switch2:

```
Switch# show ip route
Codes： K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
C    2.2.2.0/24 is directly connected, eth-0-1
C    2.2.2.02/32 is in local loopback, eth-0-1
O    3.3.3.0/24 [110/2] via 10.10.12.50, eth-0-20, 01：05：41
C    10.10.11.0/24 is directly connected, eth-0-9
C    10.10.11.50/32 is in local loopback, eth-0-9
C    10.10.12.0/24 is directly connected, eth-0-20
C    10.10.12.10/24 is in local loopback, eth-0-20
S    20.20.20.0/24 [1/0] via 10.10.12.50, eth-0-20
```

step 2 Enter the configure mode

The following commands operate on Switch2:

```
Switch# configure terminal
```

step 3 Enable RIP routing process and set metric and enable redistribute

```
Switch(config)# router rip
Switch(config-router)# default-metric 2
Switch(config-router)# redistribute static
Switch(config-router)# redistribute connected
Switch(config-router)# redistribute ospf metric 5
```

redistribute connected routes by ospf (optional)

```
Switch(config)# router ospf
Switch(config-router)# redistribute connected
```

step 4 Exit the configure mode

```
Switch(config-router)# end
```

step 5 Validation

Display the routes on Switch1:

```
Switch# show ip route
Codes：  K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
R    2.2.2.0/24 [120/3] via 10.10.11.50, eth-0-9, 00：02：36
R    3.3.3.0/24 [120/6] via 10.10.11.50, eth-0-9, 00：02：26
C    10.10.11.0/24 is directly connected, eth-0-9
C    10.10.11.10/32 is in local loopback eth-0-9
R    10.10.12.0/24 [120/3] via 10.10.11.50, eth-0-9, 00：02：36
R    20.20.20.0/24 [120/3] via 10.10.11.50, eth-0-9, 00：02：41
```

Configuring Split-horizon Parameters



Figure 5-7 rip split-horizon

Normally, routers that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the split horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routers, particularly when links are broken. However, with non-broadcast networks (such as Frame Relay), situations can arise for which this behavior is less than ideal. For these situations, you might want to disable split horizon for RIP.

You can avoid including routes in updates sent to the same gateway from which they were learned. Using the split horizon command omits routes learned from one neighbor, in updates sent to that neighbor. Using the poisoned parameter with this command includes such routes in updates, but sets their metrics to infinity. Thus, advertising these routes means that they are not reachable.

step 1 precondition

Switch1

```
interface eth-0-1
no switchport
ip address 1.1.1.1/24
!
interface eth-0-9
no switchport
ip address 10.10.11.10/24
!
router rip
network 10.10.11.0/24
redistribute connected
```

Switch2

```
interface eth-0-9
no switchport
ip address 10.10.11.50/24
!
router rip
network 10.10.11.0/24
```

step 2 Enabling debug on Switch2 (optional)

```
Switch# debug rip packet send detail
Switch# terminal monitor
```

step 3 Enter the configure mode

The following commands operate on Switch2:

```
Switch# configure terminal
```

step 4 Enter the interface configure mode and set split-horizon

Disable Split-horizon:

```
Switch(config)#interface eth-0-9
Switch(config-if)# no ip rip split-horizon
```

If debug is enabled, the following messages will be shown:

```
Apr  8 06：24：25 Switch RIP4-7： SEND[eth-0-9]： Send to 224.0.0.9：520
Apr  8 06：24：25 Switch RIP4-7： SEND[eth-0-9]： RESPONSE version 2 packet size 44
Apr  8 06：24：25 Switch RIP4-7： 1.1.1.0/24 -> 0.0.0.0 family 2 tag 0 metric 2
Apr  8 06：24：25 Switch RIP4-7： 10.10.11.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
```

Enable Split-horizon and poisoned:

```
Switch(config-if)# ip rip split-horizon
Switch(config-if)# ip rip split-horizon poisoned
```

If debug is enabled, the following messages will be shown:

```
Apr  8 06：38：35 Switch RIP4-7： SEND[eth-0-9]： Send to 224.0.0.9：520
Apr  8 06：38：35 Switch RIP4-7： SEND[eth-0-9]： RESPONSE version 2 packet size 44
Apr  8 06：38：35 Switch RIP4-7： 1.1.1.0/24 -> 0.0.0.0 family 2 tag 0 metric 16
Apr  8 06：38：35 Switch RIP4-7： 10.10.11.0/24 -> 0.0.0.0 family 2 tag 0 metric 16
```

step 5 Exit the configure mode

```
Switch(config-router)# end
```

step 6 Validation

Use the following command to display the configuration:

```
Switch# show running-config
interface eth-0-9
 no switchport
 ip address 10.10.11.50/24
!
router rip
 network 10.10.11.0/24
!
```

Use the following command to display the interface of rip:

```
Switch# show ip rip interface
eth-0-9 is up, line protocol is up
 Routing Protocol： RIP
  Receive RIP packets
  Send RIP packets
  Passive interface： Disabled
  Split horizon： Enabled with Poisoned Reversed
  IP interface address：
   10.10.11.50/24
```

Configuring Timers

RIP use several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune RIP performance to better suit your internet-work needs. You can make the following timer adjustments:

- The rate (time in seconds between updates) at which routing updates are sent.
- The interval of time (in seconds) after which a route is declared invalid.
- The amount of time (in seconds) that must pass before a route is removed from the routing table.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable RIP routing process and set the timers

Specify the routing table update timer in 10 seconds. Specifies the routing information timeout timer in 180 seconds. Specifies the routing garbage collection timer in 120 seconds:

```
Switch(config)# router rip
Switch(config-router)# timers basic 10 180 120
```

step 3 Exit the configure mode

```
Switch(config-router)# end
```

step 4 Validation

Use the following command to display the protocol state of rip process:

```
Switch# show ip protocols rip
Routing protocol is "rip"
 Sending updates every 10 seconds with +/-5 seconds, next due in 2 seconds
 Timeout after 180 seconds, Garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
```

```
Incoming update filter list for all interface is not set
Default redistribution metric is 1
Redistributing:
Default version control: send version 2, receive version 2
  Interface      Send      Recv   Key-chain
  eth-0-9        2         2
Routing for Networks:
  10.10.11.0/24
Routing Information Sources:
  Gateway         Distance  Last Update  Bad Packets  Bad Routes
  10.10.11.50          120  00:00:02           0            0
Number of routes (including connected): 5
Distance: (default is 120)
```

Configuring RIP Route Distribute Filters

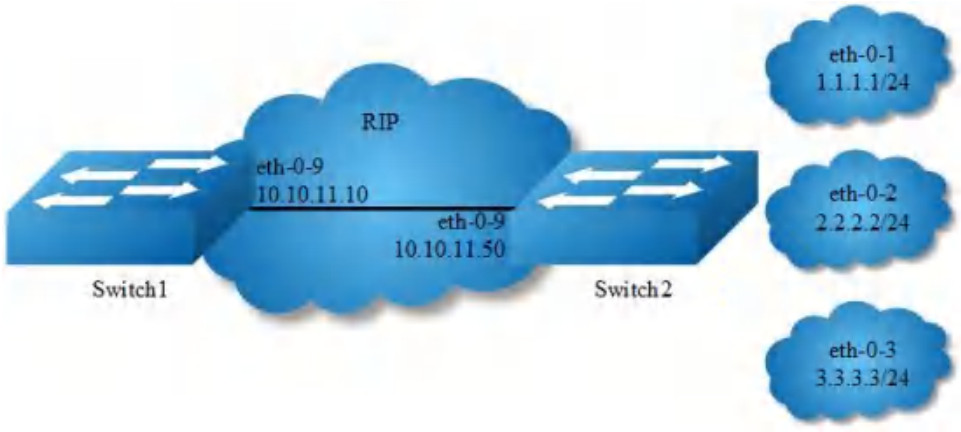


Figure 5-8 rip filter list

A RIP distribute list allows you to permit or deny learning or advertising of specific routes. A distribute list consists of the following parameters:

- An ACL or a prefix list that filter the routes.
- The direction:

  In: filter applies to learned routes.

  Out: filter applies to advertised routes
- The interface that the filer applies (optional).

step 1 precondition

Switch1

```
interface eth-0-9
no switchport
ip address 10.10.11.10/24
!
router rip
network 10.10.11.0/24
```

Switch2

```
interface eth-0-1
no switchport
```

```
ip address 1.1.1.1/24
!
interface eth-0-2
no switchport
ip address 2.2.2.2/24
!
interface eth-0-3
no switchport
ip address 3.3.3.3/24
!
interface eth-0-9
no switchport
ip address 10.10.11.50/24
!
router rip
network 1.1.1.0/24
network 2.2.2.0/24
network 3.3.3.0/24
network 10.10.11.0/24
```

Display the routes on Switch1:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
R     1.1.1.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:01:50
R     2.2.2.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:01:50
R     3.3.3.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:01:50
C     10.10.11.0/24 is directly connected, eth-0-9
C     10.10.11.10/32 is in local loopback, eth-0-9
```

step 2 Enter the configure mode

The following commands operate on Switch2:

```
Switch# configure terminal
```

step 3 Configuring prefix list

```
Switch(config)# ip prefix-list 1 deny 1.1.1.0/24
Switch(config)# ip prefix-list 1 permit any
```

step 4 Apply prefix list

```
Switch(config)# router rip
Switch(config-router)# distribute-list prefix 1 out
```

step 5 Exit the configure mode

```
Switch(config-router)# end
```

step 6 Validation

Display the routes on Switch1:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
    * - candidate default
R     2.2.2.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:00:08
R     3.3.3.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:00:08
C     10.10.11.0/24 is directly connected, eth-0-9
C     10.10.11.10/32 is in local loopback, eth-0-9
```

Configuring RIPv2 authentication (single key)



Figure 5-9 rip authentication

RIPv2 supports 2 authentication methods: plaintext and MD5 encryption.

The following example shows how to enable plaintext authentication.

To using this feature, the following steps are required:

- Specify an interface and set the authentication string
- Specify the authentication mode as "text"

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit

Switch(config-if)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.10/24
Switch(config-if)# exit
```

Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
```

```
Switch(config-if)# ip address 2.2.2.2/24
Switch(config-if)# exit

Switch(config-if)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.50/24
Switch(config-if)# exit
```

step 3 Enable RIP routing process and set the parameters

```
Switch(config)# router rip
Switch(config-router)# network 10.10.11.0/24
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

step 4 Specify the authentication string and mode

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip rip authentication string Auth1
Switch(config-if)# ip rip authentication mode text
```

step 5 Exit the configure mode

```
Switch(config-if)# end
```

step 6 Validation

Use the following command to display the database of rip:

```
Switch# show ip rip database

Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
    C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP

  Network         Next Hop      Metric From          If     Time
R  2.2.2.0/24      10.10.11.50       2 10.10.11.50    eth-0-9  00:02:52
Rc 10.10.11.0/24
```

Use the following command to display the protocol state of rip process:

```
Switch# show ip protocols rip
Routing protocol is "rip"
 Sending updates every 30 seconds with +/-5 seconds, next due in 23 seconds
 Timeout after 180 seconds, Garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Default redistribution metric is 1
 Redistributing:
  connected  metric default
 Default version control: send version 2, receive version 2
  Interface      Send       Recv   Key-chain
  eth-0-9        2          2
 Routing for Networks:
  10.10.11.0/24
 Routing Information Sources:
  Gateway        Distance  Last Update  Bad Packets  Bad Routes
  10.10.11.50         120  00:00:45             1         0
 Number of routes (including connected): 2
```

```
 Distance: (default is 120)



Switch# show ip rip interface
eth-0-9 is up, line protocol is up
 Routing Protocol: RIP
   Receive RIP packets
   Send RIP packets
   Passive interface: Disabled
   Split horizon: Enabled with Poisoned Reversed
   IP interface address:
    10.10.11.10/24
```

Use the following command to display the interface of rip:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     Dc - DHCP Client
     [*] - [AD/Metric]
     * - candidate default

R      2.2.2.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:02:28
C      10.10.11.0/24 is directly connected, eth-0-9
C      10.10.11.10/32 is in local loopback, eth-0-9
```

Configuring RIPv2 MD5 authentication (multiple keys)

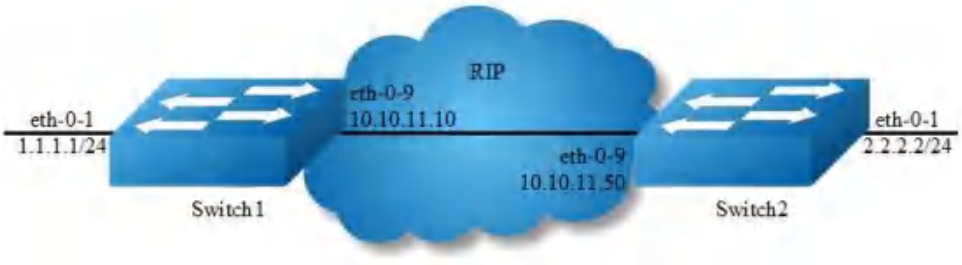


Figure 5-10 rip authentication

This example illustrates the md5 authentication of the routing information exchange process for RIP using multiple keys. Switch1 and B are running RIP and exchange routing updates. To configure authentication on Switch1, define a key chain, specify keys in the key chain and then define the authentication string or passwords to be used by the keys. Then set the time period during which it is valid to receive or send the authentication key by specifying the accept and send lifetimes.[optional].After defining the key string, specify the key chain (or the set of keys) that will be used for authentication on the interface and the authentication mode to be used. Configure Switch1 and B to have the same key ID and key string as Switch1 for the time that updates need to be exchanged.

In md5 authentication, both the key ID and key string are matched for authentication. R1 will receive only packets that match both the key ID and the key string in the specified key chain (within the accept lifetime) on that interface In the following example, Switch2 has the same key ID and key string as Switch1. For additional security, the accept lifetime and send lifetime are configured such that every fifth day the key ID and key string changes. To maintain continuity, the accept lifetimes should be configured to overlap; however, the send lifetime should not be overlapping.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit

Switch(config-if)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.10/24
Switch(config-if)# exit
```

Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 2.2.2.2/24
Switch(config-if)# exit

Switch(config-if)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.50/24
Switch(config-if)# exit
```

step 3 Enable RIP routing process and set the parameters

```
Switch(config)# router rip
Switch(config-router)# network 10.10.11.0/24
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

step 4 Create a key chain， and set the key string and lifetime

```
Switch(config)# key chain SUN
Switch(config-keychain)# key 1
Switch(config-keychain-key)# key-string key1
Switch(config-keychain-key)# accept-lifetime 12:00:00 Mar 2 2012 14:00:00 Mar 7 2012
Switch(config-keychain-key)# send-lifetime 12:00:00 Mar 2 2012 12:00:00 Mar 7 2012
Switch(config-keychain-key)# exit
```

Another key (optional):

```
Switch(config-keychain)# key 2
Switch(config-keychain-key)# key-string Earth
Switch(config-keychain-key)# accept-lifetime 12:00:00 Mar 7 2012 14:00:00 Mar 12 2012
Switch(config-keychain-key)# send-lifetime 12:00:00 Mar 7 2012 12:00:00 Mar 12 2012
Switch(config-keychain-key)# exit
```

Exit the keychain configure mode:

```
Switch(config-keychain)# exit
```

step 5 Specify the authentication string and mode

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip rip authentication key-chain SUN
Switch(config-if)# ip rip authentication mode md5
```

step 6 Exit the configure mode

```
Switch(config-if)# end
```

step 7 Validation

Use the following command to display the database of rip:

```
Switch# show ip rip database

Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
     C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP

  Network          Next Hop       Metric From          If      Time
R  2.2.2.0/24       10.10.11.50         2 10.10.11.50    eth-0-9  00:01:10
Rc 10.10.11.0/24                        1              eth-0-9
```

Use the following command to display the protocol state of rip process:

```
Switch# show ip protocols rip
Routing protocol is "rip"
 Sending updates every 30 seconds with +/-5 seconds, next due in 17 seconds
 Timeout after 180 seconds, Garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Default redistribution metric is 1
 Redistributing:
  connected  metric default
 Default version control: send version 2, receive version 2
  Interface       Send        Recv   Key-chain
  eth-0-9         2           2      SUN
 Routing for Networks:
  10.10.11.0/24
 Routing Information Sources:
  Gateway         Distance  Last Update  Bad Packets  Bad Routes
 Number of routes (including connected): 2
 Distance: (default is 120)
```

Use the following command to display the interface of rip:

```
Switch# show ip rip interface
eth-0-9 is up, line protocol is up
 Routing Protocol: RIP
  Receive RIP packets
  Send RIP packets
  Passive interface: Disabled
  Split horizon: Enabled with Poisoned Reversed
  IP interface address:
   10.10.11.10/24
```

Use the following command to display routes on the device:

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
```

```
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C      1.1.1.0/24 is directly connected, eth-0-1
C      1.1.1.1/32 is in local loopback, eth-0-1
R      2.2.2.0/24 [120/2] via 10.10.11.50, eth-0-9, 00:02:27
C      10.10.11.0/24 is directly connected, eth-0-9
C      10.10.11.10/32 is in local loopback, eth-0-9
```

Use the following command to display key chain:

```
Switch# show key chain
key chain SUN:
  key 1 -- text "key1"
    accept-lifetime <12:00:00 Mar 02 2012> - <14:00:00 Mar 07 2012>
    send-lifetime <12:00:00 Mar 02 2012> - < 12:00:00 Mar 07 2012>
  key 2 -- text "Earth"
    accept-lifetime <12:00:00 Mar 07 2012> - <14:00:00 Mar 12 2012>
    send-lifetime <12:00:00 Mar 07 2012> - < 12:00:00 Mar 12 2012>
Switch#
```

5.2.3 Application cases

N/A

5.3 Configuring OSPF

5.3.1 Overview

Function Introduction

OSPF is an Interior Gateway Protocol (IGP) designed expressly for IP networks, supporting IP subnet ting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

The implementation conforms to the OSPF Version 2 specifications with these key features:

- Definition of stub areas is supported: Routes learned through any IP routing protocol can be redistributed into another IP routing protocol. At the intradomain level, this means that OSPF can import routes learned through RIP. OSPF routes can also be exported into RIP.
- Plain text and MD5 authentication among neighboring routers within an area is supported: Configurable routing interface parameters include interface output cost, retransmission interval, interface transmit delay, router priority, router dead and hello intervals, and authentication key.

OSPF typically requires coordination among many internal routers, area border routers (ABRs) connected to multiple areas, and autonomous system boundary routers (ASBRs). The minimum configuration would use all default parameter values, no authentication, and interfaces assigned to areas. If you customize your environment, you must ensure coordinated configuration of all routers.

Principle Description

Reference to RFC 2328

5.3.2 Configuration

Basic OSPF Parameters Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the Routing process and associate the network with a specified OSPF area

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# quit
```

Note:use the following command to delete the routing process

```
Switch(config)# no router ospf 100
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ip protocols
Routing Protocol is "ospf 100"
 Redistributing:
 Routing for Networks:
  10.10.10.0/24
 Distance: (default is 110)
```

Enabling OSPF on an Interface



Figure 5-11 ospf

This example shows the minimum configuration required for enabling OSPF on an interface Switch1 and 2 are two routers in Area 0 connecting to network 10.10.10.0/24


Configure one interface so that it belongs to only one area. However, you can configure different interfaces on a router to belong to different areas.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# exit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.11/24
Switch(config-if)# exit
```

step 3 Configure the Routing process and associate the network with a specified OSPF area

Configure on Switch1:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
```

Configure on Switch2:

```
Switch(config)# router ospf 200
Switch(config-router)# network 10.10.10.0/24 area 0
```

Note: To using OSPF among two devices which are directly connected, the area IDs must be same. The ospf process IDs can be same or different.

step 4 Exit the configure mode

```
Switch(config-router)# end
```

step 5 Validation

Use the following command to display the database of ospf:

```
Switch# show ip ospf database

       OSPF Router with ID (10.10.10.10) (Process ID 100)

         Router Link States (Area 0)

Link ID       ADV Router     Age  Seq#      CkSum  Link count
10.10.10.10    10.10.10.10     26 0x80000006 0x1499 1
10.10.10.11    10.10.10.11     27 0x80000003 0x1895 1

         Net Link States (Area 0)

Link ID       ADV Router     Age  Seq#      CkSum
10.10.10.10    10.10.10.10     26 0x80000001 0xdfd8
```

Use the following command to display the interface of ospf:

```
Switch# show ip ospf interface
eth-0-9 is up, line protocol is up
 Internet Address 10.10.10.10/24, Area 0, MTU 1500
 Process ID 100, Router ID 10.10.10.10, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State DR, Priority 1, TE Metric 1
 Designated Router (ID) 10.10.10.10, Interface Address 10.10.10.10
 Backup Designated Router (ID) 10.10.10.11, Interface Address 10.10.10.11
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:06
 Neighbor Count is 1, Adjacent neighbor count is 1
 Crypt Sequence Number is 1527047183
 Hello received 25 sent 576, DD received 4 sent 4
 LS-Req received 1 sent 1, LS-Upd received 3 sent 3
 LS-Ack received 2 sent 2, Discarded 0
```

Use the following command to display the neighbor of ospf:

Switch1：

```
Switch# show ip ospf neighbor

OSPF process 100:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.10.11       1   Full/Backup       00:00:33   10.10.10.11     eth-0-9
```

Switch2：

```
Switch# show ip ospf neighbor

OSPF process 200:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.10.10       1   Full/DR         00:00:33    10.10.10.10     eth-0-9
```

Use the following command to display the ospf routes:

```
Switch# show  ip ospf route

OSPF process 100:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2

C  10.10.10.0/24 [1] is directly connected, eth-0-9, Area 0
```

Configuring Priority

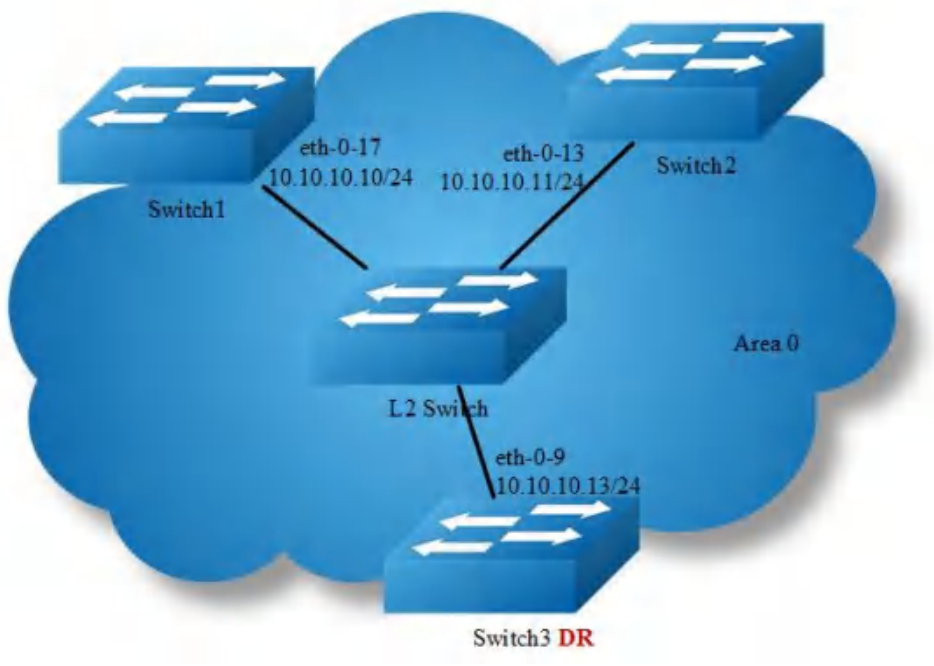


Figure 5-12 ospf priority

This example shows the configuration for setting the priority for an interface You can set a high priority for a router to make it the Designated Router (DR). Router Switch3 is configured to have a priority of 10, which is higher than the default priority (default priority is 1) of Switch1 and 2; making it the DR.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# quit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.11/24
Switch(config-if)# quit
```

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.13/24
Switch(config-if)# quit
```

Configure on L2 Switch:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# quit
```

step 3 Specify the router priority

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip ospf priority 10
Switch(config-if)# quit
```

step 4 Configure the Routing process and associate the network with a specified OSPF area

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-if)# quit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the neighbor of ospf:

Switch1：

```
Switch# show ip ospf neighbor

OSPF process 100:
Neighbor ID    Pri   State           Dead Time   Address       Interface
10.10.10.11     1   Full/Backup      00:00:31   10.10.10.11    eth-0-17
10.10.10.13    10   Full/DR          00:00:38   10.10.10.13    eth-0-17
```

Switch2：

```
Switch# show ip ospf neighbor

OSPF process 100:
Neighbor ID    Pri   State           Dead Time   Address       Interface
10.10.10.10     1   Full/DROther     00:00:39   10.10.10.10    eth-0-13
10.10.10.13    10   Full/DR          00:00:32   10.10.10.13    eth-0-13
```

Switch3：

```
Switch# show ip ospf neighbor

OSPF process 100:
Neighbor ID    Pri   State           Dead Time   Address       Interface
10.10.10.10     1   Full/DROther     00:00:37   10.10.10.10    eth-0-9
10.10.10.11     1   Full/Backup      00:00:32   10.10.10.11    eth-0-9
```

Use the following command to display the interface of ospf:

Switch1：

```
Switch# show ip ospf interface
eth-0-17 is up, line protocol is up
 Internet Address 10.10.10.10/24, Area 0, MTU 1500
 Process ID 100, Router ID 10.10.10.10, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State DROther, Priority 1, TE Metric 1
 Designated Router (ID) 10.10.10.13, Interface Address 10.10.10.13
 Backup Designated Router (ID) 10.10.10.11, Interface Address 10.10.10.11
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:10
 Neighbor Count is 2, Adjacent neighbor count is 2
 Crypt Sequence Number is 1527056133
 Hello received 106 sent 54, DD received 8 sent 9
 LS-Req received 2 sent 3, LS-Upd received 8 sent 5
 LS-Ack received 9 sent 5, Discarded 3
```

Switch2：

```
Switch# show ip ospf interface
eth-0-13 is up, line protocol is up
 Internet Address 10.10.10.11/24, Area 0, MTU 1500
 Process ID 100, Router ID 10.10.10.11, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State Backup, Priority 1, TE Metric 1
 Designated Router (ID) 10.10.10.13, Interface Address 10.10.10.13
 Backup Designated Router (ID) 10.10.10.11, Interface Address 10.10.10.11
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:10
 Neighbor Count is 2, Adjacent neighbor count is 2
 Crypt Sequence Number is 1527056130
 Hello received 110 sent 56, DD received 8 sent 7
```

```
LS-Req received 3 sent 2, LS-Upd received 12 sent 6
LS-Ack received 11 sent 8, Discarded 0
```

Switch3：

```
Switch# show ip ospf interface
eth-0-9 is up, line protocol is up
 Internet Address 10.10.10.13/24, Area 0, MTU 1500
 Process ID 100, Router ID 10.10.10.13, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State DR, Priority 10, TE Metric 1
 Designated Router (ID) 10.10.10.13, Interface Address 10.10.10.13
 Backup Designated Router (ID) 10.10.10.11, Interface Address 10.10.10.11
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:01
 Neighbor Count is 2, Adjacent neighbor count is 2
 Crypt Sequence Number is 1527056127
 Hello received 32 sent 16, DD received 9 sent 9
 LS-Req received 2 sent 2, LS-Upd received 11 sent 8
 LS-Ack received 10 sent 8, Discarded 0
```

1.3.6 Configuring OSPF Area Parameters

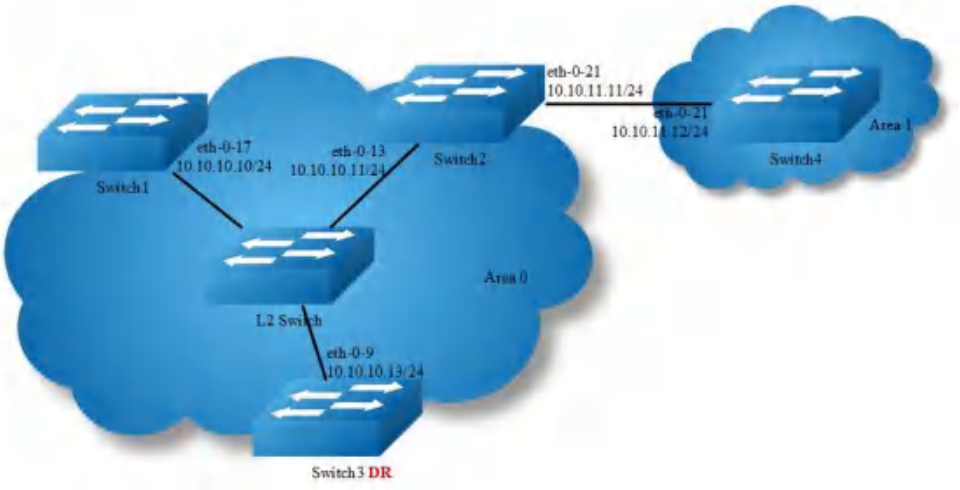


Figure 5-13 ospf area

You can optionally configure several OSPF area parameters. These parameters include authentication for password-based protection against unauthorized access to an area and stub areas. Stub areas are areas into which information on external routes is not sent. Instead, the area border router (ABR) generates a default external route into the stub area for destinations outside the autonomous system (AS).

Route summarization is the consolidation of advertised addresses into a single summary route to be advertised by other areas. If network numbers are contiguous, you can use the area range router configuration command to configure the ABR to advertise a summary route that covers all networks in the range.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
```

```
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# quit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.11/24
Switch(config-if)# quit

Switch(config)# interface eth-0-21
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.11/24
Switch(config-if)# quit
```

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.13/24
Switch(config-if)# quit
```

Configure on Switch4:

```
Switch(config)# interface eth-0-21
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.12/24
Switch(config-if)# quit
```

Configure on L2 Switch:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# quit
```

step 3 Set the ospf priority on the interface

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip ospf priority 10
Switch(config-if)# quit
```

step 4 Configure the Routing process and associate the network with a specified OSPF area

Configure on Switch1:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# quit
```

Configure on Switch2:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# network 10.10.11.0/24 area 1
Switch(config-router)# area 0 range 10.10.10.0/24
Switch(config-router)# area 1 stub no-summary
Switch(config-router)# quit
```

Configure on Switch3:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# quit
```

Configure on Switch4:

```
Switch(config)# router ospf 200
Switch(config-router)# network 10.10.11.0/24 area 1
Switch(config-router)# area 1 stub no-summary
Switch(config-router)# quit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the ospf routes:

Switch1：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C      10.10.10.0/24 is directly connected, eth-0-17
C      10.10.10.10/32 is in local loopback, eth-0-17
O IA   10.10.11.0/24 [110/2] via 10.10.10.11, eth-0-17, 00:00:04
```

Switch2：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C      10.10.10.0/24 is directly connected, eth-0-13
C      10.10.10.11/32 is in local loopback, eth-0-13
```

```
C       10.10.11.0/24 is directly connected, eth-0-21
C       10.10.11.11/32 is in local loopback, eth-0-21
```

Switch3：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       Dc - DHCP Client
       [*] - [AD/Metric]
       * - candidate default

C       10.10.10.0/24 is directly connected, eth-0-9
C       10.10.10.13/32 is in local loopback, eth-0-9
O IA    10.10.11.0/24 [110/2] via 10.10.10.11, eth-0-9, 00:06:29
```

Switch4：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       Dc - DHCP Client
       [*] - [AD/Metric]
       * - candidate default
Gateway of last resort is 10.10.11.11 to network 0.0.0.0
O*IA    0.0.0.0/0 [110/2] via 10.10.11.11, eth-0-21, 00:12:46
C       10.10.10.0/24 is directly connected, eth-0-21
C       10.10.10.12/32 is in local loopback, eth-0-21
```

Redistributing Routes into OSPF

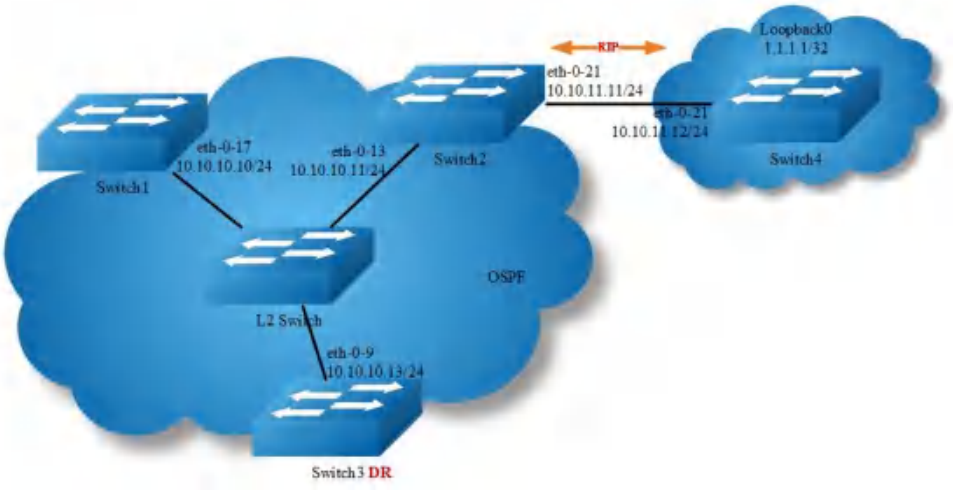


Figure 5-14 ospf redistribute

In this example the configuration causes RIP routes to be imported into the OSPF routing table and advertised as Type 5 External LSAs into Area 0.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address

Configure on Switch1:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# quit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.11/24
Switch(config-if)# quit

Switch(config)# interface eth-0-21
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.11/24
Switch(config-if)# quit
```

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.13/24
Switch(config-if)# quit
```

Configure on Switch4:

```
Switch(config)# interface eth-0-21
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.11.12/24
Switch(config-if)# quit

Switch(config)# interface loopback 0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# quit
```

Configure on L2 Switch:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# quit
Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# quit
```

step 3 Set the ospf priority on the interface

Configure on Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# ip ospf priority 10
Switch(config-if)# quit
```

step 4 Configure the Routing process and associate the network with a specified OSPF area

Configure on Switch1:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# quit
```

Configure on Switch2:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# redistribute connected
Switch(config-router)# redistribute rip
Switch(config-router)# quit
```

Configure on Switch3:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# quit
```

step 5 Enable RIP routing process and associate networks

Configure on Switch2:

```
Switch(config)# router rip
Switch(config-router)# network 10.10.11.0/24
Switch(config-router)#redistribute connected
Switch(config-router)# quit
```

Configure on Switch4:

```
Switch(config)# router rip
Switch(config-router)# network 10.10.11.0/24
Switch(config-router)# network 1.1.1.1/32
Switch(config-router)# redistribute connected
Switch(config-router)# quit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the ospf routes:

Switch1：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
```

```
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

O E2    1.1.1.1/32 [110/20] via 10.10.10.11, eth-0-17, 00:01:54
C       10.10.10.0/24 is directly connected, eth-0-17
C       10.10.10.10/32 is in local loopback, eth-0-17
O E2    10.10.11.0/24 [110/20] via 10.10.10.11, eth-0-17, 00:03:49
```

Switch2：

```
Switch# show  ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

R       1.1.1.1/32 [120/2] via 10.10.11.12, eth-0-21, 00:02:27
C       10.10.10.0/24 is directly connected, eth-0-13
C       10.10.10.11/32 is in local loopback, eth-0-13
C       10.10.11.0/24 is directly connected, eth-0-21
C       10.10.11.11/32 is in local loopback, eth-0-21
```

Switch3：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

O E2    1.1.1.1/32 [110/20] via 10.10.10.11, eth-0-9, 00:03:01
C       10.10.10.0/24 is directly connected, eth-0-9
C       10.10.10.13/32 is in local loopback, eth-0-9
O E2    10.10.11.0/24 [110/20] via 10.10.10.11, eth-0-9, 00:04:57
```

Switch4：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C       1.1.1.1/32 is directly connected, loopback0
R       10.10.10.0/24 [120/2] via 10.10.11.11, eth-0-21, 00:17:36
```

```
C    10.10.11.0/24 is directly connected, eth-0-21
C    10.10.11.12/32 is in local loopback, eth-0-21
```

Use the following command to display the database of ospf:

Switch1：

```
Switch# show ip ospf database external

       OSPF Router with ID (10.10.10.10) (Process ID 100)

         AS External Link States

LS age: 317
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 1.1.1.1 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x4a47
Length: 36
Network Mask: /32
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 20
    Forward Address: 0.0.0.0
    External Route Tag: 0

LS age: 438
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 10.10.11.0 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x0472
Length: 36
Network Mask: /24
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 20
    Forward Address: 0.0.0.0
    External Route Tag: 0
```

Switch2：

```
Switch# show ip ospf database external

       OSPF Router with ID (10.10.10.11) (Process ID 100)

         AS External Link States

LS age: 367
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 1.1.1.1 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x4a47
```

```
Length: 36
Network Mask: /32
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 20
    Forward Address: 0.0.0.0
    External Route Tag: 0

LS age: 487
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 10.10.11.0 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x0472
Length: 36
Network Mask: /24
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 20
    Forward Address: 0.0.0.0
    External Route Tag: 0
```

Switch3：

```
Switch# show ip ospf database external

       OSPF Router with ID (10.10.10.13) (Process ID 100)

         AS External Link States

LS age: 396
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 1.1.1.1 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x4a47
Length: 36
Network Mask: /32
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 20
    Forward Address: 0.0.0.0
    External Route Tag: 0

LS age: 517
Options: 0x2 (*|-|-|-|-|-|E|-)
LS Type: AS-external-LSA
Link State ID: 10.10.11.0 (External Network Number)
Advertising Router: 10.10.10.11
LS Seq Number: 80000001
Checksum: 0x0472
Length: 36
Network Mask: /24
    Metric Type: 2 (Larger than any link state path)
```

```
TOS: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

OSPF Cost

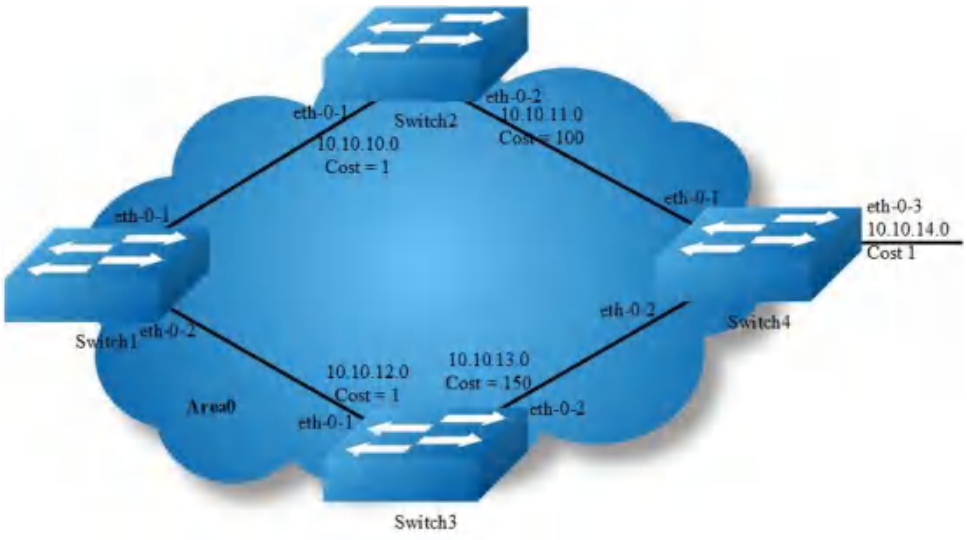


Figure 5-15 ospf cost

You can make a route the preferred route by changing its cost. In this example, cost has been configured to make Switch2 the next hop for Switch1.

The default cost on each interface is 1(1000M speed). Interface eth2 on Switch2 has a cost of 100 and interface eth2 on Switch3 has a cost of 150. The total cost to reach(Switch4 network 10.10.14.0) through Switch2 and Switch3:

Switch2: 1+1+100 = 102

Switch3: 1+1+150 = 152

Therefore, Switch1 chooses Switch2 as its next hop for destination Switch4

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address. Set the ospf cost under the interface configure mode

Configure on Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.1/24
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.12.1/24
Switch(config-if)# exit
```

Configure on Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.2/24
```

```
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.11.2/24
Switch(config-if)# ip ospf cost 100
Switch(config-if)# exit
```

Configure on Switch3:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.12.2/24
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.13.2/24
Switch(config-if)# ip ospf cost 150
Switch(config-if)# exit
```

Configure on Switch4:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.11.1/24
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.13.1/24
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.14.1/24
Switch(config-if)# exit
```

step 3 Configure the Routing process and associate the network with a specified OSPF area

Configure on Switch1:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# network 10.10.12.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch2:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# network 10.10.11.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch3:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.12.0/24 area 0
Switch(config-router)# network 10.10.13.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch4:

```
Switch(config)# router ospf 100
Switch(config-router)# network 10.10.11.0/24 area 0
```

```
Switch(config-router)# network 10.10.13.0/24 area 0
Switch(config-router)# network 10.10.14.0/24 area 0
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the ospf routes:

Switch1：

```
Switch# show ip ospf route
OSPF process 0:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
C  10.10.10.0/24 [1] is directly connected, eth-0-1, Area 0
O  10.10.11.0/24 [101] via 10.10.10.2, eth-0-1, Area 0
C  10.10.12.0/24 [1] is directly connected, eth-0-2, Area 0
O  10.10.13.0/24 [102] via 10.10.10.2, eth-0-1, Area 0
O  10.10.14.0/24 [102] via 10.10.10.2, eth-0-1, Area 0
```

Switch2：

```
Switch# show ip ospf route
OSPF process 100:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
C  10.10.10.0/24 [10] is directly connected, eth-0-1, Area 0
C  10.10.11.0/24 [100] is directly connected, eth-0-2, Area 0
O  10.10.12.0/24 [11] via 10.10.10.1, eth-0-1, Area 0
O  10.10.13.0/24 [101] via 10.10.11.1, eth-0-2, Area 0
O  10.10.14.0/24 [101] via 10.10.11.1, eth-0-2, Area 0
```

Switch3：

```
Switch# show ip ospf route
OSPF process 100:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
O  10.10.10.0/24 [1] via 10.10.12.1, eth-0-1, Area 0
O  10.10.11.0/24 [101] via 10.10.12.1, eth-0-1, Area 0
C  10.10.12.0/24 [1] is directly connected, eth-0-1, Area 0
O  10.10.13.0/24 [102] via 10.10.12.1, eth-0-1, Area 0
O  10.10.14.0/24 [102] via 10.10.12.1, eth-0-1, Area 0
```

Switch4：

```
Switch# show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    [*] - [AD/Metric]
```

```
    * - candidate default
O   10.10.10.0/24 [110/1] via 10.10.11.2, eth-0-1, 00:06:27
C   10.10.11.0/24 is directly connected, eth-0-1
O   10.10.12.0/24 [110/1] via 10.10.13.2, eth-0-2, 00:06:17
C   10.10.13.0/24 is directly connected, eth-0-2
C   10.10.14.0/24 is directly connected, eth-0-3
```

Configuring OSPF authentications

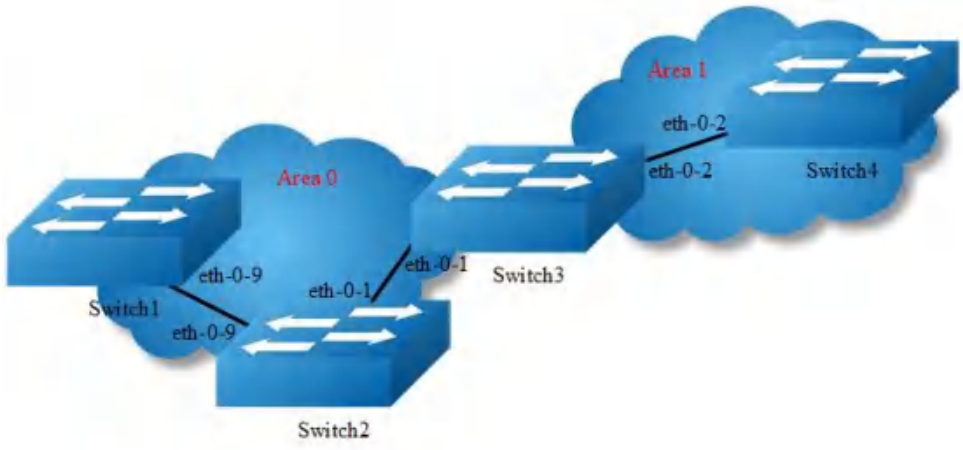


Figure 5-16 ospf authentication

In our implementation there are three types of OSPF authentications–Null authentication (Type 0), Simple Text (Type 1) authentication and MD5 (Type 2) authentication. With null authentication, routing exchanges over the network are not authenticated. In Simple Text authentication, the authentication type is the same for all routers that communicate using OSPF in a network. For MD5 authentication, you configure a key and a key-id on each router. The router generates a message digest on the basis of the key, key ID and the OSPF packet and adds it to the OSPF packet.

The Authentication type can be configured on a per-interface basis or a per-area basis. Additionally, Interface and Area authentication can be used together. Area authentication is used for an area and interface authentication is used for a specific interface in the area. If the Interface authentication type is different from Area authentication type, Interface authentication type overrides the Area authentication type. If the Authentication type is not specified for an interface, the Authentication type for the area is used. The authentication command descriptions contain details of each type of authentication. Refer to the OSPF Command Reference for OSPF authentication commands.

In the example below, Switch1 and B are configured for both the interface and area authentications. The authentication type of interface eth-0-9 on Switch1 and interface eth-0-9 on Switch2 is null authentication mode The authentication type of interface eth-0-1 on Switch2 and interface eth-0-1 on Switch3 is simple authentication mode The authentication type of interface eth-0-2 on Switch3 and interface eth-0-2 on Switch4 is MD5 authentication mode in area1,if you define area 1 authentication type first, you needn't define interface authentication type, only define authentication key value.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address. Set the ospf authentication under the interface configure mode

Configure on Switch1:

```
Switch(config)#interface eth-0-9
Switch(config-if)#no switchport
Switch(config-if)#ip address 9.9.9.1/24
Switch(config-if)#ip ospf authentication
Switch(config-if)#ip ospf authentication null
Switch(config-if)# exit
```

Configure on Switch2:

```
Switch(config)#interface eth-0-1
Switch(config-if)#no switchport
Switch(config-if)#ip address 1.1.1.1/24
Switch(config-if)#ip ospf authentication
Switch(config-if)#ip ospf authentication-key test
Switch(config-if)# exit

Switch(config)#interface eth-0-9
Switch(config-if)#no switchport
Switch(config-if)#ip address 9.9.9.2/24
Switch(config-if)#ip ospf authentication
Switch(config-if)#ip ospf authentication null
Switch(config-if)# exit
```

Configure on Switch3:

```
Switch(config)#interface eth-0-2
Switch(config-if)#no switchport
Switch(config-if)#ip address 2.2.2.1/24
Switch(config-if)# ip ospf message-digest-key 2 md5 ospf
Switch(config-if)# exit
Switch(config)#interface eth-0-1
Switch(config-if)#no switchport
Switch(config-if)#ip address 1.1.1.2/24
Switch(config-if)#ip ospf authentication
Switch(config-if)# ip ospf authentication-key test
Switch(config-if)# exit
```

Configure on Switch4:

```
Switch(config)#interface eth-0-2
Switch(config-if)#no switchport
Switch(config-if)#ip address 2.2.2.2/24
Switch(config-if)# ip ospf message-digest-key 2 md5 ospf
Switch(config-if)# exit
```

step 3 Configure the Routing process and associate the network with a specified OSPF area

Configure on Switch1:

```
Switch(config)# router ospf
Switch(config-router)# network 9.9.9.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch2:

```
Switch(config)# router ospf
Switch(config-router)# network 9.9.9.0/24 area 0
Switch(config-router)# network 1.1.1.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch3:

```
Switch(config)# router ospf
Switch(config-router)# area 1 authentication message-digest
Switch(config-router)# network 2.2.2.0/24 area 1
Switch(config-router)# network 1.1.1.0/24 area 0
Switch(config-router)# exit
```

Configure on Switch4:

```
Switch(config)# router ospf
Switch(config-router)# area 1 authentication message-digest
Switch(config-router)# network 2.2.2.0/24 area 1
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the neighbor of ospf:

Switch1：

```
Switch# show ip ospf neighbor
OSPF process 0:
Neighbor ID    Pri  State        Dead Time  Address      Interface
9.9.9.2          1  Full/DR      00:00:38   9.9.9.2      eth-0-9
```

Switch2：

```
Switch# show ip ospf neighbor
OSPF process 0:
Neighbor ID    Pri  State        Dead Time  Address      Interface
2.2.2.1          1  Full/Backup  00:00:35   1.1.1.2      eth-0-1
1.1.1.1          1  Full/Backup  00:00:38   9.9.9.1      eth-0-9
```

Switch3：

```
Switch# show ip ospf neighbor
OSPF process 0:
Neighbor ID    Pri  State        Dead Time  Address      Interface
9.9.9.2          1  Full/DR      00:00:35   1.1.1.1      eth-0-1
2.2.2.2          1  Full/DR      00:00:38   2.2.2.2      eth-0-2
```

Switch4：

```
Switch# show ip ospf neighbor
OSPF process 0:
Neighbor ID    Pri  State        Dead Time  Address      Interface
2.2.2.1          1  Full/Backup  00:00:35   2.2.2.1      eth-0-2
```

Use the following command to display the interface of ospf:

Switch3：

```
Switch# show ip ospf interface
eth-0-1 is up, line protocol is up
 Internet Address 1.1.1.2/24, Area 0, MTU 1500
 Process ID 0, Router ID 2.2.2.1, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State Backup, Priority 1, TE Metric 1
 Designated Router (ID) 9.9.9.2, Interface Address 1.1.1.1
 Backup Designated Router (ID) 2.2.2.1, Interface Address 1.1.1.2
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:01
 Neighbor Count is 1, Adjacent neighbor count is 1
 Crypt Sequence Number is 1301244696
 Hello received 385 sent 384, DD received 3 sent 5
 LS-Req received 1 sent 1, LS-Upd received 11 sent 14
```

```
 LS-Ack received 12 sent 10, Discarded 1
 Simple password authentication enabled
```

Use the following command to display the protocol state of ospf process:

Switch3：

```
Switch# show ip ospf
 Routing Process "ospf 0" with ID 2.2.2.1
 Process uptime is 1 hour 7 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583 Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 This router is an ABR, ABR Type is Alternative Cisco (RFC3509)
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum 0x000000
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 0
 External LSA database is unlimited.
 Number of LSA originated 17
 Number of LSA received 57
 Number of areas attached to this router: 2
   Area 0 (BACKBONE)
      Number of interfaces in this area is 1(1)
      Number of fully adjacent neighbors in this area is 1
      Area has no authentication
      SPF algorithm last executed 01:06:56.340 ago
      SPF algorithm executed 16 times
      Number of LSA 6. Checksum 0x034b09
   Area 1
      Number of interfaces in this area is 1(1)
      Number of fully adjacent neighbors in this area is 1
      Number of fully adjacent virtual neighbors through this area is 0
      Area has message digest authentication
      SPF algorithm last executed 00:03:29.430 ago
      SPF algorithm executed 17 times
      Number of LSA 5. Checksum 0x0230e3
```

5.3.3 Application cases

N/A

Configuring OSPF authentications password encryption

When we configure the OSPF authentication, the authentication-key is simple words.

Thus, the authentication-key is shown as simple words in system. In order to increase

the safety of our system, the OSPF authentication-key is shown as encryption words.

Additionally, the system now supports configuring OSPF authentication with encryption words.

Simple Password

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address. Set the ospf authentication under the interface configure mode and simple password

```
Switch(config)#interface eth-0-9
Switch(config-if)#no switchport
Switch(config-if)#ip address 9.9.9.1/24
Switch(config-if)#ip ospf authentication
Switch(config-if)#ip ospf authentication-key test
Switch(config-if)# exit
```

step 3 Enter the configure mode, translate to encryption password and show it

```
Switch(config)# service password-encryption
Switch(config)# show running-config
 !
 service password-encryption
 !
 interface eth-0-9
 no switchport
 ip address 9.9.9.1/24
 ip ospf authentication-key 8 af0443346357baf8
 !
```

step 4 Disable the function of showing encryption password, delete the old authentication-key and set new one, then show the password

```
Switch(config)#no service password-encryption
Switch(config)#interface eth-0-9
Switch(config-if)#no ip ospf authentication-key
Switch(config-if)#ip ospf authentication-key test123
Switch(config-if)# exit
Switch(config)# show running-config
 !
 no service password-encryption
 !
 interface eth-0-9
 no switchport
 ip address 9.9.9.1/24
 ip ospf authentication-key test123
 !
```

step 5 Configuring OSPF encryption password

```
Switch(config)#interface eth-0-9
Switch(config-if)#no ip ospf authentication-key
Switch(config-if)#ip ospf authentication-key 8 af0443346357baf8
Switch(config-if)# exit
Switch(config)# show running-config
 !
 no service password-encryption
 !
 interface eth-0-9
 no switchport
 ip address 9.9.9.1/24
 ip ospf authentication-key test123
 !
```

MD5 Password

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode, set the attributes and ip address. Set the ospf authentication under the interface configure mode and simple password

```
Switch(config)#interface eth-0-9
Switch(config-if)#no switchport
Switch(config-if)#ip address 9.9.9.1/24
Switch(config-if)#ip ospf authentication message-digest
Switch(config-if)#ip ospf message-digest-key 1 md5 ospf
Switch(config-if)# exit
```

step 3 Enter the configure mode, translate to encryption password and show it

```
Switch(config)# service password-encryption
Switch(config)# show running-config
 !
 service password-encryption
 !
 interface eth-0-9
 no switchport
 ip address 9.9.9.1/24
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 8 1f0276567f2db31f
 !
```

step 4 Disable the function of showing encryption password, delete the old authentication-key and set new one, then show the password

```
Switch(config)#no service password-encryption
Switch(config)#interface eth-0-9
Switch(config-if)#no ip ospf message-digest-key 1
Switch(config-if)#ip ospf message-digest-key 1 md5 ospf123
Switch(config-if)# exit
Switch(config)# show running-config
 !
 no service password-encryption
 !
 interface eth-0-9
 no switchport
 ip address 9.9.9.1/24
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 ospf123
 !
```

step 5 Configuring OSPF encryption password

```
Switch(config)#interface eth-0-9
Switch(config-if)#no ip ospf message-digest-key 1
Switch(config-if)#ip ospf message-digest-key 1 md5 8 1f0276567f2db31f
Switch(config-if)# exit
Switch(config)# show running-config
 !
 no service password-encryption
```

```
!
interface eth-0-9
no switchport
ip address 9.9.9.1/24
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 8 1f0276567f2db31f
!
```

5.4 Configuring Prefix-list

5.4.1 Overview

Function Introduction

Routing Policy is the technology for modifying route information to change traffic route. Prefix list is a kind of route policies that used to control and modify routing information. A prefix list is identified by list name and contains one or more ordered entries which are processed sequentially. Each entry provides a matched range for network prefix and has a unique sequence number in the list. In the matching process，switch will check entries orderly. If a entry matches conditions, this process would finish.

Principle Description

N/A

5.4.2 Configuration

Basic Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a prefix-list

Note: Create a prefix-list. If the sequence of the rule is not specified, system should automatically assign an sequence number for it. Support different actions such as permit and deny. Support to add description string for a prefix-list.

```
Switch(config)# ip prefix-list test seq 1 deny 35.0.0.0/8 le 16
Switch(config)# ip prefix-list test permit any
Switch(config)# ip prefix-list test description this prefix list is fot test
Switch(config)# ip prefix-list test permit 36.0.0.0/24
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display the prefix-list:

```
Switch# show ip prefix-list detail
Prefix-list list number: 1
Prefix-list entry number: 3
Prefix-list with the last deletion/insertion: test
ip prefix-list test:
  Description: this prefix list is fot test
  count: 3, range entries: 0, sequences: 1 - 10
    seq 1 deny 35.0.0.0/8 le 16 (hit count: 0, refcount: 0)
    seq 5 permit any (hit count: 0, refcount: 0)
    seq 10 permit 36.0.0.0/24 (hit count: 0, refcount: 0)
```

Used by rip

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a prefix-list

```
Switch(config)# ip prefix-list aa seq 11 deny 35.0.0.0/8 le 16
Switch(config)# ip prefix-list aa permit any
```

step 3 Apply the prefix-list under the router rip configure mode

```
Switch(config)# router rip
Switch(config-router)# distribute-list prefix aa out
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to display the prefix-list:

```
Switch# show ip prefix-list
ip prefix-list aa: 2 entries
   seq 11 deny 35.0.0.0/8 le 16
   seq 15 permit any
```

Use the following command to display the configuration of the device:

```
Switch# show running-config
Building configuration...
…
ip prefix-list aa seq 11 deny 35.0.0.0/8 le 16
ip prefix-list aa seq 15 permit any
…
router rip
 distribute-list prefix aa out
```

Used by Route-map

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a prefix-list

```
Switch(config)# ip prefix-list aa seq 11 deny 3.3.3.0/8 le 24
Switch(config)# ip prefix-list aa permit any
```

step 3 create a route map to match the prefix-list

```
Switch(config)# route-map abc permit
Switch(config-route-map)# match ip address prefix-list aa
Switch(config-route-map)# set local-preference 200
Switch(config-route-map)# exit

Switch(config)# route-map abc permit 20
Switch(config-route-map)# exit
```

step 4 Apply the route under the router bgp configure mode

```
Switch(config)# router bgp 1
Switch(config-router)# neighbor 1.1.1.2 remote-as 1
Switch(config-router)# neighbor 1.1.1.2 route-map abc out
Switch(config-router)# network 2.2.2.2/32
Switch(config-router)# network 3.3.3.3/32
```

step 5 Exit the configure mode

```
Switch(config-router)# end
```

step 6 Validation

Use the following command to display the route map:

```
Switch # show route-map
route-map abc, permit, sequence 10
 Match clauses:
   ip address prefix-list aa
 Set clauses:
   local-preference 200
route-map abc, permit, sequence 20
 Match clauses:
Set clauses:
```

Use the following command to display the configuration of the device:

```
Switch # show running-config
Building configuration...
…
ip prefix-list aa seq 11 deny 3.3.3.0/8 le 24
ip prefix-list aa seq 15 permit any
!
!
route-map abc permit 10
 match ip address prefix-list aa
 set local-preference 200
!
route-map abc permit 20
…
router bgp 1
 neighbor 1.1.1.2 remote-as 1
 !
 address-family ipv4
 no synchronization
 network 2.2.2.2 mask 255.255.255.255
 network 3.3.3.3 mask 255.255.255.255
 neighbor 1.1.1.2 activate
 neighbor 1.1.1.2 route-map abc out
 exit-address-family
 !
 address-family vpnv4 unicast
 no synchronization
 exit-address-family
```

5.4.3 Application cases

N/A

5.5 Configuring Route-map

5.5.1 Overview

Function Introduction

Route-map is used to control and modify routing information. The route-map command allows redistribution of routes. It has a list of match and set commands associated with it. The match commands specify the conditions under which redistribution is allowed, and the set commands specify the particular redistribution actions to be performed if the criteria enforced by match commands are met. Route maps are used for detailed control over route distribution between routing processes. Route maps also allow policy routing, and might route packets to a different route than the obvious shortest path.

If the permit parameter is specified, and the match criteria are met, the route is redistributed as specified by set actions. If the match criteria are not met, the next route map with the same tag is tested. If the deny parameter is specified, and the match criteria are met, the route is not redistributed, and any other route maps with the same map tag are not examined. Routes are checked from line to line looking for a match. If there is no match and the bottom of the route map is reached, then the router denies the route from being redistributed. There is always an implicit deny at the end of a route map.

Specify the sequence parameter to indicate the position a new route map is to have in the list of route maps already configured with the same name.

Principle Description

N/A

5.5.2 Configuration

Configuring Route-map for OSPF

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create route map and set the rule and action



The name of route-map is up to 20 characters, in this example the name is “abc”. Two actions “permit” and “deny” are supported; the default action is “permit”. The valid range for sequence number is 1-65535. If the sequence number is not specified when creating first rule of the route-map, system assigns number 10 by default.

```
Switch(config)# route-map abc permit
Switch(config-route-map)# match metric 20
Switch(config-route-map)# set tag 2
Switch(config-route-map)# exit

Switch(config)# route-map abc permit 20
Switch(config-route-map)# exit
```

step 3 Enter the router ospf configure mode, redistribute rip routes and apply the route map

```
Switch(config)# router ospf 100
Switch(config-router)# redistribute rip route-map abc
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show route-map
route-map abc, permit, sequence 10
```

```
 Match clauses:
  metric 20
 Set clauses:
  tag 2
route-map abc, permit, sequence 20
 Match clauses:
 Set clauses:
```

Configuring Route-map for BGP

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create ip access list

```
Switch(config)# ip access-list acl1
Switch(config-ip-acl)# permit any 3.3.3.0 0.0.0.255 any
Switch(config-ip-acl)# exit
```

step 3 Create route map to match the access list and set the rule and action

```
Switch(config)# route-map abc permit
Switch(config-route-map)# match ip address acl1
Switch(config-route-map)# set local-preference 200
Switch(config-route-map)# exit

Switch(config)# route-map abc permit 20
Switch(config-route-map)# exit
```

step 4 Enter the router bgp configure mode, and apply the route map

```
Switch(config)# router bgp 1
Switch(config-router)# neighbor 1.1.1.2 remote-as 1
Switch(config-router)# neighbor 1.1.1.2 route-map abc out
Switch(config-router)# network 2.2.2.2/32
Switch(config-router)# network 3.3.3.3/32
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
DUT1# show route-map
route-map abc, permit, sequence 10
 Match clauses:
  ip address acl1
 Set clauses:
  local-preference 200
route-map abc, permit, sequence 20
 Match clauses:
 Set clauses:
DUT2# show ip bgp
BGP table version is 6, local router ID is 1.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i2.2.2.2/32	1.1.1.1	0	100	0	i
*>i3.3.3.3/32	1.1.1.1	0	200	0	i

5.5.3 Application cases

N/A

5.6 Configuring Policy-Based Routing

5.6.1 Overview

Function Introduction

Policy-Based Routing(PBR) provide freedom to implement packet forwarding and routing, according to the defined policies in a way that goes beyond traditional routing protocol concerns. By using policy-based routing, customers can implement policies that selectively cause packets to take different paths.

Principle Description

N/A

5.6.2 Configuration

PBR Configuration

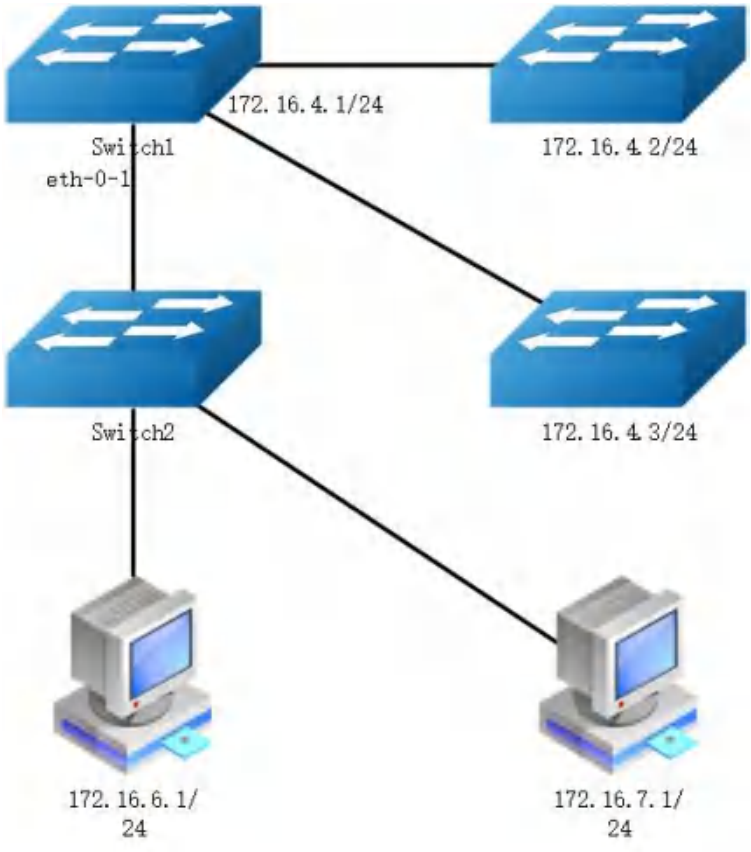


Figure 5-17 pbr

The figure above is a typical topology: After Enabling PBR on interface eth-0-1 of Switch1, packets from 172.16.6.1 should be forwarded to 172.16.4.2, and other packets should be forwarded according to the original routes.

Configure on Switch1:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create an ip access list to match source ip address

```
Switch(config)# ip access-list acl1
Switch(config-ip-acl)# 10 permit any 172.16.6.0 0.0.0.255 any
Switch(config-ip-acl)# exit
```

step 3 Create a route map, to match the ip access list and set the nexthop ip

```
Switch(config)# route-map rmap permit 10
Switch(config-route-map)# match ip address acl1
Switch(config-route-map)# set ip next-hop 172.16.4.2
Switch(config-route-map)# exit
```

step 4 Enter the interface configure mode, set the attributes and ip address, and apply the route map

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 172.16.5.2/24
Switch(config-if)# no shutdown
Switch(config-if)# ip policy route-map rmap
Switch(config-if)# exit
```

step 5 Create a static route with the nexthop ip 172.16.4.3 (optional)

To forwarding the packets which not hit the PBR, we can use a static route. Dynamic protocols such as RIP/OSPF are can also meet this requirement.

```
Switch(config)# ip route 0.0.0.0/0 172.16.4.3
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show ip policy route-map
Route-map              interface
rmap                   eth-0-1
```

Configure PBR and BFD linkage

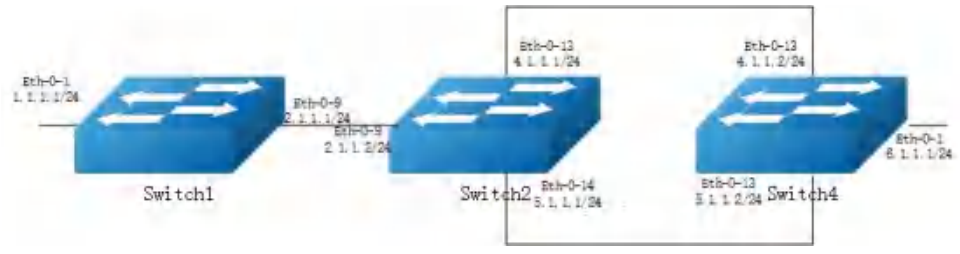


Figure 5-18 pbr

The figure above is a typical topology: Switch2 will forward packet to eth-0-13 according PBR routes, when Switch4 eth-0-13 shutdown, bfd session statues will be down, then track 1 will be down, and the PBR next-hop 4.1.1.2 will be invalid, packet will forward to eth-0-14.

step 1 Configure on Switch1:

```
Switch1# configure terminal
Switch1(config)# interface eth-0-1
Switch1(config-if)# no shutdown
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 1.1.1.1/24
Switch1(config-if)# interface eth-0-9
Switch1(config-if)# no shutdown
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 2.1.1.1/24
Switch1(config-if)# quit
Switch1(config)# ip route 5.1.1.0/24 2.1.1.2
Switch1(config)# ip route 6.1.1.0/24 2.1.1.2
```

step 2 Configure on Switch2:

```
Switch2# configure terminal
Switch2(config)# ip access-list acl1
Switch2(config-ip-acl)# 10 permit any host 2.1.1.1 any
Switch2(config-ip-acl)# quit
Switch2(config)# route-map rmap permit 10
Switch2(config-route-map)# match ip address acl1
Switch2(config-route-map)# set ip next-hop 4.1.1.2 track 1
Switch2(config-route-map)# quit
Switch2(config)# interface eth-0-9
Switch2(config-if)# no shutdown
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 2.1.1.2/24
Switch2(config-if)# ip policy route-map rmap
Switch2(config-if)# interface eth-0-13
Switch2(config-if)# no shutdown
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 4.1.1.1/24
Switch2(config-if)# interface eth-0-14
Switch2(config-if)# no shutdown
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 5.1.1.1/24
Switch2(config-if)# quit
Switch2(config)# track 1 bfd source interface eth-0-13 destination 4.1.1.2
Switch2(config-track)# quit
Switch2(config)# ip route 1.1.1.0/24 2.1.1.1
Switch2(config)# ip route 6.1.1.0/24 5.1.1.2
```

step 3 Configure on Switch4:

```
Switch4# configure terminal
Switch4(config)# interface eth-0-1
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 6.1.1.1/24
Switch4(config-if)# interface eth-0-13
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 4.1.1.2/24
Switch4(config-if)# interface eth-0-14
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 5.1.1.2/24
```

```
Switch4(config-if)# quit
Switch4(config)# track 1 bfd source interface eth-0-13 destination 4.1.1.1
Switch4(config-track)# quit
Switch4(config)# ip route 1.1.1.0/24 5.1.1.1
Switch4(config)# ip route 2.1.1.0/24 5.1.1.1
```

step 3 ping 6.1.1.1 Switch2 will forward packet to eth-0-13

```
Switch1# ping 6.1.1.1
PING 6.1.1.1 (6.1.1.1) 56(84) bytes of data.
64 bytes from 6.1.1.1: icmp_seq=1 ttl=63 time=417 ms
64 bytes from 6.1.1.1: icmp_seq=2 ttl=63 time=428 ms
64 bytes from 6.1.1.1: icmp_seq=3 ttl=63 time=441 ms
64 bytes from 6.1.1.1: icmp_seq=4 ttl=63 time=469 ms
64 bytes from 6.1.1.1: icmp_seq=5 ttl=63 time=461 ms

--- 6.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 6810ms
rtt min/avg/max/mdev = 417.834/443.810/469.720/19.470 ms
```

step 4 shutdown eth-0-13 of Switch4

```
Switch4# configure terminal
Switch4(config)# interface eth-0-13
Switch4(config-if)# shutdown
```

step 5 Validation

```
Switch2# show track
Track 1
  Type            : BFD state
  Source interface   : eth-0-13
  Destination IP     : 4.1.1.2
  BFD Local discr    : 8192
  rmap            : pref 10 track 1
  State           : down

Switch2# show bfd session
Abbreviation:
LD: Local Discriminator.  RD: Remote Discriminator
S: Single hop session.   M: Multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
SBFD: Seamless BFD
A: Admin down.   D:Down.   I:Init.   U:Up.
================================================================================
LD       RD       TYPE ST   UP-Time   Remote-Addr    Sbfd-Type VRF

8192     0        S-DD D   00:00:00  4.1.1.2       None     default

Number of Sessions:   1
Switch2 will forward packet to eth-0-14
Switch# ping 6.1.1.1
PING 6.1.1.1 (6.1.1.1) 56(84) bytes of data.
64 bytes from 6.1.1.1: icmp_seq=1 ttl=63 time=414 ms
64 bytes from 6.1.1.1: icmp_seq=2 ttl=63 time=432 ms
64 bytes from 6.1.1.1: icmp_seq=3 ttl=63 time=424 ms
64 bytes from 6.1.1.1: icmp_seq=4 ttl=63 time=525 ms
```

```
64 bytes from 6.1.1.1: icmp_seq=5 ttl=63 time=437 ms

--- 6.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 6563ms
rtt min/avg/max/mdev = 414.720/446.816/525.276/39.949 ms
```

5.6.3 Application cases

N/A

5.6.4 Configuring BGP

5.6.5 Overview

Function Introduction

The Border Gateway Protocol (BGP) is an inter-Autonomous System routing protocol.

The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASes) that reachability information traverses. This information is sufficient for constructing a graph of AS connectivity for this reachability, from which routing loops may be pruned and, at the AS level, some policy decisions may be enforced.

BGP-4 provides a set of mechanisms for supporting Classless Inter-Domain Routing (CIDR) [RFC1518, RFC1519]. These mechanisms include support for advertising a set of destinations as an IP prefix and eliminating the concept of network "class" within BGP. BGP-4 also introduces mechanisms that allow aggregation of routes, including aggregation of AS paths.

Routing information exchanged via BGP supports only the destination-based forwarding paradigm, which assumes that a router forwards a packet based solely on the destination address carried in the IP header of the packet. This, in turn, reflects the set of policy decisions that can (and cannot) be enforced using BGP. BGP can support only those policies conforming to the destination-based forwarding paradigm.

Principle Description

For more BGP information please reference [RFC 1771, RFC 4271].

5.6.6 Configuration

Configure EBGP

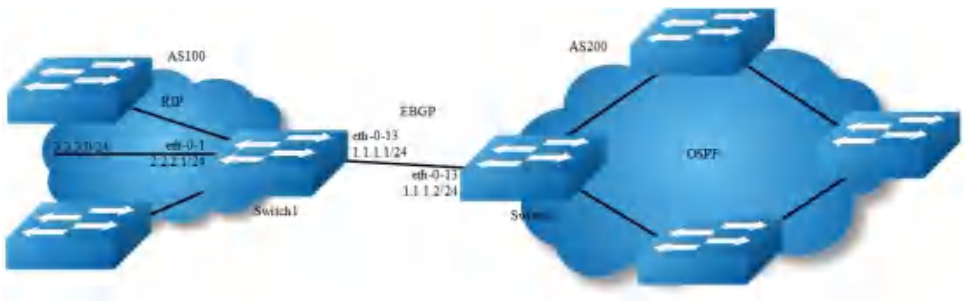


Figure 5-19 EBGP

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes

Switch1:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
```

```
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit

Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 2.2.2.1/24
Switch(config-if)# exit
```

Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.2/24
Switch(config-if)# exit
```

step 3 Configure a static route

Switch1:

```
Switch(config)# ip route 3.3.3.0/24 2.2.2.2
```

step 4 Configure the Routing process and set the router id, set the neighbor, associate the network, and set the redistribute attributes

Switch1:

```
Switch(config)# router bgp 100
Switch(config-router)# bgp router-id 10.10.10.10
Switch(config-router)# neighbor 1.1.1.2 remote-as 200
Switch(config-router)# neighbor 1.1.1.2 ebgp-multihop
Switch(config-router)# network 4.0.0.0/8
Switch(config-router)# redistribute static
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

Switch2:

```
Switch(config)# router bgp 200
Switch(config-router)# bgp router-id 11.11.11.11
Switch(config-router)# neighbor 1.1.1.1 remote-as 100
Switch(config-router)# neighbor 1.1.1.1 ebgp-multihop
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Switch1:

```
Switch# show ip bgp neighbors
BGP neighbor is 1.1.1.2, remote AS 200, local AS 100, external link
 BGP version 4, remote router ID 11.11.11.11
 BGP state = Established, up for 00:00:10
 Last read 00:00:08, hold time is 180, keepalive interval is 60 seconds
 Neighbor capabilities:
  Route refresh: advertised and received (old and new)
```

```
  4-Octet ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
 Received 5 messages, 1 notifications, 0 in queue
 Sent 8 messages, 0 notifications, 0 in queue
 Route refresh request: received 0, sent 0
 Minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
 BGP table version 1, neighbor version 1
 Index 0, Offset 0, Mask 0x1
 1 accepted prefixes
 1 announced prefixes

Connections established 2; dropped 1
 External BGP neighbor may be up to 255 hops away.
Local host: 1.1.1.1, Local port: 48036
Foreign host: 1.1.1.2, Foreign port: 179
Nexthop: 1.1.1.1
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 00:00:18, due to BGP Notification received
Notification Error Message: (Cease/Other Configuration Change.)
```

Switch2:

```
Switch# show ip bgp neighbors
BGP neighbor is 1.1.1.1, remote AS 100, local AS 200, external link
 BGP version 4, remote router ID 10.10.10.10
 BGP state = Established, up for 00:00:16
 Last read 00:00:15, hold time is 180, keepalive interval is 60 seconds
 Neighbor capabilities:
  Route refresh: advertised and received (old and new)
  4-Octet ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
 Received 4 messages, 0 notifications, 0 in queue
 Sent 6 messages, 1 notifications, 0 in queue
 Route refresh request: received 0, sent 0
 Minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
 BGP table version 1, neighbor version 1
 Index 1, Offset 0, Mask 0x2
 1 accepted prefixes
 1 announced prefixes

Connections established 1; dropped 1
 External BGP neighbor may be up to 255 hops away.
Local host: 1.1.1.2, Local port: 179
Foreign host: 1.1.1.1, Foreign port: 48036
Nexthop: 1.1.1.2
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 00:00:19, due to BGP Notification sent
Notification Error Message: (Cease/Other Configuration Change.)
```

Configure IBGP

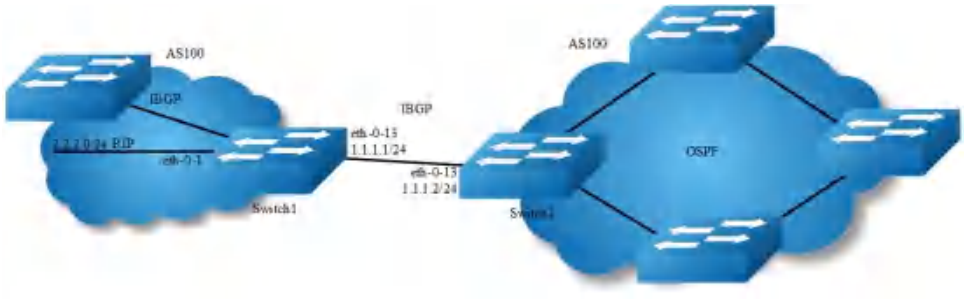


Figure 5-20 IBGP

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes

Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 2.2.2.1/24
Switch(config-if)# exit

Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit

Switch(config)#interface loopback 0
Switch(config-if)# ip address 10.10.10.10/32
Switch(config-if)# exit
```

Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.2/24
Switch(config-if)# exit

Switch(config)# interface loopback 0
Switch(config-if)# ip address 11.11.11.11/32
Switch(config-if)# exit
```

step 3 Configure a static route

Switch1:

```
Switch (config)# ip route 11.11.11.11/32 1.1.1.2
```

Switch2:

```
Switch (config)# ip route 10.10.10.10/32 1.1.1.1
```

step 4 Configure the Routing process and set the router id, set the neighbor, associate the network, and set the redistribute attributes

Switch1:

```
Switch(config)# router bgp 100
Switch(config-router)# bgp router-id 10.10.10.10
Switch(config-router)# neighbor 11.11.11.11 remote-as 100
Switch(config-router)# neighbor 11.11.11.11 update-source loopback 0
Switch(config-router)# network 4.0.0.0/8
Switch(config-router)# redistribute static
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

Switch2:

```
Switch(config)# router bgp 100
Switch(config-router)# bgp router-id 11.11.11.11
Switch(config-router)# neighbor 10.10.10.10 remote-as 100
Switch(config-router)# neighbor 10.10.10.10 update-source loopback 0
Switch(config-router)# redistribute connected
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Switch1:

```
Switch# show ip bgp neighbors
BGP neighbor is 11.11.11.11, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 11.11.11.11
 BGP state = Established, up for 00:00:06
 Last read 00:00:06, hold time is 180, keepalive interval is 60 seconds
 Neighbor capabilities:
  Route refresh: advertised and received (old and new)
  4-Octet ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
 Received 4 messages, 1 notifications, 0 in queue
 Sent 4 messages, 0 notifications, 0 in queue
 Route refresh request: received 0, sent 0
 Minimum time between advertisement runs is 5 seconds
 Update source is loopback0
For address family: IPv4 Unicast
 BGP table version 1, neighbor version 1
 Index 0, Offset 0, Mask 0x1
 0 accepted prefixes
 0 announced prefixes

 Connections established 2; dropped 1
Local host: 10.10.10.10, Local port: 179
Foreign host: 11.11.11.11, Foreign port: 38689
Nexthop: 10.10.10.10
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 00:00:15, due to BGP Notification received
Notification Error Message: (Cease/Other Configuration Change.)
```

Switch2:

```
Switch# show ip bgp neighbors
BGP neighbor is 10.10.10.10, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 10.10.10.10
 BGP state = Established, up for 00:00:00
 Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
 Neighbor capabilities:
  Route refresh: advertised and received (old and new)
  4-Octet ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
 Received 4 messages, 0 notifications, 0 in queue
 Sent 4 messages, 1 notifications, 0 in queue
 Route refresh request: received 0, sent 0
 Minimum time between advertisement runs is 5 seconds
 Update source is loopback0
For address family: IPv4 Unicast
 BGP table version 1, neighbor version 0
 Index 0, Offset 0, Mask 0x1
 0 accepted prefixes
 0 announced prefixes

 Connections established 2; dropped 1
Local host: 11.11.11.11, Local port: 38689
Foreign host: 10.10.10.10, Foreign port: 179
Nexthop: 11.11.11.11
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 00:00:10, due to BGP Notification sent
Notification Error Message: (Cease/Other Configuration Change.)
```

5.6.7 Application cases

N/A

5.7 Configuring ISIS

5.7.1 Overview

Function Introduction

Intermediate System to Intermediate System(ISIS) is a link state routing protocol that uses the shortest path first (SPF) algorithm for routing algorithms. It is actually very similar to OSPF. It also uses Hello protocol to find neighboring nodes and uses a propagation protocol to send link information. ISIS can operate on different subnets, including broadcast LANs, WANs and point-to-point links.

Principle Description

NET

The Network Entity Title (NET) indicates the network layer information of the IS itself, excluding the transport layer information (SEL = 0). It can be regarded as a special kind of NSAP, that is, an NSAP address whose SEL is 0. Therefore, NET is the same length as NSAP, with a maximum of 20 bytes and a minimum of 8 bytes. Generally, a router can be configured with a NET. When an area needs to be re-divided, for example, multiple areas are combined, or an area is divided into multiple areas. In this case, multiple NETs can be configured during reconfiguration Still can guarantee the correctness of the route. As a router default can be configured up to three regional addresses, so up to only three NET configuration. When configuring multiple NETs, you must ensure that their System IDs are the same. For example, NET is: ab.cdef.1234.5678.9abc.00, where Area is ab.cdef, System ID is 1234.5678.9abc, and SEL is 00.

ISIS area

1. Two-level structure In order to support large-scale routing networks, IS-IS adopts a two-level hierarchical structure in the routing domain. A large routing domain is divided into one or more Areas. Routes in the area are managed by Level-1 routers and inter-area routes are managed by Level-2 routers.

2. Level-1 and Level-2

- Level-1 router The Level-1 router is responsible for the intra-area routing. It only establishes the neighbor relationship with the Level-1 and Level-1-2 routers in the same area and maintains a Level-1 LSDB. The Level-1 router contains the routing information of the area. The packet is forwarded to the nearest Level-1-2 router.
- Level-2 router The Level-2 router is responsible for inter-area routing. It can establish the neighbor relationship with Level-2 and Level-1-2 routers in the same area or other areas and maintains a Level-2 LSDB. The LSDB contains inter-area routing information. All Level-2 routers and Level-1-2 routers form the backbone network in the routing domain and are responsible for communication between different areas. The Level-2 routers in the routing domain must be physically contiguous to ensure continuity of the backbone network. Only Level-2 routers can exchange data packets or routing information with routers outside the routing domain.
- Level-1-2 router Routers belonging to Level-1 and Level-2 are called Level-1-2 routers. They can establish Level-1 neighbor relationships with Level-1 and Level-1-2 routers in the same area or with Level-1 routers in the same area or with other areas Level-2 and Level-1-2 routers form a Level-2 neighbor relationship. Level-1 routers must pass through Level-1-2 routers to connect to other areas. The Level-1-2 router maintains two LSDBs. The Level-1 LSDB is used for intra-area routing. The Level-2 LSDB is used for inter-area routing.

3. The route type of the interface For a router of type Level-1-2, you may need to set up Level-1 adjacency with only one peer and establish only Level-2 adjacency with the other peer. You can set the routing layer type of the corresponding interface to limit the adjacencies that can be established on the interface. For example, Level-1 interfaces can only establish Level-1 adjacencies. Level-2 interfaces can only establish Level-2 adjacencies. For Level-1-2 routers, you can also save bandwidth by preventing Level-1 Hello packets from being sent to the Level-2 backbone network by configuring some interfaces as Level-2.

4. Route infiltration (Route Leaking) Generally, an IS-IS area is also called a Level-1 area. Routes in the area are managed by Level-1 routers. All Level-2 routers form a Level-2 area. Therefore, an IS-IS routing domain can contain multiple Level-1 areas but only one Level-2 area.

5.7.2 Configuration

Basic ISIS Parameters Configuration

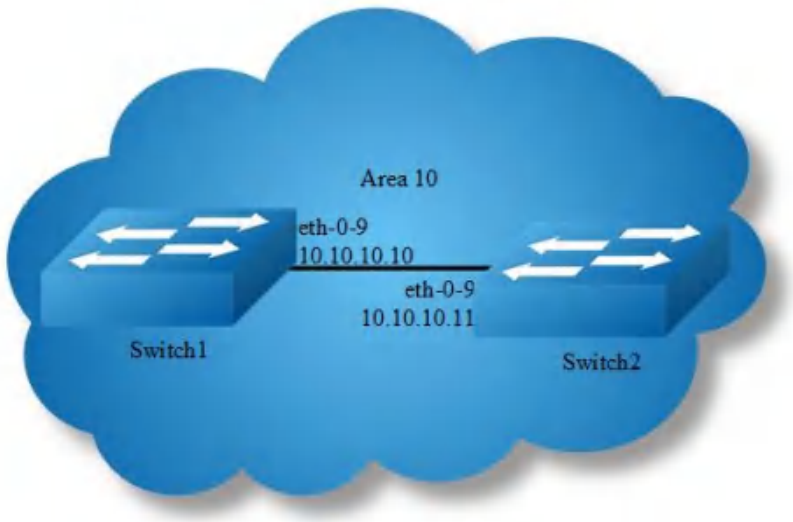


Figure 5-21 RIPng

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the Routing process and set the net

configuration for Switch1:

```
Switch(config)# router isis
Switch(config-router)# net 10.0000.0000.0001.00
Switch(config-router)# exit
```

configuration for Switch2:

```
Switch(config)# router isis
Switch(config-router)# net 10.0000.0000.0002.00
Switch(config-router)# exit
```

step 3 Enable ipv4 isis on the interface

configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# ip router isis
Switch(config)# interface loopback 0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# ip router isis
```

configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.11/24
Switch(config-if)# ip router isis
Switch(config)# interface loopback 0
Switch(config-if)# ip address 2.2.2.2/32
Switch(config-if)# ip router isis
```

step 4 Validation

Display the result on Switch1:

```
Switch# show clns neighbors

Area (null):
System Id     Interface   SNPA            State  Holdtime  Type Protocol
0000.0000.0002 eth-0-9    4a98.a825.3d00     Up    21      L1   IS-IS
                                Up    21      L2   IS-IS

Switch# show isis database verbose
Area (null):
IS-IS Level-1 Link State Database:
LSPID             LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
0000.0000.0001.00-00* 0x00000004   0x3244       1082           0/0/0
 Area Address: 10
 NLPID:       IPV4
 IP Address:  10.10.10.10
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 1.1.1.1 255.255.255.255
0000.0000.0001.01-00* 0x00000001   0x21B9       895            0/0/0
 Metric:  0        IS 0000.0000.0001.00
 Metric:  0        IS 0000.0000.0002.00
0000.0000.0002.00-00  0x00000004   0xFA75       1076           0/0/0
 Area Address: 10
```

```
 NLPID:      IPV4
 IP Address:  10.10.10.11
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 2.2.2.2 255.255.255.255

IS-IS Level-2 Link State Database:
LSPID             LSP Seq Num  LSP Checksum  LSP Holdtime     ATT/P/OL
0000.0000.0001.00-00* 0x00000005  0xFCCE       1109           0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.10
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  20       IP 2.2.2.2 255.255.255.255
 Metric:  10       IP 1.1.1.1 255.255.255.255
0000.0000.0001.01-00* 0x00000001  0x21B9       895            0/0/0
 Metric:  0        IS 0000.0000.0001.00
 Metric:  0        IS 0000.0000.0002.00
0000.0000.0002.00-00  0x00000005  0x7B4E       1107           0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.11
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 2.2.2.2 255.255.255.255
 Metric:  20       IP 1.1.1.1 255.255.255.255

Switch# show ip isis route

Codes: C - connected, E - external, L1 - IS-IS level-1, L2 - IS-IS level-2
     ia - IS-IS inter area, D - discard, e - external metric

Area (null):
    Destination       Metric     Next-Hop          Interface     Tag
C   1.1.1.1/32        10         --                loopback0          0
L1  2.2.2.2/32        20         10.10.10.11       eth-0-9            0
C   10.10.10.0/24     10         --                eth-0-9            0
```

Display the result on Switch2:

```
Switch# show clns neighbors

Area (null):
System Id     Interface   SNPA            State  Holdtime  Type Protocol
0000.0000.0001 eth-0-9    a821.1873.ae00     Up    9       L1   IS-IS
                                Up    9       L2   IS-IS

Switch# show isis database verbose
Area (null):
IS-IS Level-1 Link State Database:
LSPID             LSP Seq Num  LSP Checksum  LSP Holdtime     ATT/P/OL
0000.0000.0001.00-00  0x00000004  0x3244       934            0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.10
```

```
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 1.1.1.1 255.255.255.255
0000.0000.0001.01-00  0x00000001   0x21B9      745           0/0/0
 Metric:  0        IS 0000.0000.0001.00
 Metric:  0        IS 0000.0000.0002.00
0000.0000.0002.00-00* 0x00000004   0xFA75      930           0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.11
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 2.2.2.2 255.255.255.255

IS-IS Level-2 Link State Database:
LSPID             LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
0000.0000.0001.00-00  0x00000005   0xFCCE      961           0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.10
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  20       IP 2.2.2.2 255.255.255.255
 Metric:  10       IP 1.1.1.1 255.255.255.255
0000.0000.0001.01-00  0x00000001   0x21B9      747           0/0/0
 Metric:  0        IS 0000.0000.0001.00
 Metric:  0        IS 0000.0000.0002.00
0000.0000.0002.00-00* 0x00000005   0x7B4E      960           0/0/0
 Area Address: 10
 NLPID:      IPV4
 IP Address:  10.10.10.11
 Metric:  10       IS 0000.0000.0001.01
 Metric:  10       IP 10.10.10.0 255.255.255.0
 Metric:  10       IP 2.2.2.2 255.255.255.255
 Metric:  20       IP 1.1.1.1 255.255.255.255

Switch# show ip isis route

Codes: C - connected, E - external, L1 - IS-IS level-1, L2 - IS-IS level-2
    ia - IS-IS inter area, D - discard, e - external metric

Area (null):
   Destination       Metric    Next-Hop         Interface    Tag
L1  1.1.1.1/32       20        10.10.10.10      eth-0-9        0
C   2.2.2.2/32       10        --               loopback0      0
C   10.10.10.0/24    10        --               eth-0-9        0
```

5.7.3 Application cases

N/A

6 Multicast Configuration Guide

6.1 Configuring IP Multicast-Routing

6.1.1 Overview

Function Introduction

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmit of multicast data.

Distribution of stock quotes, video transmissions such as news services and remote classrooms, and video conferencing are all examples of applications that use multicast routing.

- Internet Group Management Protocol (IGMP) is used among hosts on a LAN and the routers (and multilayer switches) on that LAN to track the multicast groups of which hosts are members.
- Protocol-Independent Multicast (PIM) protocol is used among routers and multilayer switches to track which multicast packets to forward to each other and to their directly connected LANs. PIM has two modes: Sparse-mode and Dense-mode.

Principle Description

N/A

6.1.2 Configuration

Configuring multicast route limit

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 set the limit of the multicast route

```
Switch(config)# ip multicast route-limit 1000
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ip mroute route-limit
Max Multicast Route Limit Number: 1000
Multicast Route Limit Warning Threshold: 1000
Multicast Hardware Route Limit: 1023
Current Multicast Route Entry Number: 0
```

6.1.3 Application cases

N/A

6.2 Configuring IGMP

6.2.1 Overview

Function Introduction

To participate in IP multicasting, multicast hosts, routers, and multilayer switches must have the IGMP operating. This protocol defines the querier and host roles:

- A querier is a network device that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver that sends report messages (in response to query messages) to inform a querier of a host membership.
- A set of queries and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use IGMP messages to join and leave multicast groups. – Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message. Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group.

A host can be a member of more than one multicast group at a time. How active a multicast group is and what members it has can vary from group to group and from time to time. A multicast group can be active for a long time, or it can be very short-lived. Membership in a group can constantly change. A group that has members can have no activity.

IGMP packets are sent using these IP multicast group addresses:

- IGMP general queries are destined to the address 224.0.0.1 (all systems on a subnet).
- IGMP group-specific queries are destined to the group IP address for which the switch is querying.
- IGMP group membership reports are destined to the group IP address for which the switch is reporting.
- IGMP Version 2 (IGMPv2) leave messages are destined to the address 224.0.0.2 (all-multicast-routers on a subnet). In some old host IP stacks, leave messages might be destined to the group IP address rather than to the all-routers address.

Principle Description

Reference to RFC 1112， RFC 2236， RFC 3376

6.2.2 Configuration

There is no explicit command to enable IGMP, which is always combined with PIM-SM. When PIM-SM is enabled on an interface, IGMP will be enabled automatically on this interface, vice versa. But notice, before IGMP can work, IP Multicast-routing must be enabled globally firstly. We support build IGMP group record by learning IGMP packets or configuring static IGMP group by administrator.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ip multicast-routing globally

```
Switch(config)# ip multicast-routing
```

step 3 Enter the interface configure mode， set the attributes and ip address

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.10/24
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.11.10/24
Switch(config-if)# exit
```

step 4 Enable pim-sm on the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
```

step 5 Set the attributes for igmp

```
Switch(config)# interface eth-0-1
Switch(config-if)# ip igmp version 2
Switch(config-if)# ip igmp query-interval 120
Switch(config-if)# ip igmp query-max-response-time 12
Switch(config-if)# ip igmp robustness-variable 3
Switch(config-if)# ip igmp last-member-query-count 3
Switch(config-if)# ip igmp last-member-query-interval 2000
Switch(config-if)# exit
```

step 6 Set the maxinum igmp group count(optional)

The maxinum igmp group count is limited globally or per-interface.

```
Switch(config)# ip igmp limit 2000

Switch(config)# interface eth-0-1
Switch(config-if)# ip igmp limit 1000
```

step 7 Set a static igmp group

```
Switch(config-if)# ip igmp static-group 228.1.1.1
Switch(config-if)# exit
```

step 8 Set igmp proxy(optional)

```
Switch(config)# interface eth-0-1
Switch(config-if)# ip igmp proxy-service
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# ip igmp mroute-proxy eth-0-1
Switch(config-if)# exit
```

step 9 Exit the configure mode

```
Switch(config)# end
```

step 10 Validation

Use the following command to display the information of igmp interfaces:

```
Switch# show ip igmp interface
Interface eth-0-1 (Index 1)
 IGMP Inactive, Version 2 (default) proxy-service
 IGMP host version 2
 IGMP global limit is 2000
 IGMP global limit states count is currently 0
 IGMP interface limit is 1000
 IGMP interface has 0 group-record states
 IGMP activity: 0 joins, 0 leaves
 IGMP query interval is 120 seconds
 IGMP querier timeout is 366 seconds
 IGMP max query response time is 12 seconds
 Last member query response interval is 2000 milliseconds
 Group Membership interval is 372 seconds
 Last memeber query count is 3
 Robustness Variable is 3
Interface eth-0-2 (Index 2)
 IGMP Inactive, Version 2 (default)
```

```
IGMP mroute-proxy interface is eth-0-1
IGMP global limit is 2000
IGMP global limit states count is currently 0
IGMP interface limit is 16384
IGMP interface has 0 group-record states
IGMP activity: 0 joins, 0 leaves
IGMP query interval is 125 seconds
IGMP querier timeout is 255 seconds
IGMP max query response time is 10 seconds
Last member query response interval is 1000 milliseconds
Group Membership interval is 260 seconds
Last memeber query count is 2
Robustness Variable is 2
```

Use the following command to display the information of groups:

```
Switch# show ip igmp groups
IGMP Connected Group Membership
Group Address   Interface      Uptime   Expires Last Reporter
228.1.1.1       eth-0-1        00:00:05 stopped  -
```

6.2.3 Application cases

N/A

6.3 Configuring PIM-SM

6.3.1 Overview

Function Introduction

The Protocol Independent Multicasting-Sparse Mode (PIM-SM) is a multicast routing protocol designed to operate efficiently across Wide Area Networks (WANs) with sparsely distributed groups. It helps network nodes that are geographically dispersed to conserve bandwidth, and reduces traffic by simultaneously delivering a single stream of information to multiple locations.

PIM-SM uses the IP multicast model of receiver-initiated membership, supporting both shared and shortest-path trees, and uses soft-state mechanisms to adapt to changing network conditions. It relies on a topology-gathering protocol to populate a multicast routing table with routes.

Principle Description

The PIM-SM module is based on the following IETF standard: RFC 4601

Terminology:

- **Rendezvous Point (RP)**: A Rendezvous Point (RP) router is configured as the root of the non-source-specific distribution tree for a multicast group. Join messages from receivers for a group are sent towards the RP. Data from senders is sent to the RP so that receivers can discover who the senders are, and receive traffic destined for the group.
- **Multicast Routing Information Base (MRIB)**: The MRIB is a multicast topology table derived from the unicast routing table. In PIM-SM, the MRIB is used to decide where to send Join/Prune messages. It also provides routing metrics for destination addresses. These metrics are used when sending and processing Assert messages.
- **Reverse Path Forwarding**: Reverse Path Forwarding (RPF) is a concept of an optimized form of flooding, where the router accepts a packet from SourceA through Interface IF1 only if IF1 is the interface the router would use in order to reach SourceA. It determines whether the interface is correct by consulting its unicast routing tables. The packet that arrives through interface IF1 is forwarded because the routing table lists this interface as the shortest path to the network. The router's unicast routing table determines the shortest path for the multicast packets. Because a router accepts a packet from only one neighbor, it floods the packet only once, meaning that (assuming point-to-point links) each packet is transmitted over each link once in each direction.
- **Tree Information Base (TIB)**: The TIB is the collection of state at a PIM router storing the state of all multicast distribution trees at that router. It is created by receiving Join/Prune messages, Assert messages, and IGMP information from local hosts.
- **Upstream**: Towards to root of the tree. The root of the tree might be either the Source or the RP.

- **Downstream**: Away from the root of the tree. The root of tree might be either the Source or the RP.
- **Source-Based Trees**: In the Source-Based Trees concept, the forwarding paths are based on the shortest unicast path to the source. If the unicast routing metric is hop counts, the branches of the multicast Source-Based Trees are minimum hop. If the metric is delay, the branches are minimum delay. For every multicast source, there is a corresponding multicast tree that directly connects the source to all receivers. All traffic to the members of an associated group passes along the tree made for their source. Source-Based Trees have two entries with a list of outgoing interfaces– the source address and the multicast group.
- **Shared Trees**: Shared trees or RP trees (RPT) rely on a central router called the Rendezvous Point (RP) that receives all traffic from the sources, and forwards that traffic to the receivers. All hosts might not be receivers. There is a single tree for each multicast group, regardless of the number of sources. Only the routers on the tree know about the group, and information is sent only to interested receivers. With an RP, receivers have a place to join, even if no source exists. The shared tree is unidirectional, and information flows only from the RP to the receivers. If a host other than the RP has to send data on the tree, the data must first be tunneled to the RP, and then multicast to the members. This means that even if a receiver is also a source, it can only use the tree to receive packets from the RP, and not to send packets to the RP (unless the source is located between the RP and the receivers).
- **Bootstrap Router (BSR)**: When a new multicast sender starts sending data packets, or a new receiver starts sending the Join message towards the RP for that multicast group, it needs to know the next-hop router towards the RP. The BSR provides group-to-RP mapping information to all the PIM routers in a domain, allowing them to map to the correct RP address.
- **Sending out Hello Messages**: PIM routers periodically send Hello messages to discover neighboring PIM routers. Hello messages are multicast using the address 224.0.0.13 (ALL-PIM-ROUTERS group). Routers do not send any acknowledgement that a Hello message was received. A hold time value determines the length of time for which the information is valid. In PIM-SM, a downstream receiver must join a group before traffic is forwarded on the interface.
- **Electing a Designated Router**: In a multi-access network with multiple routers connected, one of them is selected to act as a designated router (DR) for a given period of time. The DR is responsible for sending Join/Prune messages to the RP for local members.
- **Determining the RP**: PIM-SM uses a Bootstrap Router (BSR) to originate Bootstrap messages, and to disseminate RP information. The messages are multicast to the group on each link. If the BSR is not apparent, the routers flood the domain with advertisements. The router with the highest priority (if priorities are same, the higher IP address applies) is selected to be the RP. Routers receive and store Bootstrap messages originated by the BSR. When a DR gets a membership indication from IGMP for (or a data packet from) a directly connected host, for a group for which it has no entry, the DR maps the group address to one of the candidate RPs that can service that group. The DR then sends a Join/Prune message towards that RP. In a small domain, the RP can also be configured statically.
- **Joining the Shared Tree**: To join a multicast group, a host sends an IGMP message to its upstream router, after which the router can accept multicast traffic for that group. The router sends a Join message to its upstream PIM neighbor in the direction of the RP. When a router receives a Join message from a downstream router, it checks to see if a state exists for the group in its multicast routing table. If a state already exists, the Join message has reached the shared tree, and the interface from which the message was received is entered in the Outgoing Interface list. If no state exists, an entry is created, the interface is entered in the Outgoing Interface list, and the Join message is again sent towards the RP.
- **Registering with the RP**: A DR can begin receiving traffic from a source without having a Source or a Group state for that source. In this case, the DR has no information on how to get multicast traffic to the RP through a tree. When the source DR receives the initial multicast packet, it encapsulates it in a Register message, and unicasts it to the RP for that group. The RP decapsulates each Register message, and forwards the extracted data packet to downstream members on the RPT. Once the path is established from the source to the RP, the DR begins sending traffic to the RP as standard IP multicast packets, as well as encapsulated within Register messages. The RP temporarily receives packets twice. When the RP detects the normal multicast packets, it sends a Register-Stop message to the source DR, meaning it should stop sending register packets.
- **Sending Register-Stop Messages**: When the RP begins receiving traffic from the source, both as Register messages and as unencapsulated IP packets, it sends a Register-Stop message to the DR. This notifies the DR that the traffic is now being received as standard IP multicast packets on the SPT. When the DR receives this message, it stops encapsulating traffic in Register messages.
- **Pruning the Interface**: Routers attached to receivers send Prune messages to the RP to disassociate the source from the RP. When an RP receives a Prune message, it no longer forwards traffic from the source indicated in the Prune message. If all members of a multicast group are pruned, the IGMP state of the DR is deleted, and the interface is removed from the Source and Group lists of the group.
- **Forwarding Multicast Packets**: PIM-SM routers forward multicast traffic onto all interfaces that lead to receivers that have explicitly joined a multicast group. Messages are sent to a group address in the local subnetwork, and have a Time to Live (TTL) of 1. The router performs an RPF check, and forwards the packet. Traffic that arrives on the correct interface is sent onto all outgoing interfaces that lead to downstream receivers if the downstream router has sent a join to this router, or is a member of this group.

6.3.2 Configuration

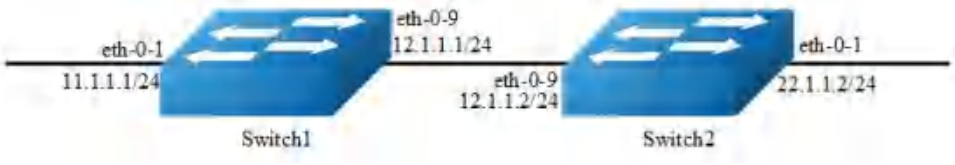


Figure 6-1 Pim sm

PIM-SM is a soft-state protocol. The main requirement is to enable PIM-SM on desired interfaces, and configure the RP information correctly, through static or dynamic methods. All multicast group states are maintained dynamically as the result of IGMP Report/Leave and PIM Join/Prune messages.

This section provides PIM-SM configuration examples for two relevant scenarios. The following graphic displays the network topology used in these examples:

Configuring General PIM Sparse-mode (static RP)

In this example, using the above topology, Switch1 is the Rendezvous Point (RP), and all routers are statically configured with RP information. While configuring the RP, make sure that:

Every router includes the ip pim rp-address 11.1.1.1 statement, even if it does not have any source or group member attached to it.

There is only one RP address for a group scope in the PIM domain.

All interfaces running PIM-SM must have sparse-mode enabled.

Here is a sample configuration:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode， set the attributes and ip address, and enable pim-sm

Configuring on Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 11.1.1.1/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.1/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
```

Configuring on Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 22.1.1.2/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
```

```
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.2/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
```

step 3 Add static routes

Configuring on Switch1:

```
Switch(config)# ip route 22.1.1.0/24 12.1.1.2
```

Configuring on Switch2:

```
Switch(config)# ip route 11.1.1.0/24 12.1.1.1
```

step 4 Configure the static rp address

```
Switch(config)# ip pim rp-address 11.1.1.1
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the following command to show ip pim sparse-mode rp mapping. 11.1.1.1 is the RP for all multicast groups 224.0.0.0/4 which is statically configured.

```
Switch# show ip pim sparse-mode rp mapping
PIM group-to-RP mappings
Group(s): 224.0.0.0/4, Static
  RP: 11.1.1.1
    Uptime: 00:08:21
```

Use the following command to show the interface information:

```
Switch# show ip pim sparse-mode interface
Address       Interface VIFindex Ver/  Nbr    DR    DR       HoldTime
                        Mode   Count  Prior
11.1.1.1      eth-0-1  2       v2/S  0      1     11.1.1.1      105
12.1.1.1      eth-0-9  0       v2/S  1      1     12.1.1.2      105
```

Use the following command to show the pim sparse-mode multicast routes:

Switch1：

```
Switch# show ip pim sparse-mode mroute detail
IP Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 0
(S,G,rpt) Entries: 0
FCR Entries: 0
(*, 224.1.1.1) Uptime: 00:01:32
 RP: 11.1.1.1, RPF nbr: None, RPF idx: None
 Upstream:
  State: JOINED, SPT Switch: Enabled, JT: off
  Macro state: Join Desired,
 Downstream:
  eth-0-9:
    State: JOINED, ET Expiry: 179 secs, PPT: off
```

```
  Assert State: NO INFO, AT: off
   Winner: 0.0.0.0, Metric: 4294967295, Pref: 4294967295, RPT bit: on
  Macro state: Could Assert, Assert Track
Join Olist:
 eth-0-9
```

Switch2：

```
Switch# show ip pim sparse-mode mroute detail
IP Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 0
(S,G,rpt) Entries: 0
FCR Entries: 0
(*, 224.1.1.1) Uptime: 00:00:43
 RP: 11.1.1.1, RPF nbr: 12.1.1.1, RPF idx: eth-0-9
 Upstream:
  State: JOINED, SPT Switch: Enabled, JT Expiry: 18 secs
  Macro state: Join Desired,
 Downstream:
  eth-0-1:
   State: NO INFO, ET: off, PPT: off
   Assert State: NO INFO, AT: off
    Winner: 0.0.0.0, Metric: 4294967295, Pref: 4294967295, RPT bit: on
   Macro state: Could Assert, Assert Track
 Local Olist:
  eth-0-1
```

Configuring General PIM Sparse-mode (dynamic RP)

A static configuration of RP works for a small, stable PIM domain; however, it is not practical for a large and not-suitable internet work. In such a network, if the RP fails, the network administrator might have to change the static configurations on all PIM routers. Another reason for choosing dynamic configuration is a higher routing traffic leading to a change in the RP.

We use the BSR mechanism to dynamically maintain the RP information. For configuring RP dynamically in the above scenario, Switch1 on eth-0-1 and Switch2 on eth-0-9 are configured as Candidate RP using the ip pim rp-candidate command. Switch2 on eth-0-9 is also configured as Candidate BSR. Since no other router has been configured as Candidate BSR, the Switch2 becomes the BSR router, and is responsible for sending group-to-RP mapping information to all other routers in this PIM domain.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode，set the attributes and ip address, and enable pim-sm

Configuring on Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 11.1.1.1/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.1/24
```

```
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
```

Configuring on Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 22.1.1.2/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.2/24
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
```

step 3 Add static routes

Configuring on Switch1:

```
Switch(config)# ip route 22.1.1.0/24 12.1.1.2
```

Configuring on Switch2:

```
Switch(config)# ip route 11.1.1.0/24 12.1.1.1
```

step 4 Configure the rp candidate

Configuring on Switch1:

```
Switch(config)# ip pim rp-candidate eth-0-1
```

Configuring on Switch2:

```
Switch(config)# ip pim rp-candidate eth-0-9
Switch(config)# ip pim bsr-candidate eth-0-9
```



The highest priority router is chosen as the RP. If two or more routers have the same priority, a hash function in the BSR mechanism is used to choose the RP, to make sure that all routers in the PIM-domain have the same RP for the same group. Use the ip pim rp-candidate IFNAME PRIORITY command to change the default priority of any candidate RP.

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Use the show ip pim sparse-mode rp mapping command to display the group-to-RP mapping details. The output displays information about RP candidates. There are two RP candidates for the group range 224.0.0.0/4. RP Candidate 11.1.1.1 has a default priority of 192, whereas, RP Candidate 12.1.1.2 has been configured to have a priority of 2. Since RP candidate 12.1.1.2 has a higher priority, it is selected as RP for the multicast group 224.0.0.0/24. Only permit filters would be cared in group list.

Switch2：

```
switch# show ip pim sparse-mode rp mapping
PIM group-to-RP mappings
This system is the bootstrap router (v2)
Group(s): 224.0.0.0/4
 RP: 12.1.1.2
```

```
  Info source: 12.1.1.2, via bootstrap, priority 2
     Uptime: 01:55:20, expires: 00:02:17
RP: 11.1.1.1
  Info source: 11.1.1.1, via bootstrap, priority 192
     Uptime: 01:55:23, expires: 00:02:13
```

To display information about the RP router for a particular group, use the following command. This output displays that 12.1.1.2 has been chosen as the RP for the multicast group 224.1.1.1.

Switch2：

```
switch# show ip pim sparse-mode rp-hash 224.1.1.1
  RP: 12.1.1.2
  Info source: 12.1.1.2, via bootstrap
```

After RP information reaches all PIM routers in the domain, various state machines maintain all routing states as the result of Join/Prune from group membership. To display information on interface details and the multicast routing table, refer to the Configuring RP Statically section above.

Configuring Boostrap Router

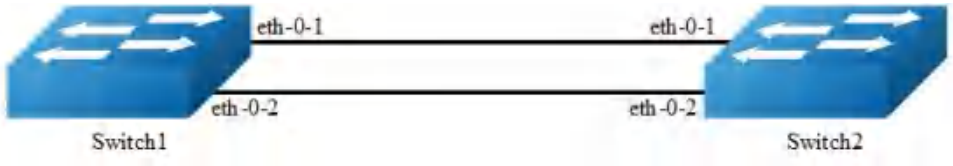


Figure 6-2 bsr

Every PIM multicast group needs to be associated with the IP address of a Rendezvous Point (RP). This address is used as the root of a group-specific distribution tree whose branches extend to all nodes in the domain that want to receive traffic sent to the group. For all senders to reach all receivers, all routers in the domain use the same mappings of group addresses to RP addresses. In order to determine the RP for a multicast group, a PIM router maintains a collection of group-to-RP mappings, called the RP-Set.

The Bootstrap Router (BSR) mechanism for the class of multicast routing protocols in the PIM domain use the concept of a Rendezvous Point as a means for receivers to discover the sources that send to a particular multicast group. The BSR mechanism is one way that a multicast router can learn the set of group-to-RP mappings required in order to function.

Some of the PIM routers within a PIM domain are configured as Candidate-RPs (C-RPs). A subset of the C-RPs will eventually be used as the actual RPs for the domain. An RP configured with a lower value in the priority field has higher a priority.

Some of the PIM routers in the domain are configured to be Candidate-BSRs (C-BSRs). One of these C-BSRs is elected to be the bootstrap router (BSR) for the domain, and all PIM routers in the domain learn the result of this election through BSM (Bootstrap messages). The C-BSR with highest value in priority field is Elected-BSR.

The C-RPs then reports their candidacy to the elected BSR, which chooses a subset of the C-RPs and distributes corresponding group-to-RP mappings to all the routers in the domain through Bootstrap messages.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure the bsr candidate and rp candidate

Switch1：

```
Switch(config)# ip pim bsr-candidate eth-0-1
```

Switch2：

```
Switch(config)# ip pim bsr-candidate eth-0-1 10 25
Switch(config)# ip pim rp-candidate eth-0-1 priority 0
```

step 3 Configure the priority of rp candidate

```
Switch(config)# ip pim rp-candidate eth-0-1 priority 0
```

step 4 Configure the priority of dr and enable receive and send unicast bsm packets

```
Switch(config)# interface eth-0-1
Switch(config-if)# ip pim dr-priority 10
Switch(config-if)# ip pim unicast-bsm
```

step 5 Exit the configure mode

```
Switch(config-if)# end
```

step 6 Validation

Verify the C-BSR state on rtr1

```
Switch# show ip pim sparse-mode bsr-router
PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
BSR address: 20.0.1.21
Uptime: 00:37:12, BSR Priority: 64, Hash mask length: 10
Next bootstrap message in 00:00:04
Role: Candidate BSR
State: Elected BSR
```

Verify the C-BSR state on rtr2

```
Switch# show ip pim sparse-mode bsr-router
PIMv2 Bootstrap information
BSR address: 20.0.1.21
Uptime: 00:02:39, BSR Priority: 64, Hash mask length: 10
Expires: 00:00:03
Role: Candidate BSR
State: Pending BSR
Switch# show ip pim sparse-mode bsr-router
PIMv2 Bootstrap information
BSR address: 20.0.1.21
Uptime: 00:40:20, BSR Priority: 64, Hash mask length: 10
Expires: 00:02:07
Role: Candidate BSR
State: Candidate BSR
```

Verify RP-set information on E-BSR

```
Switch# sh ip pim sparse-mode rp mapping
PIM Group-to-RP Mappings
This system is the Bootstrap Router (v2)
Group(s): 224.0.0.0/4
RP: 20.0.1.11
Info source: 20.0.1.11, via bootstrap, priority 0
Uptime: 00:00:30, expires: 00:02:04
```

Verify RP-set information on C-BSR

```
Switch# show ip pim sparse-mode rp mapping
PIM Group-to-RP Mappings
Group(s): 224.0.0.0/4
RP: 20.0.1.11
Info source: 20.0.1.21, via bootstrap, priority 0
Uptime: 00:00:12, expires: 00:02:18
```

Configuring PIM-SSM feature

The Source Specific Multicast feature is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (no shared trees) are created.

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM).

PIM-SSM can work with PIM-SM on the multicast router. By default, PIM-SSM is disabled.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ssm

Enable by default range：

```
Switch(config)# ip pim ssm default
```

Enable pim-ssm on the switch and set the ssm group range as group range specified in an access list:

```
Switch(config)# ip pim ssm range ipacl
```

The 2 commands above are alternative. The final configuration should over write the previous one and take effect.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config | include pim
ip pim ssm range ipacl
```

6.3.3 Application cases

N/A

6.4 Configuring PIM-DM

6.4.1 Overview

Function Introduction

The Protocol Independent Multicasting-Dense Mode (PIM-DM) is a multicast routing protocol designed to operate efficiently across Wide Area Networks (WANs) with densely distributed groups. It helps network nodes that are geographically dispersed to conserve bandwidth, and reduces traffic by simultaneously delivering a single stream of information to multiple locations.

PIM-DM assumes that when a source starts sending, all down stream systems want to receive multicast datagrams. Initially, multicast datagrams are flooded to all areas of the network. PIM-DM uses RPF to prevent looping of multicast datagrams while flooding. If some areas of the network do not have group members, PIM-DM will prune off the forwarding branch by instantiating prune state.

Prune state has a finite lifetime. When that lifetime expires, data will again be forwarded down the previously pruned branch. Prune state is associated with an (S,G) pair. When a new member for a group G appears in a pruned area, a router can "graft" toward the source S for the group, thereby turning the pruned branch back into a forwarding branch.

Principle Description

The PIM-DM module is based on the following IETF standard: RFC 3973

6.4.2 Configuration

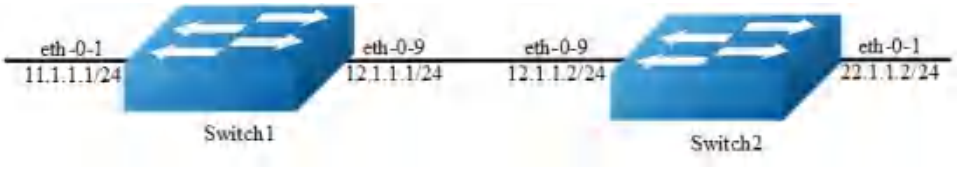


Figure 6-3 Pim dm

PIM-DM is a soft-state protocol. The main requirement is to enable PIM-DM on desired interfaces. All multicast group states are maintained dynamically as the result of IGMP Report/Leave and PIM messages.

This section provides PIM-DM configuration examples for two relevant scenarios. The following graphic displays the network topology used in these examples:

In this example, using the above topology, multicast data stream comes to eth-0-1 of Switch1, host is connected to eth-0-1 of Switch2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode， set the attributes and ip address, and enable pim-dm

Configuring on Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 11.1.1.1/24
Switch(config-if)# ip pim dense-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.1/24
Switch(config-if)# ip pim dense-mode
Switch(config-if)# exit
```

Configuring on Switch2:

```
Switch# configure terminal
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 22.1.1.2/24
Switch(config-if)# ip pim dense-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ip address 12.1.1.2/24
Switch(config-if)# ip pim dense-mode
Switch(config-if)# exit
```

step 3 Add static routes

Configuring on Switch1:

```
Switch(config)# ip route 22.1.1.0/24 12.1.1.2
```

Configuring on Switch2:

```
Switch(config)# ip route 11.1.1.0/24 12.1.1.1
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

The "show ip pim dense-mode interface" command displays the interface details for Switch1.

```
Switch# show ip pim dense-mode interface
Address        Interface VIFIndex Ver/   Nbr
                         Mode   Count
11.1.1.1       eth-0-1  0       v2/D   0
12.1.1.1       eth-0-9  1       v2/D   1
```

The "show ip pim dense-mode neighbor" command displays the neighbor details for Switch1.

```
Switch# show ip pim dense -mode neighbor
Neighbor-Address Interface        Uptime/Expires    Ver
12.1.1.2       eth-0-9          00:01:00/00:01:44 v2
```

The "show ip pim dense-mode mroute detail" command displays the IP multicast routing table.

Switch1：

```
Switch# show ip pim dense-mode mroute
PIM-DM Multicast Routing Table
(11.1.1.2, 225.1.1.1)
 Source directly connected on eth-0-1
 State-Refresh Originator State: Originator
 Upstream IF: eth-0-1
  Upstream State: Forwarding
  Assert State: NoInfo
 Downstream IF List:
  eth-0-9, in 'olist':
    Downstream State: NoInfo
    Assert State: NoInfo
```

Switch2：

```
Switch# show ip pim dense-mode mroute
PIM-DM Multicast Routing Table
(11.1.1.2, 225.1.1.1)
 RPF Neighbor: none
 Upstream IF: eth-0-9
  Upstream State: AckPending
  Assert State: NoInfo
 Downstream IF List:
  eth-0-1, in 'olist':
    Downstream State: NoInfo
    Assert State: NoInfo
```

6.4.3 Application cases

N/A

6.5 Configuring IGMP Snooping

6.5.1 Overview

Function Introduction

Layer 2 switches can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded only to those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also deletes entries per entry if it does not receive IGMP membership reports from the multicast clients. The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send report and are added to the forwarding table entry. The switch forwards only one report per IP multicast group to the multicast router. It creates one entry per VLAN in the Layer 2 forwarding table for each MAC group from which it receives an IGMP report.

Layer 2 multicast groups learned through IGMP snooping are dynamic. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned settings

Limitations And Notice:

VRRP, RIP and OSPF used multicast IP address, so you need to avoid use such multicast IP addresses, which have same multicast MAC address with multicast IP address reserved by VRRP, RIP and OSPF.

VRRP used multicast group address 224.0.0.18, so when igmp snooping and VRRP are working, you need to avoid using multicast group address that matched same mac address with group address 224.0.0.18.

OSPF used multicast group address 224.0.0.5, so when igmp snooping and OSFP are working, you need to avoid using multicast group address that matched same mac address with group address 224.0.0.18.

RIP used multicast group address 224.0.0.9, so when igmp snooping and RIP are working, you need to avoid using multicast group address that matched same mac address with group address 224.0.0.9.

Principle Description

N/A

6.5.2 Configuration

Enable Globally Or Per Vlan

IGMP Snooping can be enabled globally or per vlan. If IGMP Snooping is disabled globally, it can't be active on any vlan even it is enabled on the vlan. If IGMP snooping is enabled globally, it can be disabled on a vlan. On the other hand, the global configuration can overwrite the per vlan configuration. By default, IGMP snooping is enabled globally and per vlan.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enable igmp snooping globally and per-vlan

```
Switch(config)# ip igmp snooping
Switch(config)# ip igmp snooping vlan 1
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the following command to display igmp snooping of a vlan:

```
Switch # show ip igmp snooping vlan 1
Global Igmp Snooping Configuration
----------------------------------------------------
```

```
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Disabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Report-Suppression     :Enabled
Vlan 1
-----------
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Disabled
Igmp Snooping Report-Suppression     :Enabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Group Access-list      :N/A
Igmp Snooping Mrouter Port           :
Igmp Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Fast Leave

When IGMP Snooping fast leave is enabled, the igmp snooping group will be removed at once upon receiving a corresponding igmp report. Otherwise the switch will send out specified igmp specific query, if it doesn't get response in specified period, it will remove the group. By default, igmp snooping fast-leave is disabled globally and per vlan.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enable Fast Leave globally and per-vlan

```
Switch(config)#ip igmp snooping fast-leave
Switch(config)#ip igmp snooping vlan 1 fast-leave
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch # show ip igmp snooping vlan 1
Global Igmp Snooping Configuration
----------------------------------------------------
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Enabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Report-Suppression     :Enabled
Vlan 1
-----------
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Enabled
Igmp Snooping Report-Suppression     :Enabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
```

```
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Group Access-list         :N/A
Igmp Snooping Mrouter Port              :
Igmp Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Querior Parameters

In order for IGMP, and thus IGMP snooping, to function, an multicast router must exist on the network and generate IGMP queries. The tables created for snooping (holding the member ports for a each multicast group) are associated with the querier. Without a querier the tables are not created and snooping will not work.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Set the global attributes of igmp snooping

```
Switch(config)# ip igmp snooping query-interval 100
Switch(config)# ip igmp snooping query-max-response-time 5
Switch(config)# ip igmp snooping last-member-query-interval 2000
Switch(config)# ip igmp snooping discard-unknown
```

step 3 Set the per-vlan attributes of igmp snooping

```
Switch(config)# ip igmp snooping vlan 1 querier address 10.10.10.1
Switch(config)# ip igmp snooping vlan 1 querier
Switch(config)# ip igmp snooping vlan 1 query-interval 200
Switch(config)# ip igmp snooping vlan 1 query-max-response-time 5
Switch(config)# ip igmp snooping vlan 1 querier-timeout 100
Switch(config)# ip igmp snooping vlan 1 last-member-query-interval 2000
Switch(config)# ip igmp snooping vlan 1 discard-unknown
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch # show ip igmp snooping querier
Global Igmp Snooping Querier Configuration
-------------------------------------------------
Version                     :2
Last-Member-Query-Interval (msec) :2000
Last-Member-Query-Count         :2
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)            :100
Global Source-Address           :0.0.0.0
TCN Query Count                 :2
TCN Query Interval (sec)        :10
TCN Query Max Respose Time (sec)  :5
Vlan 1:  IGMP snooping querier status
-----------------------------------------------
Elected querier is : 0.0.0.0
-----------------------------------------------
Admin state                 :Enabled
Admin version               :2
Operational state           :Non-Querier
Querier operational address     :10.10.10.1
Querier configure address       :10.10.10.1
```

```
Last-Member-Query-Interval (msec) :2000
Last-Member-Query-Count         :2
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)            :200
Querier-Timeout (sec)           :100
```

Configuring Mrouter Port

An IGMP Snooping mrouter port is a switch port which is assumed to connect a multicast router. The mrouter port is configured on the vlan or learnt dynamic. When IGMP general query packet or PIMv2 hello packet is received on port of speficified VLAN, this port becomes mrouter port of this vlan. All the igmp queries received on this port will be flooded on the belonged vlan. All the igmp reports and leaves received on this vlan will be forwarded to the mrouter port, directly or aggregated, depending on the report-suppression configuration. In addition, all the multicast traffic on this vlan will be forwarded to this mrouter port.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enable igmp snooping report suppression globally

```
Switch(config)# ip igmp snooping report-suppression
```

step 3 Configure mrouter port, Enable igmp snooping report suppression, and set igmp snooping dynamic mrouter port aging interval for a vlan

```
Switch(config)# ip igmp snooping vlan 1 mrouter interface eth-0-1
Switch(config)# ip igmp snooping vlan 1 report-suppression
Switch(config)# ip igmp snooping vlan 1 mrouter-aging-interval 200
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show ip igmp snooping vlan 1
Global Igmp Snooping Configuration
------------------------------------------------
Igmp Snooping                       :Enabled
Igmp Snooping Fast-Leave            :Disabled
Igmp Snooping Version               :2
Igmp Snooping Robustness Variable   :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Report-Suppression     :Enabled
Vlan 1
-----------
Igmp Snooping                       :Enabled
Igmp Snooping Fast-Leave            :Disabled
Igmp Snooping Report-Suppression    :Enabled
Igmp Snooping Version               :2
Igmp Snooping Robustness Variable   :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Group Access-list      :N/A
Igmp Snooping Mrouter Port          :eth-0-1
Igmp Snooping Mrouter Port Aging Interval(sec) :200
```

Configuring Querier TCN

System supports to adapt the multicast router learning and updating after STP convergence by configuring the TCN querier count and querier interval.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Configuring the TCN querier count and querier interval

```
Switch(config)# ip igmp snooping querier tcn query-count 5
Switch(config)# ip igmp snooping querier tcn query-interval 20
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch # show ip igmp snooping querier
Global Igmp Snooping Querier Configuration
---------------------------------------------------
Version                     :2
Last-Member-Query-Interval (msec) :1000
Max-Query-Response-Time (sec)    :10
Query-Interval (sec)           :125
Global Source-Address          :0.0.0.0
TCN Query Count                :5
TCN Query Interval (sec)       :20
Vlan 1:  IGMP snooping querier status
-----------------------------------------------
Elected querier is : 0.0.0.0
-----------------------------------------------
Admin state                 :Disabled
Admin version               :2
Operational state           :Non-Querier
Querier operational address     :0.0.0.0
Querier configure address       :N/A
Last-Member-Query-Interval (msec) :1000
Max-Query-Response-Time (sec)    :10
Query-Interval (sec)           :125
Querier-Timeout (sec)          :255
```

Configuring Report Suppression

The switch uses IGMP report suppression to forward only one IGMP report per multicast router query to multicast devices. When IGMP router suppression is enabled (the default), the switch sends the first IGMP report from all hosts for a group to all the multicast routers. The switch does not send the remaining IGMP reports for the group to the multicast routers. This feature prevents duplicate reports from being sent to the multicast devices.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enable Report Suppression globally and per-vlan

```
Switch(config)# ip igmp snooping report-suppression
Switch(config)# ip igmp snooping vlan 1 report-suppression
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch # show ip igmp snooping
Global Igmp Snooping Configuration
---------------------------------------------------
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Disabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Report-Suppression     :Enabled
Vlan 1
-----------
Igmp Snooping                    :Enabled
Igmp Snooping Fast-Leave         :Disabled
Igmp Snooping Report-Suppression     :Enabled
Igmp Snooping Version            :2
Igmp Snooping Robustness Variable    :2
Igmp Snooping Max-Member-Number      :2048
Igmp Snooping Unknown Multicast Behavior  :Flood
Igmp Snooping Group Access-list      :N/A
Igmp Snooping Mrouter Port           :
Igmp Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Static group

The switch can build IGMP Snooping Group when receiving IGMP report packet on Layer 2 port of specified VLAN. We also support configure static IGMP Snooping Group by specifying IGMP group, Layer 2 port and VLAN.

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Configure static group

```
Switch(config)# ip igmp snooping vlan 1 static-group 229.1.1.1 interface eth-0-2
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ip igmp snooping groups
VLAN  Interface    Group-Address    Uptime    Expires-time
1     eth-0-2      229.1.1.1        00:01:08  stopped
```

6.5.3 Application cases

N/A

6.6 Configuring MVR

6.6.1 Overview

Function Introduction

Multicast VLAN Registration (MVR) is designed for applications using wide-scale deployment of multicast traffic across an Ethernet ring-based service provider network (for example, the broadcast of multiple television channels over a service-provider network). MVR allows

a subscriber on a port to subscribe and unsubscribe to a multicast stream on the network-wide multicast VLAN. It allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs. MVR provides the ability to continuously send multicast streams in the multicast VLAN, but to isolate the streams from the subscriber VLANs for bandwidth and security reasons.

MVR assumes that subscriber ports subscribe and unsubscribe (join and leave) these multicast streams by sending out IGMP join and leave messages. These messages can originate from an IGMP version-2-compatible host with an Ethernet connection. Although MVR operates on the underlying mechanism of IGMP snooping, the two features operation affect with each other. One can be enabled or disabled with affecting the behavior of the other feature. If IGMP snooping and MVR are both enabled, MVR reacts only to join and leave messages from multicast groups configured under MVR. The switch CPU identifies the MVR IP multicast streams and their associated MAC addresses in the switch forwarding table, intercepts the IGMP messages, and modifies the forwarding table to include or remove the subscriber as a receiver of the multicast stream, and the receivers must be in a different VLAN from the source. This forwarding behavior selectively allows traffic to cross between different VLANs.

Principle Description

Terminology:

terminology	Description
MVR	Multicast Vlan Registration.
Source vlan	The vlan for receiving multicast traffic for MVR.
Source port	The port in the source vlan for sending report or leave to upstream.
Receiver port	The port not in source vlan for receiving report or leave for downstream.

6.6.2 Configuration

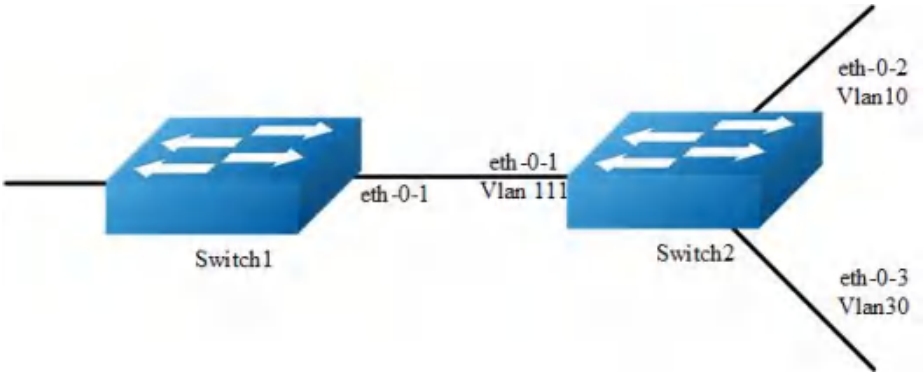


Figure 6-4 mvr

Enable IGMP&PIM-SM in the interface of eth-0-1 of Switch1.

Configure Switch2: eth-0-1 in vlan111, eth-0-2 in vlan10, and eth-0-3 vlan30.

Enable MVR in the Switch2, it is required that only one copy of multicast traffic from Switch1 is sent to Switch2, but HostA and HostC can both receive this multicast traffic.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

Configure on swich1:

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 111,10,30
Switch(config-vlan)# quit
```

step 3 Enter the interface configure mode，set the attributes and ip address, and enable pim-sm

Configure on swich1:

```
switch(config)# interface eth-0-1
switch(config-if)# no switchport
switch(config-if)# no shutdown
switch(config-if)# ip address 12.12.12.12/24
switch(config-if)# ip pim sparse-mode
switch(config-if)# exit
```

Configure on swich2:

```
Switch(config)# interface vlan 111
Switch(config-if)# exit
Switch(config)# interface vlan 10
Switch(config-if)# exit
Switch(config)# interface vlan 30
Switch(config-if)# exit
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan111
Switch(config)# interface eth-0-2
Switch(config-if)# switchport access vlan10
Switch(config)# interface eth-0-3
Switch(config-if)# switchport access vlan30
Switch(config-if)# exit
```

step 4 Enable MVR

Configure on swich2:

```
Switch(config)# no ip multicast-routing
Switch(config)# mvr
Switch(config)# mvr vlan 111
Switch(config)# mvr group 238.255.0.1 64
Switch(config)# mvr source-address 12.12.12.1
Switch(config)# interface eth-0-1
Switch(config-if)# mvr type source
Switch(config)# interface eth-0-2
Switch(config-if)# mvr type receiver vlan 10
Switch(config)# interface eth-0-3
Switch(config-if)# mvr type receiver vlan 30
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Switch1

```
Switch# show ip igmp groups
IGMP Connected Group Membership
Group Address   Interface         Uptime   Expires  Last Reporter
238.255.0.1     eth-0-1           00:01:16 00:03:49 12.12.12.1
```

```
238.255.0.2     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.3     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.4     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.5     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.6     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.7     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.8     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.9     eth-0-1          00:01:16 00:03:49 12.12.12.1
238.255.0.10    eth-0-1          00:01:16 00:03:49 12.12.12.1
……
238.255.0.64    eth-0-1          00:01:16 00:03:49 12.12.12.1
```

Switch2

```
Switch# show mvr
MVR Running: TRUE
MVR Multicast VLAN: 111
MVR Source-address: 12.12.12.1
MVR Max Multicast Groups: 1024
MVR Hw Rt Limit: 508
MVR Current Multicast Groups: 255

Switch# show mvr groups
VLAN  Interface     Group-Address     Uptime    Expires-time
10    eth-0-2       238.255.0.1       00:03:23  00:02:03
10    eth-0-2       238.255.0.2       00:02:16  00:02:03
10    eth-0-2       238.255.0.3       00:02:16  00:02:03
10    eth-0-2       238.255.0.4       00:02:16  00:02:03
10    eth-0-2       238.255.0.5       00:02:16  00:02:03
10    eth-0-2       238.255.0.6       00:02:16  00:02:04
10    eth-0-2       238.255.0.7       00:02:16  00:02:04
10    eth-0-2       238.255.0.8       00:02:16  00:02:04
10    eth-0-2       238.255.0.9       00:02:16  00:02:04
10    eth-0-2       238.255.0.10      00:02:16  00:02:04
……
10    eth-0-2       238.255.0.64      00:01:50  00:02:29
```

6.6.3 Application cases

N/A

7 PoE

7.1 PoE Overview

PoE (Power over Ethernet, remote power supply**):** It is a mechanism for supplying power to a PD (Powered Device) connected remotely through the Ethernet to realize parallel power supply and data transmission. Common PD devices include IP phones, wireless APs, IP cameras, and various new IoT terminal devices. PoE technology is widely used for its characteristics of centralized power supply, centralized management, convenient installation, and unified software and hardware standards.

PoE system involves the following devices:

PSE（Power-Sourcing Equipment）： A PSE is a PoE device that provides power to PDs and supports detection, analysis, and intelligent power management.

PD（Power Device）： A PD is a device that receives power from a PSE.PDs are classified into standard and non-standard PDs depending on whether they conform to IEEE standards：802.3af，802.3at，802.3bt.

PoE power supply： A PoE power supply provides power to a PoE system. The number of PDs connected to a PSE is limited by the power output of the PoE power supply.

7.2 Configuring Global PoE Non-Standard Compatibility

After "poe legacy" is configured in the global configuration mode, the non-standard compatible configuration on the interface that supports the PoE function will take effect. The port can detect PD devices that are not 802.3af or 802.3at standard and supply power to these devices.

SwitchA# configure terminal	Enter configuration mode
Switch(config)# poe legacy at	Enables compatibility detection for non-standard 802.3at PD devices. Default：disable.

7.3 Configuring Global PoE power management scheme

Configure the switch power management scheme in global configuration mode, including the power limit mode and power policy for PoE ports.

limit-mode: Power limit mode. The port power can be allocated based on the PD class automatically detected by the PoE port, or the port power can be allocated according to the max-power parameter configured by the user on the port. The port allocation power values corresponding to different categories are as follows:

class	power limit（W）
1	4
2	7
3	15.4
4	30
5	45
6	60
7	75
8	90

policy: Power allocation policy. There are two strategies： dynamic and static. When configured as dynamic, the remaining available power is calculated from the total PoE power allocable power minus the actual power allocated by all ports. Then decide whether to supply power to the newly connected PD according to the remaining available power and the current power consumption of the newly connected PD; when the configuration is static, the system calculates the maximum power supply power of each port in use according to limit-mode and reserves power for it . Then, it is determined whether to supply power to the newly connected PD according to the remaining available power and the current power consumption of the newly connected PD.

Autoclass: When configuring the static power allocation policy, you can choose whether to enable the autoclass function. If autoclass is enabled and the PD also supports the autoclass function, the power can be reserved for the port more reasonably through the negotiation between the PSE and the PD, so as to avoid the waste of reserving too much power for low-power loads, and rational use of resources can be used in more supply power to the PD on the interface.

SwitchA# configure terminal	Enter configuration mode
Switch(config)# poe power-management limit-mode class	Configure the power limit mode of each interface to use the class-based power limit mode.
Switch(config)# poe power-management policy static autoclass enable	Configure the power management policy as static and enable the autoclass function. If the PD supports this function, it is more reasonable to reserve power.

For example, the power limit mode in the three PoE configurations is also "based on port's configured max power". However, when the "power policy" is dynamic, static, and static, the auto class function is enabled at the same time. When comparing the "granted power", "power limit", and "available power" in the "PoE status" and "PoE summary" in the above three configurations, you can more intuitively see the difference in power distribution results.The following are examples of configuration and status display:

```
Switch# configure terminal
Switch(config)# poe power-management limit-mode class
Switch(config)# end
Switch# show poe status
Codes : IF - interface  CH - channel   ST - enable status
    MD - mode     CL - class     PR - priority
    V - voltage(V)       mA - current(mA)      TP - temperature('C)
    GT - granted power(W)  LM - power limit(W)   CS - consumption(W)
    AP - available power(W)
    DIS - disable  EN - enable
    CR - critical  HI - high     LO - low
    OP - operation status of poe
+---------+----+---+------+---+---+---+---+---+---+------+------+-----+------
IF       ST  OP  MD    CL  PR  V   mA  TP  GT    LM    CS    AP
eth-0-1   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-2   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-3   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-4   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-5   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-6   EN  OFF  POE_BT --  LO  0.0  0   37  0.000  --    0.000  --
eth-0-7   EN  OFF  POE_BT --  LO  0.0  0   37  0.000  --    0.000  --
eth-0-8   EN  OFF  POE_BT --  LO  0.0  0   37  0.000  --    0.000  --
eth-0-9   EN  ON   POE_BT 8   LO  54.6 679 37  37.254  41.000  37.055  3.945
eth-0-10  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-11  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-12  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-13  EN  ON   POE_BT 6   LO  54.7 34  39  1.639   60.000  1.858   58.142
eth-0-14  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
```

```
eth-0-15  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-16  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-17  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-18  EN  OFF  POE_BT -- LO 0.0 0  43 0.000  --    0.000  --
eth-0-19  EN  ON   POE_BT 4  LO 54.5 33 43 1.797  30.000 1.797  28.203
eth-0-20  EN  OFF  POE_BT -- LO 0.0 0  43 0.000  --    0.000  --
eth-0-21  EN  OFF  POE_BT -- LO 0.0 0  43 0.000  --    0.000  --
eth-0-22  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-23  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-24  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --

Switch# show poe summary
-------------------------------------------------
Power Source Equipment NO: 1
Operation status: enable
Channel Num for power: 48
PSE Firmware Version: Ve0.17.36.01
PSE SRAM Patch Version: 0x0
PSE PSU Information
    PSU1: Absent
    PSU2: 400 W Present
PSE Power Policy: Dynamic
PSE Power Limit Mode: Class Based Limit
PSE Legacy PD Enable: Disable
Power Limit: 400000 mW
Power Available: 359094 mW
Power Actual Consump: 40483 mW
Power Allocated: 40906 mW
Power Reserved: 30000 mW
Power Threshold: 90%
PSE Reserved Power Alert: -
PSE Threshold Power Alert: -
PSE Device Temperature
    Device 0: 39 degC
    Device 1: 39 degC
    Device 2: 41 degC
    Device 3: 39 degC
    Device 4: 40 degC
    Device 5: 42 degC
PSE Device Faults
    SRAM Fault: -
    TSD Fault: -
    VPWR Fault: -
    VDD Fault: -
Switch#
Switch# configure terminal
Switch(config)# poe power-management policy dynamic
Switch(config)# end
Switch# show poe status
Codes : IF - interface  CH - channel   ST - enable status
     MD - mode     CL - class     PR - priority
     V - voltage(V)       mA - current(mA)      TP - temperature('C)
     GT - granted power(W)  LM - power limit(W)    CS - consumption(W)
     AP - available power(W)
     DIS - disable  EN - enable
```

```
    CR - critical   HI - high      LO - low
    OP - operation status of poe
+---------+----+---+-----+----+---+---+---+---+-----+------+------+-----
IF       ST  OP  MD    CL  PR  V   mA  TP  GT    LM     CS     AP
eth-0-1   EN  OFF  POE_BT -- LO  0.0  0   36  0.000  --     0.000  --
eth-0-2   EN  OFF  POE_BT -- LO  0.0  0   36  0.000  --     0.000  --
eth-0-3   EN  OFF  POE_BT -- LO  0.0  0   36  0.000  --     0.000  --
eth-0-4   EN  OFF  POE_BT -- LO  0.0  0   36  0.000  --     0.000  --
eth-0-5   EN  OFF  POE_BT -- LO  0.0  0   36  0.000  --     0.000  --
eth-0-6   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-7   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-8   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-9   EN  ON   POE_BT 8   LO  54.9 679 39  37.099  41.000  37.267  3.733
eth-0-10  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-11  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-12  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-13  EN  ON   POE_BT 6   LO  54.7 34  39  1.847   90.000  1.858   88.142
eth-0-14  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-15  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-16  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-17  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --     0.000  --
eth-0-18  EN  OFF  POE_BT -- LO  0.0  0   42  0.000  --     0.000  --
eth-0-19  EN  ON   POE_BT 4   LO  54.5 34  42  1.797   75.000  1.852   73.148
eth-0-20  EN  OFF  POE_BT -- LO  0.0  0   42  0.000  --     0.000  --
eth-0-21  EN  OFF  POE_BT -- LO  0.0  0   42  0.000  --     0.000  --
eth-0-22  EN  OFF  POE_BT -- LO  0.0  0   40  0.000  --     0.000  --
eth-0-23  EN  OFF  POE_BT -- LO  0.0  0   40  0.000  --     0.000  --
eth-0-24  EN  OFF  POE_BT -- LO  0.0  0   40  0.000  --     0.000  --

Switch# show poe summary
-----------------------------------------------------
Power Source Equipment NO: 1
Operation status: enable
Channel Num for power: 48
PSE Firmware Version: Ve0.17.36.01
PSE SRAM Patch Version: 0x0
PSE PSU Information
    PSU1: Absent
    PSU2: 400 W Present
PSE Power Policy: Dynamic
PSE Power Limit Mode: Port Based Limit
PSE Legacy PD Enable: Disable
Power Limit: 400000 mW
Power Available: 359186 mW
Power Actual Consump: 40977 mW
Power Allocated: 40814 mW
Power Reserved: 30000 mW
Power Threshold: 90%
PSE Reserved Power Alert: -
PSE Threshold Power Alert: -
PSE Device Temperature
    Device 0: 36 degC
    Device 1: 36 degC
    Device 2: 39 degC
    Device 3: 39 degC
```

```
    Device 4: 39 degC
    Device 5: 42 degC
PSE Device Faults
    SRAM Fault: -
    TSD Fault: -
    VPWR Fault: -
    VDD Fault: -
Switch#
Switch# configure terminal
Switch(config)# poe power-management limit-mode port
Switch(config)# poe power-management policy static autoclass enable
Switch(config)# end
Switch# show poe status
Codes : IF - interface  CH - channel   ST - enable status
      MD - mode      CL - class     PR - priority
      V - voltage(V)        mA - current(mA)       TP - temperature('C)
      GT - granted power(W)   LM - power limit(W)    CS - consumption(W)
      AP - available power(W)
      DIS - disable   EN - enable
      CR - critical   HI - high      LO - low
      OP - operation status of poe
+---------+----+---+------+---+---+----+--+--+------+-------+-----+---
IF        ST  OP  MD    CL  PR  V   mA  TP  GT    LM    CS    AP
eth-0-1   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-2   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-3   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-4   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-5   EN  OFF  POE_BT --  LO  0.0  0   36  0.000  --    0.000  --
eth-0-6   EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-7   EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-8   EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-9   EN  ON   POE_BT 8   LO  54.6 679 39  38.000 41.000 37.053 3.947
eth-0-10  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-11  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-12  EN  OFF  POE_BT --  LO  0.0  0   39  0.000  --    0.000  --
eth-0-13  EN  ON   POE_BT 6   LO  54.4 34  39  90.000 90.000 1.847  88.153
eth-0-14  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-15  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-16  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-17  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-18  EN  OFF  POE_BT --  LO  0.0  0   42  0.000  --    0.000  --
eth-0-19  EN  ON   POE_BT 4   LO  54.5 33  42  75.000 75.000 1.798  73.202
eth-0-20  EN  OFF  POE_BT --  LO  0.0  0   42  0.000  --    0.000  --
eth-0-21  EN  OFF  POE_BT --  LO  0.0  0   42  0.000  --    0.000  --
eth-0-22  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-23  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --
eth-0-24  EN  OFF  POE_BT --  LO  0.0  0   41  0.000  --    0.000  --

Switch# show poe summary
-----------------------------------------------------
Power Source Equipment NO: 1
Operation status: enable
Channel Num for power: 48
PSE Firmware Version: Ve0.17.36.01
PSE SRAM Patch Version: 0x0
```

```
PSE PSU Information
    PSU1: Absent
    PSU2: 400 W Present
PSE Power Policy: Static w/ Autoclass
PSE Power Limit Mode: Port Based Limit
PSE Legacy PD Enable: Disable
Power Limit: 400000 mW
Power Available: 197000 mW
Power Actual Consump: 40907 mW
Power Allocated: 203000 mW
Power Reserved: 30000 mW
Power Threshold: 90%
PSE Reserved Power Alert: -
PSE Threshold Power Alert: -
PSE Device Temperature
    Device 0: 35 degC
    Device 1: 35 degC
    Device 2: 39 degC
    Device 3: 39 degC
    Device 4: 40 degC
    Device 5: 43 degC
PSE Device Faults
    SRAM Fault: -
    TSD Fault: -
    VPWR Fault: -
    VDD Fault: -
Switch# configure terminal
Switch(config)# poe power-management policy static autoclass disable
Switch(config)# end
Switch# show poe status
Codes : IF - interface  CH - channel   ST - enable status
     MD - mode     CL - class     PR - priority
     V - voltage(V)       mA - current(mA)      TP - temperature('C)
     GT - granted power(W)  LM - power limit(W)   CS - consumption(W)
     AP - available power(W)
     DIS - disable   EN - enable
     CR - critical   HI - high     LO - low
     OP - operation status of poe
+--------+----+----+------+---+---+----+---+---+-----+------+------+----
IF       ST  OP  MD    CL  PR V    mA  TP  GT    LM    CS    AP
eth-0-1   EN  OFF  POE_BT -- LO  0.0  0   34  0.000  --    0.000  --
eth-0-2   EN  OFF  POE_BT -- LO  0.0  0   34  0.000  --    0.000  --
eth-0-3   EN  OFF  POE_BT -- LO  0.0  0   34  0.000  --    0.000  --
eth-0-4   EN  OFF  POE_BT -- LO  0.0  0   34  0.000  --    0.000  --
eth-0-5   EN  OFF  POE_BT -- LO  0.0  0   34  0.000  --    0.000  --
eth-0-6   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-7   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-8   EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-9   EN  ON   POE_BT 8  LO  54.9 679 39  90.000 41.000 37.259 3.741
eth-0-10  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-11  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-12  EN  OFF  POE_BT -- LO  0.0  0   39  0.000  --    0.000  --
eth-0-13  EN  ON   POE_BT 6  LO  54.4 34  39  90.000 90.000 1.847  88.153
eth-0-14  EN  OFF  POE_BT -- LO  0.0  0   40  0.000  --    0.000  --
eth-0-15  EN  OFF  POE_BT -- LO  0.0  0   40  0.000  --    0.000  --
```

```
eth-0-16  EN  OFF  POE_BT -- LO 0.0 0  40 0.000  --    0.000  --
eth-0-17  EN  OFF  POE_BT -- LO 0.0 0  40 0.000  --    0.000  --
eth-0-18  EN  OFF  POE_BT -- LO 0.0 0  42 0.000  --    0.000  --
eth-0-19  EN  ON   POE_BT 4  LO 54.5 33 42 75.000 75.000 1.797  73.203
eth-0-20  EN  OFF  POE_BT -- LO 0.0 0  42 0.000  --    0.000  --
eth-0-21  EN  OFF  POE_BT -- LO 0.0 0  42 0.000  --    0.000  --
eth-0-22  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-23  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --
eth-0-24  EN  OFF  POE_BT -- LO 0.0 0  41 0.000  --    0.000  --

Switch# show poe summary
-------------------------------------------------
Power Source Equipment NO: 1
Operation status: enable
Channel Num for power: 48
PSE Firmware Version: Ve0.17.36.01
PSE SRAM Patch Version: 0x0
PSE PSU Information
    PSU1: Absent
    PSU2: 400 W Present
PSE Power Policy: Static w/o Autoclass
PSE Power Limit Mode: Port Based Limit
PSE Legacy PD Enable: Disable
Power Limit: 400000 mW
Power Available: 145000 mW
Power Actual Consump: 40903 mW
Power Allocated: 255000 mW
Power Reserved: 30000 mW
Power Threshold: 90%
PSE Reserved Power Alert: -
PSE Threshold Power Alert: -
PSE Device Temperature
    Device 0: 36 degC
    Device 1: 36 degC
    Device 2: 39 degC
    Device 3: 39 degC
    Device 4: 41 degC
    Device 5: 42 degC
PSE Device Faults
    SRAM Fault: -
    TSD Fault: -
    VPWR Fault: -
    VDD Fault: -
Switch#
```

Usually, we configure the limit-mode as class and the policy as dynamic to reasonably allocate the system power and ensure that as many PD devices are connected as possible. However, if the power of the PD device fluctuates greatly during use, the power supply policy needs to be configured according to the actual situation to avoid unpredictable changes between power supply and power failure of some devices due to insufficient power.

7.4 Configuring power-reserved

The power-reserved is to reserve a part of the allocable power of the PoE power supply as the PoE power supply protection of the whole machine. When the remaining power of the system is less than the reserved protection power, it will no longer supply power to the newly connected PD devices. At the same time, the PoE state light of the front panel is on orange.

SwitchA# configure terminal	Enter configuration mode
Switch(config)# poe power-reserved 60000	Configure reserved-power of poe to 60000mW.default：30000.

7.5 Configure threshold-power-percentage

You can configure the alarm threshold for real-time PoE power consumption. When the total real-time power consumption exceeds or falls below the alarm threshold for the first time, the system will prompt the user. This threshold is also used to determine the state of the PoE LED on the front panel of the switch. If the power consumption does not exceed the alarm threshold, the PoE LED turns green; if the power consumption exceeds the alarm threshold, it turns orange.

SwitchA# configure terminal	Enter configuration mode
Switch(config)# poe power-threshold 85	Configure threshold-power-percentage of poe to 85%.Default：90%.

7.6 Enable port PoE feature

For a ports that supports the PoE function, you can enable or disable the PoE function of the ports in the port configuration mode. You can also attach the time-range to formulate a power supply task for the PoE port(s) to supply or not supply power in a specific time period.

SwitchA# configure terminal Switch(config)# interface eth-0-9	Enter configuration mode
Switch(config-if)# poe disable Switch(config-if)# poe enable time-range plan1	Configure this port to supply power during the time period defined by “plan1”, and not supply power at other times. Default:enable。

7.7 Configure max-power supply

In interface configuration mode, the max-power supply of the port can be configured. The value range is 1-120000mW, and the default value is 90000mW.

SwitchA# configure terminal Switch(config)# interface eth-0-9	Enter configuration mode
Switch(config-if)# poe max-power 75000	Configure the maximum power supply of this port to 75000mW. Default:90000mW。

7.8 Configuring the Power Supply Priority

Reasonable configuration of the power supply priority of the PoE ports, combined with the configuration of the global power-management scheme in the previous section, can guarantee the power supply to more important PD to a limited extent when the external power supply of the switch is close to or exceeds the full load.

The PoE power supply priority of a port is critical, high, and low from high to low. The default priority of all POE ports is low. When a new PD device is connected to a port with a higher priority when the external power supply is fully loaded, the PSE will power off the PD

connected to the port with a lower priority, and then supply power to the newly connected PD; If all ports have the same priority, the PD connected to the port with the smaller port number is given priority to supply power.

SwitchA# configure terminal Switch(config)# interface eth-0-9	Enter configuration mode
Switch(config-if)# poe priority critical	Configure the power supply priority of the port to be the highest:critical.

7.9 Configure Port Compatibility

In the interface configuration mode, configure poe legacy to enable the compatibility detection function, which can detect and supply power to these non-standard PD devices. The type of compatible non-standard PD device is specified by the poe legacy command configuration in global configuration mode. When using it, pay attention to the combination of global and interface configuration.

SwitchA# configure terminal Switch(config)# poe legacy at	Enter configuration mode Enabled to check compatibility of non-standard 802.3at PDs.
Switch(config)# interface eth-0-9 Switch(config-if)# poe legacy	The interface is enabled to check compatibility of non-standard PDs.

7.10 Trigger an autoclass measurement

In the interface configuration mode, the autoclass can be manually triggered to measure the power of the PD, and then a more reasonable power limit parameter can be set for the port according to the measured result, so as to allocate the power of the whole machine more reasonably.

SwitchA# configure terminal Switch(config)# interface eth-0-9	Enter interface configuration mode
Switch(config-if)# poe autoclass ch1 ch2 adjustment ch1 ch2	Trigger port eth-0-9 to restart autoclass power measurement manually and adjust the max-power value on channel 1 and 2:

7.11 Example for Configuring PoE

Networking Requirements

In an office network, port 1 of the PoE switch is connected to the wireless AP in the company lobby to provide Wi-Fi services for guests, and needs to maintain power supply during working hours; ports 5 and 6 are connected to IP phones of business departments, and require the highest priority power supply; Ports 7 and 8 are connected to wireless APs to provide WiFi coverage for office areas, and require the second highest priority power supply; port 24 is connected to a IP camera, and also requires the highest priority power supply.The power distribution is dynamically adjusted according to the actual power output.



Procedure

#Define a time-range entry for working hours:

```
Switch# configure terminal
Switch(config)# time-range working
Switch(config-tm-range)# periodic 8:30 daily to 18:30
Switch(config-tm-range)# exit
```

#Configuring a global power management scheme:

```
Switch(config)# poe power-management policy dynamic
Switch(config)# poe power-management limit-mode class
```

#Configuring the PoE Power Supply Policy for the ports:

```
Switch(config)# interface eth-0-1
Switch(config-if)# poe disable
Switch(config-if)# poe enable time-range working
Switch(config-if)# exit
Switch(config)# interface range eth-0-5 - 6
Switch(config-if-range)# poe priority critical
Switch(config-if-range)# poe enable
Switch(config-if-range)# exit
Switch(config)# interface eth-0-24
Switch(config-if)# poe priority critical
Switch(config-if)# poe enable
Switch(config-if)# exit
Switch(config)# interface range eth-0-7 - 8
Switch(config-if-range)# poe priority high
Switch(config-if-range)# poe enable
Switch(config-if-range)# exit
Switch(config)#exit
Switch#
```

#Save the configuration:

```
Switch# write
Switch#
```

8 Security Configuration Guide

8.1 Configuring Port Security

8.1.1 Overview

Function Introduction

Port security feature is used to limit the number of "secure" MAC addresses learnt on a particular interface. The interface will forward packets only with source MAC addresses that match these secure addresses. The secure MAC addresses can be created manually, or learnt automatically. After the number of secure MAC addresses reaches the limit for the number of secure MAC addresses, new MAC address can't be learnt or configured on the interface. if the interface then receives a packet with a source MAC address that is different with any of the secure addresses, it is considered as a security violation and should be discarded.

Port security feature also binds a MAC to a port so that the port does not forward packets with source addresses that are outside of defined addresses. If a MAC addresses configured or learnt on a secure port attempts to access another port, this is also considered as a security violation.

Three types of secure MAC addresses are supported:

- Secure dynamic MAC address: By using the command "switchport port security" to enable port security on the interface but not the Sticky MAC function, the number of MAC addresses learned by the interface will be limited.
- Secure static MAC address: Manually configure the static MAC address through the command line "switchport port security mac address".
- Sticky MAC address: The MAC address converted to when the port is secured and the Sticky MAC function is also enabled.

If a security violation occurs, the packets to be forwarded will be dropped. User can configure the action by command "switchport port-security violation". There are three actions can be chosen：

- errdisable: discard the packet and set the port to errdisable status. Please reference to Ethernet configuration guide, chapter errdisable.
- protect: discard only.
- restrict： discard and record the event in log.

Principle Description

N/A

8.1.2 Configuration

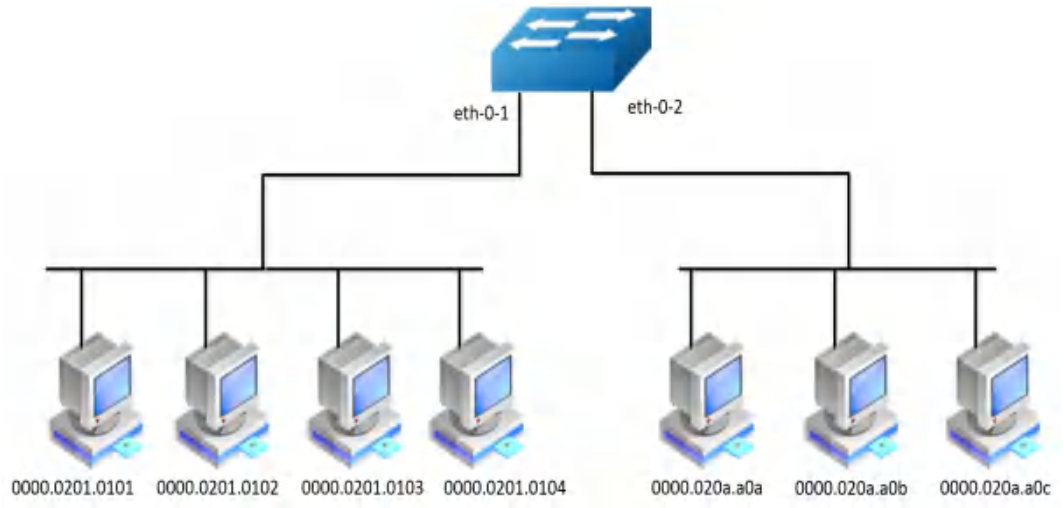


Figure 8-1 Port Security

According to the topology above, only receive three Mac entries and discard source mac 0000.000B.000B after the following configuration:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode，set the attributes, and enable port-security

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 2
Switch(config-if)# switchport port-security mac-address 0000.0201.0101 vlan 1
Switch(config-if)# switchport port-security violation restrict
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 2
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show port-security
Port Security Work Mode: Hardware

Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolationMode
          (Count)    (DynamicCount)
-----------------------------------------------------------------
 eth-0-1        2          2     restrict
 eth-0-2        2          0     protect
Switch#
Switch# show mac address-table
       Mac Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   Mac Address     Type      Ports
----   -----------     --------  -----
1      0000.0201.0103   dynamic(*)  eth-0-1
1      0000.020a.0a0b   static(*)   eth-0-2
1      0000.0201.0102   dynamic(*)  eth-0-1
1      0000.020a.0a0a   static(*)   eth-0-2
1      0000.0201.0101   static(*)   eth-0-1
11     00e0.4c02.172f   dynamic    eth-0-31
Switch#
Switch#
show port-security address-table
       Secure MAC address table
-----------------------------------------------------------
Vlan   Mac Address     Type            Ports
```

```
----   -----------     -------          -------
1      0000.0201.0102   SecureLearned      eth-0-1
1      0000.0201.0101   SecureConfigured   eth-0-1
1      0000.020a.0a0b   StickyConfigured   eth-0-2
1      0000.020a.0a0a   StickyConfigured   eth-0-2
1      0000.0201.0103   SecureLearned      eth-0-1
Switch# show running-config interface eth-0-2
Building configuration...
!
interface eth-0-2
 switchport port-security mac-address sticky
 switchport port-security
 switchport port-security maximum 2
 switchport port-security mac-address sticky 0000.020a.0a0a vlan 1
 switchport port-security mac-address sticky 0000.020a.0a0b vlan 1
!
Switch#
```

8.1.3 Application cases

N/A

8.2 Configuring Vlan Security

8.2.1 Overview

Function Introduction

Vlan security feature is used to limit the total number of MAC addresses learnt in a particular vlan. The MAC addresses can be added manually, or learnt automatically. After the device reaches the limit for the number of MAC addresses on the vlan, if the vlan receives a packet with an unknown source MAC address, the configured action will take effect.

Two types of MAC addresses are supported:

- Static MAC addresses: These are manually configured by users.
- Dynamic MAC addresses: These are dynamically learnt.

User can set the action for unknown source MAC packets after the MAC address table count exceed max by using command line "vlan *X* mac-limit action". Three types of actions are supported:

- Discard: Packet with an unknown source MAC address from the vlan will be discarded and its source MAC address will not be learnt.
- Warn: Packet with an unknown source MAC address from the vlan will be discarded, its source MAC address will not be learnt, but warning log will be printed in syslog.
- Forward: Packets from the vlan will be forwarded without MAC learning or warning log.

MAC address learning feature can be enabled or disabled per-VLAN.

Principle Description

N/A

8.2.2 Configuration

Configuring vlan mac-limit

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, set the the maximum of MAC addresses and the action at exceeding

```
Switch# configure terminal
Switch(config)# vlan database
Switch(config)# vlan 2
Switch(config-vlan)# vlan 2 mac-limit maximum 100
Switch(config-vlan)# vlan 2 mac-limit action discard
Switch(config-vlan)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show vlan-security
 Vlan  learning-en  max-mac-count  cur-mac-count  action
 -------------------------------------------------------
 2    Enable     100          0           Discard
```

Configuring vlan mac learning

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, set the mac learning states

```
Switch(config)# vlan database
Switch(config)# vlan 2
Switch(config-vlan)# vlan 2 mac learning disable
Switch(config-vlan)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show vlan-security
 Vlan  learning-en  max-mac-count  cur-mac-count  action
 -------------------------------------------------------
 2    Disable    100          0           Discard
```

8.2.3 Application cases

N/A

8.3 Configuring Time-Range

8.3.1 Overview

Function Introduction

A time range is created that defines specific absolute times or periodic times of the day and week in order to implement time-based function, such as ACLs. The time range is identified by a name and then referenced by a function， which by itself has no relevance. Therefore, the time restriction is imposed on the function itself. The time range relies on the system clock.

Principle Description

N/A

8.3.2 Configuration

Create an absolute time range

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a time-range and set absolute time

```
Switch(config)# time-range test-absolute
Switch(config-tm-range)# absolute start 1:1:2 jan 1 2012 end 1:1:3 jan 7 2012
Switch(config-tm-range)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
DUT1# show time-range
time-range test-absolute
  absolute start 01:01:02 Jan 01 2012 end 01:01:03 Jan 07 2012
```

Create a periodic time range

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a time-range and set periodic time

```
Switch(config)# time-range test-periodic
Switch(config-tm-range)# periodic 1:1 mon to 1:1 wed
Switch(config-tm-range)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
DUT1# show time-range
time-range test-periodic
  periodic 01:01 Mon to 01:01 Wed
```

8.3.3 Application cases

N/A

8.4 Configuring ACL

8.4.1 Overview

Function Introduction

Access control lists (ACLs) classify traffic with the same characteristics. The ACL can have multiple access control entries (ACEs), which are commands that match fields against the contents of the packet. ACLs can filter packets received on interface by many fields such as ip address, mac address and deny or permit the packets.

Network plays an important role in today's life; people are paying more attention on network safety as well as relying on it. ACL can realize to protect the network security in lots of situations, for example:

- Limit the access of network. To make sure the network is safe, it can limit users to access some services by configuring security ACL (if only access WWW and email service, other services like Telnet will be prohibited); or allow accessing within some assigned time; or only allow some of the hosts to access the network.
- Internet virus attack the internal network of enterprises, it can prevent virus' data getting into internal network by security ACL.
- The server resource of an enterprise is more earlier to be divulged; according to security ACL, it can control the access to it.
- When the administrators want to supervise the network access of one user, he can monitor the network traffic to the supervised equipments by strategy ACL.
- Administrators need to put some data streams into firewall tools to check the security; they can apply strategy ACL to redirect the data stream to firewall tools to specified port, and then conduct checking the safety.

Principle Description

The following terms and concepts are used to describe ACL：

- **Access control entry (ACE)**: Each ACE includes an action element (permit or deny) and a series of filter element based on criteria such as source address, destination address, protocol, and protocol-specific parameters.
- **MAC ACL**: MAC ACL can filter packet by mac-sa and mac-da, and the mac-address can be masked, or configured as host id, or configured as any to filter all MAC addresses. MAC ACL can also filter other L2 fields such as COS, VLAN-ID, INNER-COS, INNER-VLAN-ID, L2 type, L3 type.
- **IPv4 ACL**: IPv4 ACL can filter packet by ip-sa and ip-da, and ip-address can be masked, or configured as host id, or configured as any to filter all IPv4 address. IPv4 ACL can also filter other L3 fields such as DSCP, L4 protocol and L4 fields such as TCP port, UDP port, and so on.
- **Time Range**: Time range can define a period of time only between which the ACE can be valid if the ACE is associated to the time range.

Rule description

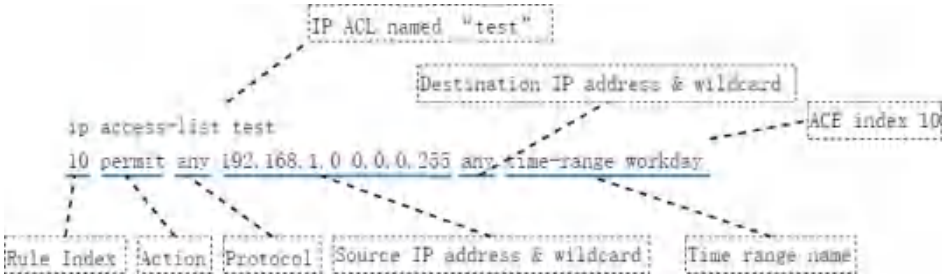


Figure 8-2 ACL Rule description

- [ACL name] used to distinguish different ACL
- [ACE] full name is "Access control entry". One ACL includes one or more ACEs.
- [Rule ID] The Rule ID has two roles. The first role of the rule ID is to mark the ACEs. Users can delete the ACE with specified rule ID. The second role of the rule ID is to specify the priority of each ACEs. The valid range of the rule ID is 1-131071, the ACE's rule ID with smaller number has higher priority and match first.
- [action] includes permit and deny, indicates allow or deny matched frames to pass through.
- [field] ACL provides extremely abundant matching fields. ACL also provides other matching fields such as Layer 2 Ethernet frame header information (for example source MAC, destination MAC, ethertype, VLAN), DSCP, TTL, fragment/non-fragment , TCP/UDP port number, TCP Flags and so on.

Matching Rules

There are two types of matching mode for ACL, one is match-any mode (default) and another is match-all mode. Users can choose either match-any mode or match-all mode when create the traffic classification.

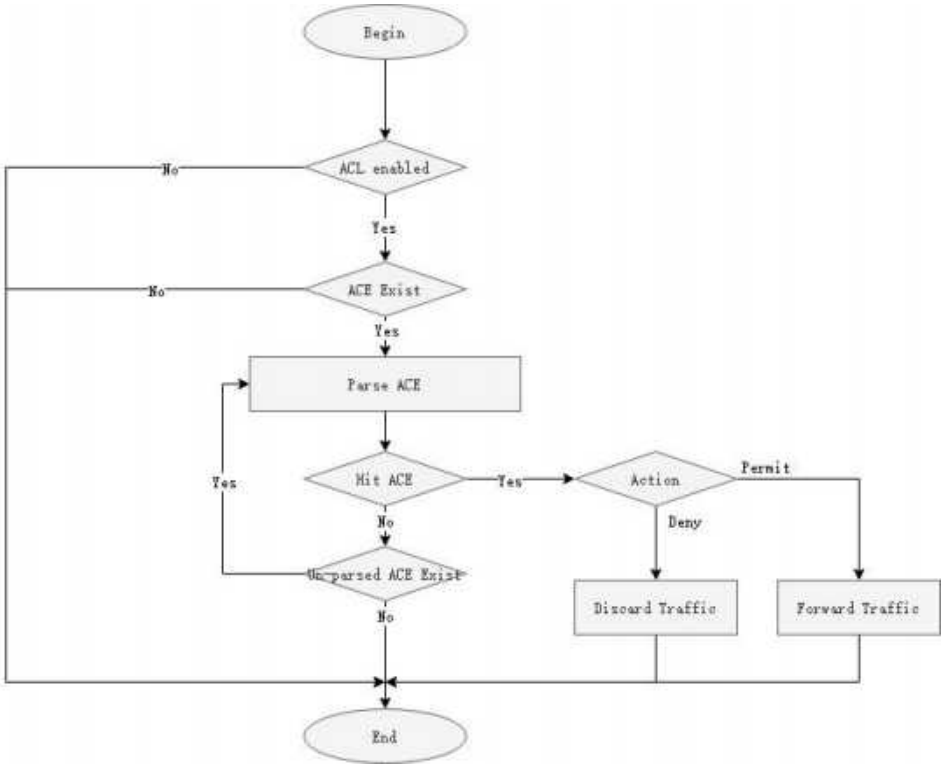


Figure 8-3 Flow chart for ACL match-any mode

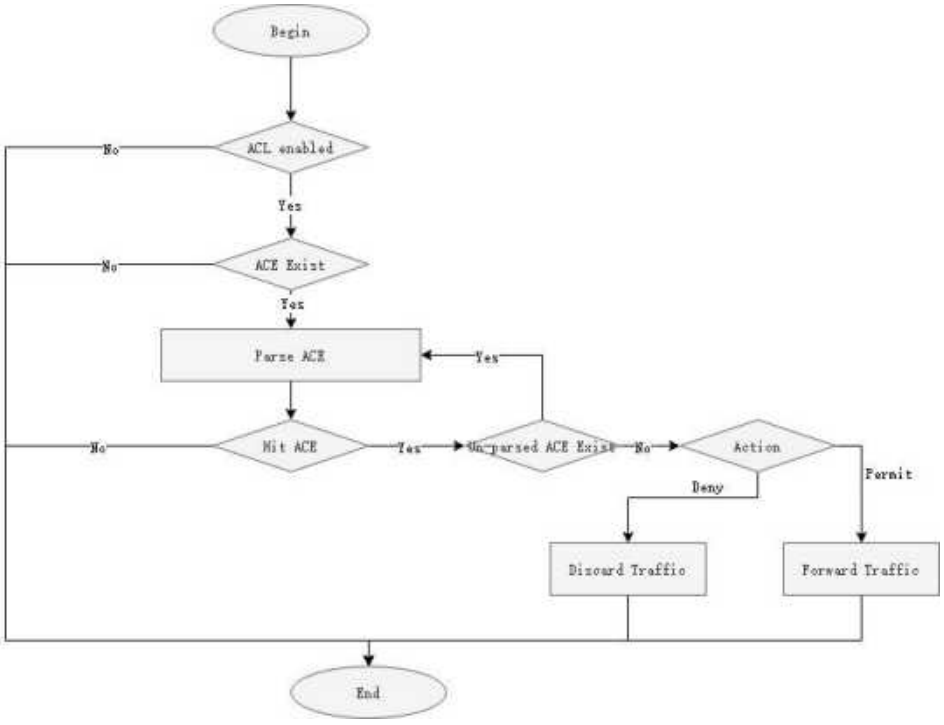


Figure 8-4 Flow chart for ACL match-all mode

NOTE

- To frames that are not matched ACL, the default action will allow the messages passing through.

- The frames that are exiting from monitor or redirect from the same inbound of device cannot match the ACL at outbound
- In match-all mode, an action of ACE (permit or deny) in ACL must be identical otherwise it cannot be configured.
- Since match-all mode needs frames that match all ACEs as matched, so ACEs under the same ACL does not allow conflict fields; otherwise it cannot be configured.

ACL fields

ACL supports very abundant matching fields, the followings are introducing a few matching fields that use a lot:

Source/Destination IP address and its wildcard-mask

There are three formats of IP address matching components in ACL:

- Any: match all source or destination IP address
- host ...: only match a single IP address, it does not need to configure wildcard-mask
- A.B.C.D ...: match the IP address within a range; it needs to configure wildcard-mask.

Wildcard-mask of the IP address and reverse subnet mask are similar, it is also a 32 Bit integer array which is used for showing which digits will be checked in IP address. In each Bit, "o" represents checking the corresponding Bit; "1" represents does not check corresponding Bit. Compared with subnet mask of IP address, "0" and "1" in subnet mask must be sequential; which is different than wildcard-mask.

Example: Hope that all the IP address from 192.168.1.0/24 can pass, it can be configure as the following:

permit any 192.168.1.0 0.0.0.255 any

Table 8-1 Parameter description

Project	Decimal with equal value	Binary with equal value
Reference address	192.168.1.0	11000000.10101000.00000001.00000000
Wildcard-mask	0.0.0.255	00000000.00000000.00000000.11111111
Make sure the range of address	IP address between 192.164.1.0 to 192.168.1.255	11000000.10101000.00000001.XXXXXXXX X can be "0" or "1"

Source/Destination MAC address and Wildcard-mask

Source/Destination MAC in ACL is similar to IP address, the difference is MAC address uses hexadecimal and has 48 bits, IP address uses decimal and has 32 bits. For methods of application, please read the instruction "Source/Destination IP address and its wildcard-mask".

IP Protocol Type

There are three kinds of format about IP protocol types in ACL:

- Any: match all IP protocol types.
- IP protocol number: the value of configuring IP protocol type that user defines; the range should be 0 to 255.
- Gre|icmp|igml|nvgre|tcp|udp: these are some frequently used protocols that the system encapsulated.

Frequent protocol types including: ICMP (protocol 1), TCP (protocol 6), UDP (protocol 17), GRE (protocol 47), IGMP (protocol 2), IPinIP(protocol 4), OSPF(protocol 89)

TCP/UDP port number

Source port number form： **src-port** { **eq** port | **gt** port | **lt** port | **range** port-start port-end}

Destination port number form： **dst-port** { **eq** port | **gt** port | **lt** port | **range** port-start port-end}

- **eq** port : assign equal to the port number at source or destination.
- **gt** port : assign greater than the port number at source or destination
- **lt** port : assign smaller than the port number at source or destination

- **range** port-start port-end：assign the range of port number at source or destination . Port-start is the initial value of port range，Port-end is the ending value of port range.

Table 8-2 TCP ports and its corresponding protocols

Port number	Protocol	Explanation
7	Echo	Echo service
9	Discard	Empty service that use for connect testing
13	Daytime	Send date and time to host
19	Character generator	Service that generate characters; sending unlimited character streams
20	FTP data connections	FTP data port
21	File Transfer Protocol(FTP)	File Transport Protocol(FTP) port
23	Telnet	Telnet service
25	Simple Mail Transport Protocol (SMTP)	Simple mail transport protocol
37	Time	Time protocol
43	Nicname（WHOIS）	Menu service
49	TAC Access Control System (TACACS)	Access system that used for test and access which based on TCP/IP (TACACS access host protocol)
53	Domain Name Service (DNS)	Domain name service
70	Gopher	Information retrieval protocol (internet documents searching and retrieving)
79	Finger	Used for Finger service that users' contact information, searching online users and other information on remote host
80	World Wide Web (HTTP)	Used for hypertext transport protocol (HTTP) from world wide web (WWW) service. It is used for surfing webpage.
101	NIC hostname server	Host's name service on NIC machine.
109	Post Office Protocol v2	Post protocol version 2
110	Post Office Protocol v3	Post protocol version 3
111	Sun Remote Procedure Call (RPC)	Remote procedure call protocol from SUN Inc. It is used for executing remote commands, and is used by network file system.
119	Network News Transport Protocol (NNTP)	Network news transport protocol, bear USENET networking
179	Border Gateway Protocol (BGP)	Border Gateway Protocol
194	Internet Relay Chat (IRC)	Internet Relay Chat（multicast chatting protocol）
512	Exec (rsh)	Used for verifying the progress of remote execution.
513	Login (rlogin)	Remote login

514	Remote commands	Remote commands, does not need to login remote shell and remote copy
515	Printer service	Printer
517	Talk	remote communication service for clients
540	Unix-to-Unix Copy Program	Copy service from Unix to Unix
543	Kerberos login	Kerberos version 5 for remote login
544	Kerberos shell	Kerberos version 5 for remote shell

Table 8-3 UDP ports and its corresponding protocols

Port number	Protocol	Explanation
7	Echo	Echo service
9	Discard	Empty service that use for connect testing
37	Time	Time protocol
42	Host Name Server	host name service
53	Domain Name Service (DNS)	Domain name service
65	TACACS-Database Service	TACACS database service
67	Bootstrap Protocol Server	Bootstrap protocol server side，DHCP server used
68	Bootstrap Protocol Client	Bootstrap protocol client side，DHCP client used
69	Trivial File Transfer Protocol (TFTP)	Trivial file transfer protocol
90	DNSIX Security Attribute Token Map	DNSIX security attribute symbol map
111	SUN Remote Procedure Call (SUN RPC)	SUN Inc remove procedure call protocol，it used for doing remote commands, and is used by network file system
123	Network Time Protocol (NTP)	Network time protocol, virus will use it
137	NETBIOS Name Service	NETBIOS name service
138	NETBIOS Datagram Service	NETBIOS datagram service
139	NETBIOS Session Service	NETBIOS communication session service
161	SNMP	Simple network management protocol
162	SNMPTRAP	Simple Network Management Protocol TRAP
177	X Display Manager Control Protocol (XDMCP)	X will display the manager control protocol
434	MobileIP-Agent	Mobile IP agent
435	MobileIP-MN	Mobile IP managements
512	Mail notify	Mail notification, which can use to notify users that there is an email reached
513	Who	Login user list

514	Syslog	UNIX system data server
517	Talk	Remote communication between server and client side
520	Routing Information Protocol	RIP router protocol

TCP identification Information

There are three modes that TCP identification information:

- Match-any mode: match-any { ack | fin | psh | rst | syn | urg } TCP identification information can be configured more than one after match-any at the same time, but if it matches one of those, it will match successfully.
- Match-all mode: match-all { ack | fin | psh | rst | syn | urg }
- P identification information can be configure more than one after match-all at the same time, and frames need to match all of these configurations to match successfully.
- Established: represents identification is ACK or RST

Six identification spots in TCP head frames:

- URG (100000): identify urgent pointer is valid
- ACK (010000): identify confirm that the serial number is valid
- PSH (001000): identify receiver should send the frame sections to application layer as soon as possible
- RST (000100): identifying re-established connections
- SYN (000010): sync number, use it to initiate a link
- FIN (000001): logo sender finishing sending

8.4.2 Configuration

Basic MAC/IP filtering

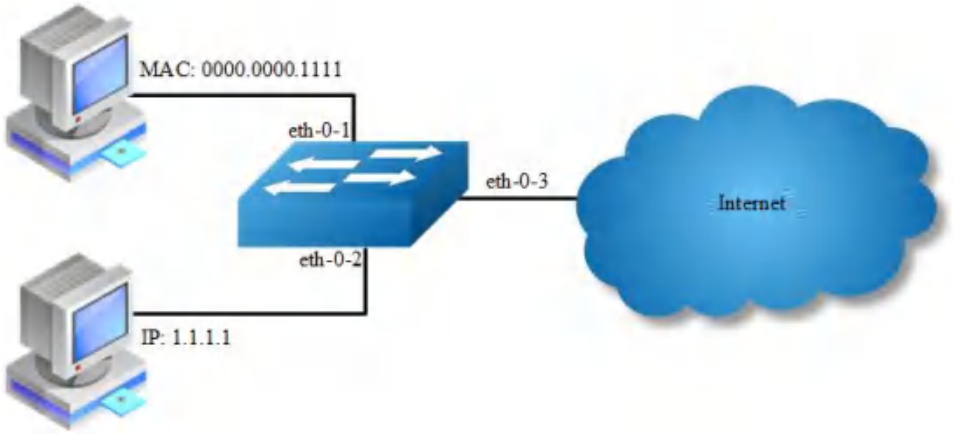


Figure 8-5 acl

In this example, use MAC ACL on interface eth-0-1, to permit packets with source mac 0000.0000.1111 and deny any other packets. Use IPv4 ACL on interface eth-0-2, to permit packets with source ip 1.1.1.1/24 and deny any other packets.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create access list

mac access list:

```
Switch(config)# mac access-list mac
Switch(config-mac-acl)# permit src-mac host 0000.0000.1111 dest-mac any
Switch(config-mac-acl)# deny src-mac any dest-mac any
Switch(config-mac-acl)# exit
```

ip access list:

```
Switch(config)# ip access-list ipv4
Switch(config-ip-acl)# permit any 1.1.1.1 0.0.0.255 any
Switch(config-ip-acl)# deny any any any
Switch(config-ip-acl)# exit
```

step 3 Create class-map, and bind the access list

```
Switch(config)# class-map cmap1
Switch(config-cmap)# match access-group mac
Switch(config-cmap)# exit

Switch(config)# class-map cmap2
Switch(config-cmap)# match access-group ipv4
Switch(config-cmap)# exit
```

step 4 Create policy-map and bind the class map

```
Switch(config)# policy-map pmap1
Switch(config-pmap)# class cmap1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

Switch(config)# policy-map pmap2
Switch(config-pmap)# class cmap2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

step 5 Apply the policy to the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy input pmap1
Switch(config-if)# exit

Switch(config-if)# interface eth-0-2
Switch(config-if)# service-policy input pmap2
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

The result of show running-config is as follows:

```
Switch# show running-config
mac access-list mac
 10 permit src-mac host 0000.0000.1111 dest-mac any
 20 deny src-mac any dest-mac any
!
ip access-list ipv4
 10 permit any 1.1.1.0 0.0.0.255 any
```

```
 20 deny any any any
!
class-map match-any cmap1
 match access-group mac
!
class-map match-any cmap2
 match access-group ipv4
!
policy-map pmap1
 class cmap1
!
policy-map pmap2
 class cmap2
!
interface eth-0-1
 service-policy input pmap1
!
interface eth-0-2
 service-policy input pmap2
!
```

Access control with a time-range

Topology

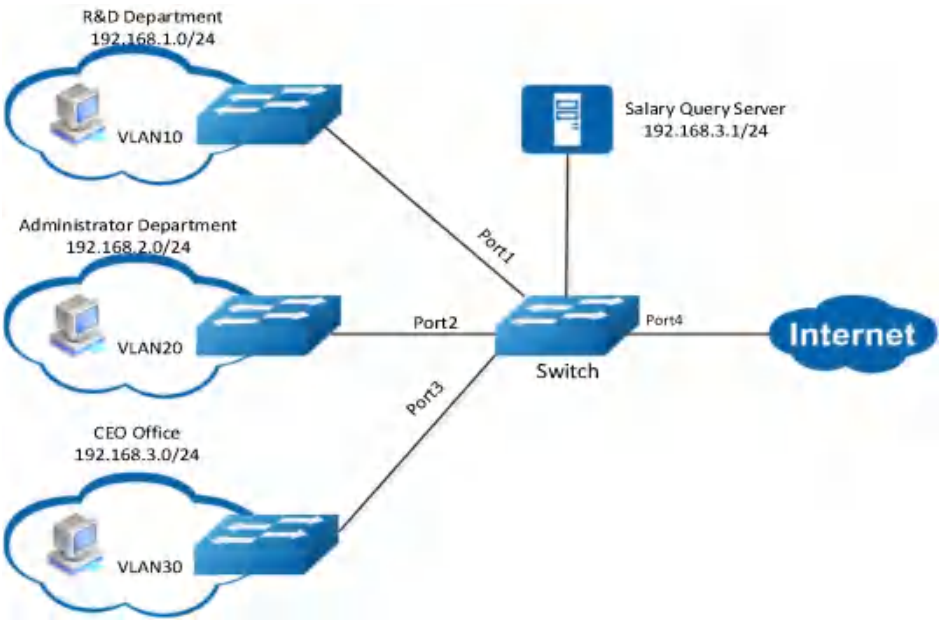


Figure 8-6 Enterprise departments network

Configuration Ideas

R&D and administrative departments can NOT access the salary query server (IP address is 192.168.10.10) during work time (9:00 – 18:00). CEO office is not limited.

Configuration Steps

step 1 Create VLAN 10, 20 and 30

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20,30
Switch(config-vlan)# exit
```

step 2 Configure port 1-3 as trunk port, and join VLAN 10, 20 and 30

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# interface eth-0-3
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
```

step 3 Configure IP address for VLAN interfaces

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.1.254/24
Switch(config-if)# interface vlan 20
Switch(config-if)# ip address 192.168.2.254/24
Switch(config-if)# interface vlan 30
Switch(config-if)# ip address 192.168.3.254/24
```

step 4 Configure port 4 as switchport and set an ip address

```
Switch(config-if)# interface eth-0-4
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.168.10.254/24
Switch(config-if)# exit
```

step 5 Configure a time range with Monday to Friday, 9.00 to 18.00 periodic

```
Switch(config)# time-range time-1
Switch(config-tm-range)# periodic 09:00 weekdays to 18:00
```

step 6 Configure ACE for R&D department to access salary query server

```
Switch(config)# ip access-list test
Switch(config-ip-acl)# deny any 192.168.1.0 0.0.0.255 host 192.168.10.10 time-range time-1
```

step 7 Configure ACE for administrative department to access salary query server

```
Switch(config)# ip access-list test
Switch(config-ip-acl)# deny any 192.168.2.0 0.0.0.255 host 192.168.10.10 time-range time-1
```

step 8 Create a class-map and bind the ACL

```
Switch(config)# class-map match-any cmap
Switch(config-cmap)# match access-group test
```

step 9 create a tpolicy-map and bind the class-map. Enable statistic feature for this class-map

```
Switch(config)# policy-map pmap
Switch(config-pmap)# class cmap
Switch(config-pmap-c)# statistics enable
```

step 10 Apply the policy-map to the interface

R&D department's data that access server will get into switch from interface eth-0-1, so apply traffic policing at eth-0-1 at inbound

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy input pmap
```

Administrative department's data that access server will get into switch from interface eth-0-2, so apply traffic policing at eth-0-2 at inbound

```
Switch(config)# interface eth-0-2
Switch(config-if)# service-policy input pmap
```

Validation

Display the configuration information of ACL

```
Switch# show access-list ip
ip access-list test
 10 deny any 192.168.1.0 0.0.0.255 host 192.168.10.10 time-range time-1
 20 deny any 192.168.2.0 0.0.0.255 host 192.168.10.10 time-range time-1
```

Display configuration information of class-map

```
Switch# show class-map
   CLASS-MAP-NAME: class-default (match-any)
   CLASS-MAP-NAME: cmap (match-any)
        match access-group: test
```

Display configuration of policy-map

```
Switch# show policy-map
 POLICY-MAP-NAME: pmap
   State: attached
   CLASS-MAP-NAME: cmap
    match access-group: test
       statistics  : enable
```

Display the statistics information that hit the ACEs

```
Switch# show policy-map statistics interface eth-0-1 input ace-based
 Interface: eth-0-1
 Ingress service policy: pmap

Class name: cmap, operator : match-any
 access-group test
  10 deny any 192.168.1.0 0.0.0.255 host 192.168.10.10 time-range time-1 ( 0 match 0 bytes)
  20 deny any 192.168.2.0 0.0.0.255 host 192.168.10.10 time-range time-1 ( 0 match 0 bytes)
total 0 match 0 bytes
```

Configuration file

Configuration of Switch's ACL

```
vlan database
 vlan 10,20,30
   periodic 09:00 weekdays to 18:00
!
ip access-list test
  10 deny any 192.168.1.0 0.0.0.255 host 192.168.10.10 time-range time-1
  20 deny any 192.168.2.0 0.0.0.255 host 192.168.10.10 time-range time-1
!
class-map match-any cmap
 match access-group test
```

```
!
policy-map pmap
 class cmap
  statistics enable
!
interface eth-0-1
 switchport mode trunk
 switchport trunk allowed vlan add 10
 service-policy input pmap
!
time-range time-1
interface eth-0-2
 switchport mode trunk
 switchport trunk allowed vlan add 20
 service-policy input pmap
!
interface eth-0-3
 switchport mode trunk
 switchport trunk allowed vlan add 30
!
interface eth-0-4
 no switchport
 ip address 192.168.10.254/24
!
interface vlan10
ip address 192.168.1.254/24
!
interface vlan20
ip address 192.168.2.254/24
!
interface vlan30
ip address 192.168.3.254/24
!
```

Limit Users from Different VLAN to Access to Each Other Using ACL

Topology

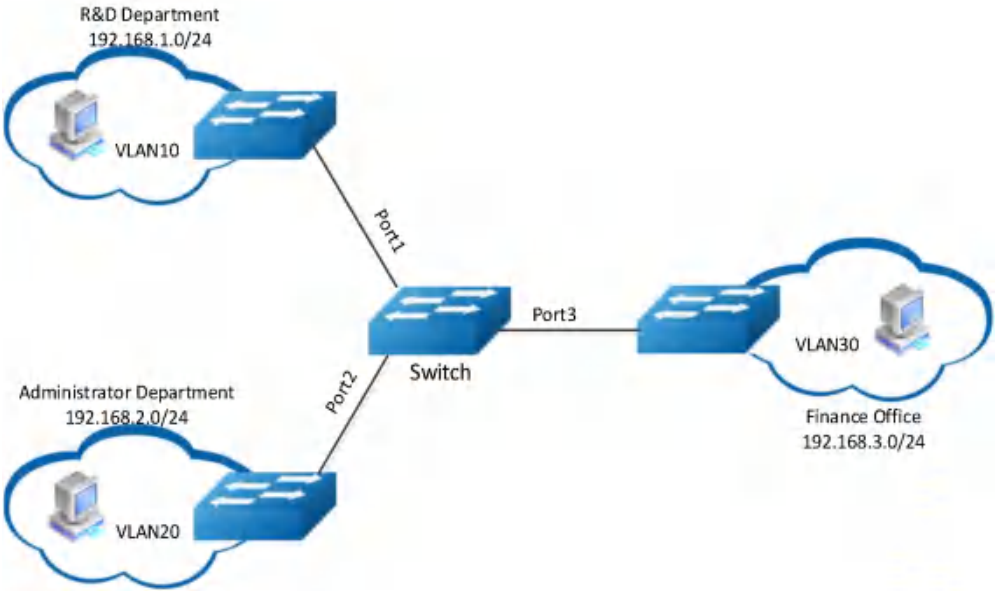


Figure 8-7 Enterprise departments network

Configuration Ideas

Network administrator divides different departments into different VLAN, and different VLANs are divided into different network segment IP. To prevent divulging company's secret information between different departments, it is asked to prohibit access for different departments by switches; that is, forbid networking between different VLAN.

Configuration Steps

step 1 Create VLAN 10, 20 and 30

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20,30
Switch(config-vlan)# exit
```

step 2 Configure port 1-3 as trunk interface and join into VLAN 10, 20 and 30

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# interface eth-0-3
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
```

step 3 Configure IP address of VLAN interface

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.1.254/24
Switch(config-if)# interface vlan 20
Switch(config-if)# ip address 192.168.2.254/24
Switch(config-if)# interface vlan 30
Switch(config-if)# ip address 192.168.3.254/24
```

step 4 create ACL "for-rd" to deny the traffic from R&D department to administrative/finance department.

```
Switch(config)# ip access-list for-rd
Switch(config-ip-acl)# deny any 192.168.1.0  0.0.0.255  192.168.2.0  0.0.0.255
Switch(config-ip-acl)# deny any 192.168.1.0  0.0.0.255  192.168.3.0  0.0.0.255
```

step 5 Create ACL "for-administrative" to deny the traffic from administrative department to R&D/finance department.

```
Switch(config)# ip access-list for-administrative
Switch(config-ip-acl)# deny any 192.168.2.0  0.0.0.255  192.168.1.0  0.0.0.255
Switch(config-ip-acl)# deny any 192.168.2.0  0.0.0.255  192.168.3.0  0.0.0.255
```

step 6 create ACL "for-finance" to deny the traffic from finance department to R&D/administrative department.

```
Switch(config)# ip access-list for-finance
Switch(config-ip-acl)# deny any 192.168.3.0  0.0.0.255  192.168.1.0  0.0.0.255
Switch(config-ip-acl)# deny any 192.168.3.0  0.0.0.255  192.168.2.0  0.0.0.255
```

step 7 Create traffic classification "cmap-rd", apply "for-rd" into this classification.

```
Switch(config)# class-map match-any cmap-rd
Switch(config-cmap)# match access-group for-rd
```

step 8 create "camp-administrative", apply "for-administrative" into this traffic classification

```
Switch(config)# class-map match-any cmap-administrative
Switch(config-cmap)# match access-group for-administrative
```

step 9 create "cmap-finance", apply "for-finance" into this traffic classification

```
Switch(config)# class-map match-any cmap-finance
Switch(config-cmap)# match access-group for-finance
```

step 10 create policy-map "pmap-rd", apply "cmap-rd" into this policy-map, and enable the ACL statistic function

```
Switch(config)# policy-map pmap-rd
Switch(config-pmap)# class cmap-rd
Switch(config-pmap-c)# statistics enable
```

step 11 Create policy-map "pmap-administrative", apply "cmap-administrative" into this policy-map, and enable the ACL statistic function

```
Switch(config)# policy-map pmap-administrative
Switch(config-pmap)# class cmap-administrative
Switch(config-pmap-c)# statistics enable
```

step 12 create policy-map "pmap-finance", apply "cmap-finance" into this policy-map, and enable the ACL statistic function

```
Switch(config)# policy-map pmap-finance
Switch(config-pmap)# class cmap-finance
Switch(config-pmap-c)# statistics enable
```

step 13 apply traffic policy-map "pmap-rd" on VLAN 10 (R&D department)

```
Switch(config)# interface vlan 10
Switch(config-if)# service-policy input pmap-rd
```

step 14 apply traffic policy-map "pmap-administrative" on VLAN 20 (administrative department)

```
Switch(config)# interface vlan 20
Switch(config-if)# service-policy input pmap- administrative
```

step 15 apply traffic policy-map "pmap-finance" on VLAN 30 (finance department)

```
Switch(config)# interface vlan 30
Switch(config-if)# service-policy input pmap-finance
```

Validation

Display configuration information of ACL

```
Switch# show access-list ip
ip access-list for-rd
 10 deny any 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
 20 deny any 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
ip access-list for-administrative
 30 deny any 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
 40 deny any 192.168.2.0 0.0.0.255 192.168.3.0 0.0.0.255
ip access-list for-finance
 30 deny any 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
   40 deny any 192.168.3.0 0.0.0.255 192.168.2.0 0.0.0.255
```

Display traffic classification configuration

```
Switch# show class-map
  CLASS-MAP-NAME: class-default (match-any)
```

```
  CLASS-MAP-NAME: cmap-rd (match-any)
    match access-group: for-rd
  CLASS-MAP-NAME: cmap-administrative (match-any)
    match access-group: for-administrative
  CLASS-MAP-NAME: cmap-finance (match-any)
      match access-group: for-finance
```

Display traffic policing configuration

```
Switch# show policy-map
 POLICY-MAP-NAME: pmap-rd
  State: attached
  CLASS-MAP-NAME: cmap-rd
   match access-group: for-rd
    statistics  : enable
 POLICY-MAP-NAME: pmap-administrative
  State: attached
  CLASS-MAP-NAME: cmap-administrative
   match access-group: for-administrative
    statistics  : enable
 POLICY-MAP-NAME: pmap-finance
  State: attached
  CLASS-MAP-NAME: cmap-finance
   match access-group: for-finance
    statistics  : enable
```

Configuration file

Configuration of Switch's ACL

```
vlan database
 vlan 10,20,30
!
ip access-list for-rd
 10 deny any 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
 20 deny any 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
ip access-list for-administrative
 10 deny any 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
 20 deny any 192.168.2.0 0.0.0.255 192.168.3.0 0.0.0.255
ip access-list for-finance
 10 deny any 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
 20 deny any 192.168.3.0 0.0.0.255 192.168.2.0 0.0.0.255
!
class-map match-any cmap-rd
 match access-group for-rd
!
class-map match-any cmap-administrative
 match access-group for-administrative
!
class-map match-any cmap-finance
 match access-group for-finance
!
policy-map pmap-rd
 class cmap-rd
statistics enable
!
policy-map pmap-administrative
```

```
 class cmap-administrative
  statistics enable
!
policy-map pmap-finance
 class cmap-finance
  statistics enable
!
interface eth-0-1
 switchport mode trunk
 switchport trunk allowed vlan add 10
!
interface eth-0-2
 switchport mode trunk
 switchport trunk allowed vlan add 20
!
interface eth-0-3
 switchport mode trunk
 switchport trunk allowed vlan add 30
!
interface eth-0-4
 no switchport
 ip address 192.168.10.254/24
!
interface vlan10
 service-policy input pmap-rd
 ip address 192.168.1.254/24
!
interface vlan20
 service-policy input pmap-administrative
 ip address 192.168.2.254/24
!
interface vlan30
 service-policy input pmap-finance
 ip address 192.168.3.254/24
```

NOTE

- Apply ACL on VLAN interface, it can enable all physical interfaces in this VLAN, but it is only valid for layer3 forwarding packets.
- Apply ACL on VLAN, it can enable all physical interfaces in this VLAN and valid for both layer2/layer3 forwarding packets.
- The configuration example in this section will create different ACLs that apply to different class-map and policy-map. This configuration can save ACL resources. If does not care about saving ACL resources, it can apply all ACLs into one class-map and policy-map to simplify the configuration.

ACL flow Redirect

Topology

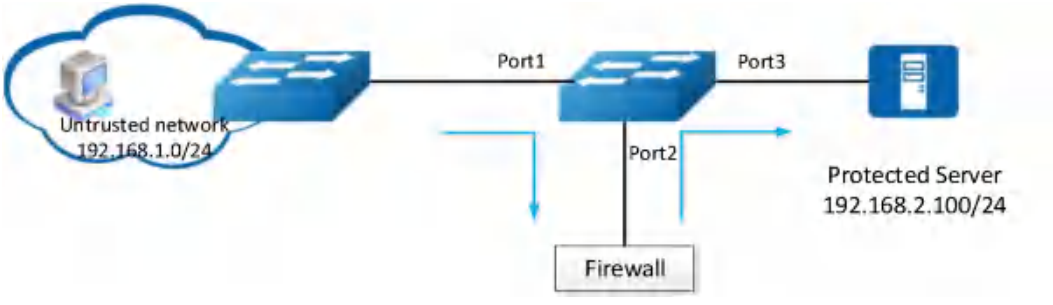


Figure 8-8 ACL flow Redirect

Configuration Ideas

Network administrator need to put those traffic from untrusted networks to access the protected servers into firewalls; after the firewalls finish filtering, it will send the traffic back to the switch.

Configuration Steps

step 1 create VLAN 10 and 20

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20
```

step 2 Configure port 1\2\3 as trunk port, and join into corresponding VLAN

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# ex
Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# ex
Switch(config)# interface eth-0-3
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
```

step 3 Configure IP address of VLAN

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.1.254/24
Switch(config-if)# exit
Switch(config)# interface vlan 20
Switch(config-if)# ip address 192.168.2.254/24
```

step 4 Create ACL to match the traffic from untrusted networks to the protected server

```
Switch(config)# ip access-list test
Switch(config-ip-acl)# permit any any host 192.168.2.100
```

step 5 Create a class-map and bind the ACL

```
Switch(config)# class-map cmap
Switch(config-cmap)# match access-group test
```

step 6 Create policy-map and bind the class map, enable the statistic feature and redirect feature

```
Switch(config)# policy-map pmap
Switch(config-pmap)# class cmap
Switch(config-pmap-c)# statistics enable
Switch(config-pmap-c)# redirect to interface eth-0-2
```

step 7 Disable MAC learning function of interface eth-0-2

```
Switch(config)# interface eth-0-2
Switch(config-if)# mac learning disable
```

step 8 Apply the policy-map to the interface

```
Switch(config)#interface eth-0-1
Switch(config-if) service-policy input pmap
```

Validation

check the configuration information of ACL regulation

```
Switch# show access-list ip
ip access-list test
 10 permit any any host 192.168.2.100
```

Display configuration of traffic classification

```
Switch# show class-map
   CLASS-MAP-NAME: class-default (match-any)
   CLASS-MAP-NAME: cmap (match-any)
    match access-group: test
```

Display configuration of traffic policing

```
Switch# show policy-map
 POLICY-MAP-NAME: pmap
   State: detached
   CLASS-MAP-NAME: cmap
    match access-group: test
      statistics  : enable
      redirect    : interface eth-0-2
```

Configuration file

Configuration of Switch's ACL

```
vlan database
 vlan 10,20
!
ip access-list test
 10 permit any any host 192.168.2.100
!
class-map match-any cmap
 match access-group test
!
policy-map pmap
 class cmap
  statistics enable
  redirect to interface eth-0-2
!
interface eth-0-1
 switchport mode trunk
 switchport trunk allowed vlan add 10
 service-policy input pmap
!
interface eth-0-2
 switchport mode trunk
 switchport trunk allowed vlan add 10
 mac learning disable
!
interface eth-0-3
 switchport mode trunk
 switchport trunk allowed vlan add 20
```

```
!
interface vlan10
 ip address 192.168.1.254/24
!
interface vlan20
 ip address 192.168.2.254/24
```

NOTE

Since firewall will send it back to switch after it disposing messages. If interface eth-0-2 does not close its MAC learning function, it will lead to learn the same MAC address for eth-0-1 and eth-0-2, which will lead MAC address flapping; and the messages that the server replies cannot reach port eth-0-1 correctly. Therefore, closing MAC learning function of interface eth-0-2 is necessary.

To enable monitor function based on ACL, refer to above configuration examples.

8.5 Configuring Extern ACL

8.5.1 Overview

Function Introduction

Extend IPv4 ACL combines MAC filters with IP filters in one access list. Different from MAC and IP ACL, extend ACL can access-control all packets (IP packets and non-IP packets). Extend ACL supported extend IPv4 ACL.

Principle Description

Following is a brief description of terms and concepts used to describe the extend ACL：

- **Extend IPv4 ACL**： Extend IPv4 ACL takes advantages of MAC ACL and IPv4 ACL, which combines MAC ACE with IPv4 ACE in an ACL to provide more powerful function of access-controlling traverse packets.
- **MAC ACE**： Filter packets by mac-sa and mac-da, and the mac-address can be masked, or configured as host id, or configured as any to filter all MAC addresses. Other L2 fields, such as COS, VLAN-ID, INNER-COS, INNER-VLAN-ID, L2 type, L3 type, can also be filtered by MAC ACE.
- **IPv4 ACE**： Filter packets by ip-sa and ip-da, and ip-address can be masked, or configured as host id, or configured as any to filter all IPv4 address. Other L3 fields such as DSCP, L4 protocol and L4 fields, such as TCP port, UDP port, can also be filtered by IPv4 ACE.

The MAC ACE and IPv4 ACE in an extend IPv4 ACL can be configured alternately in arbitrary order which is completely specified by user.

8.5.2 Configuration

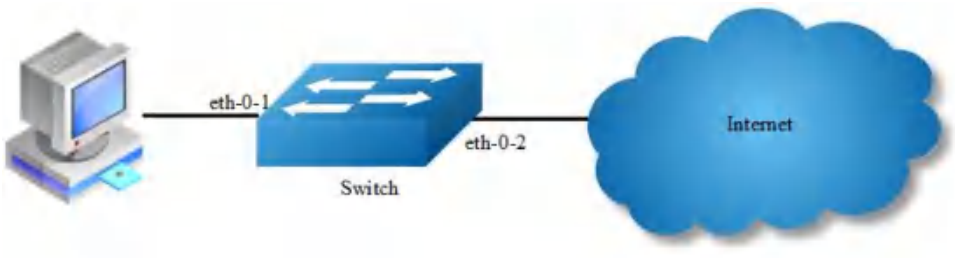


Figure 8-9 extern acl

In this example, use extend IPv4 ACL on interface eth-0-1, to permit packets with source mac 0000.0000.1111 and cos value of 2, permit all TCP packets, and deny any other packets.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create access list

```
Switch(config)# ip access-list ipxacl extend
Switch(config-ex-ip-acl)# permit src-mac host 0000.0000.1111 dest-mac any cos 2
Switch(config-ex-ip-acl)# permit tcp any any
Switch(config-ex-ip-acl)# deny src-mac any dest-mac any
Switch(config-ex-ip-acl)# end
```

step 3 Create class-map, and bind the access list

```
Switch(config)# class-map cmap
Switch(config-cmap)# match access-group ipxacl
Switch(config-cmap)# exit
```

step 4 Create policy-map and bind the class map

```
Switch(config)# policy-map pmap
Switch(config-pmap)# class cmap
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

step 5 Apply the policy to the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy input pmap
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

The result of show running-config is as follows:

```
Switch# show running-config
ip access-list ipxacl extend
 10 permit src-mac host 0000.0000.1111 dest-mac any cos 2
 20 permit tcp any any
 30 deny src-mac any dest-mac any
!
class-map match-any cmap
 match access-group ipxacl
!
policy-map pmap
 class cmap
!
interface eth-0-1
 service-policy input pmap
!
Switch# show access-list ip
ip access-list ipxacl extend
 10 permit src-mac host 0000.0000.1111 dest-mac any cos 2
 20 permit tcp any any
 30 deny src-mac any dest-mac any
```

8.6 Configuring IPv6 ACL

8.6.1 Overview

Function Introduction

Access control lists for IPv6 (ACLv6) classify traffic with the same characteristics. The ACLv6 can have multiple access control entries (ACEs), which are commands that match fields against the contents of the packet. ACLv6 can filter packets received on interface by many fields such as ipv6 address and deny or permit the packets.

Principle Description

The following terms and concepts are used to describe ACLv6.

- **Access control entry (ACE):** Each ACE includes an action element (permit or deny) and a filter element based on criteria such as source address, destination address, protocol, and protocol-specific parameters.
- **IPv6 ACL:** IPv6 ACL can filter packet by ipv6-sa and ipv6-da, and ipv6-address can be masked, or configured as host id, or configured as any to filter all IPv6 address. IPv6 ACL can also filter other L3 fields such as L4 protocol and L4 fields such as TCP port, UDP port, and so on.
- **Time Range:** Time range can define a period of time only between which the ACE can be valid if the ACE is associated to the time range.

8.6.2 Configuration

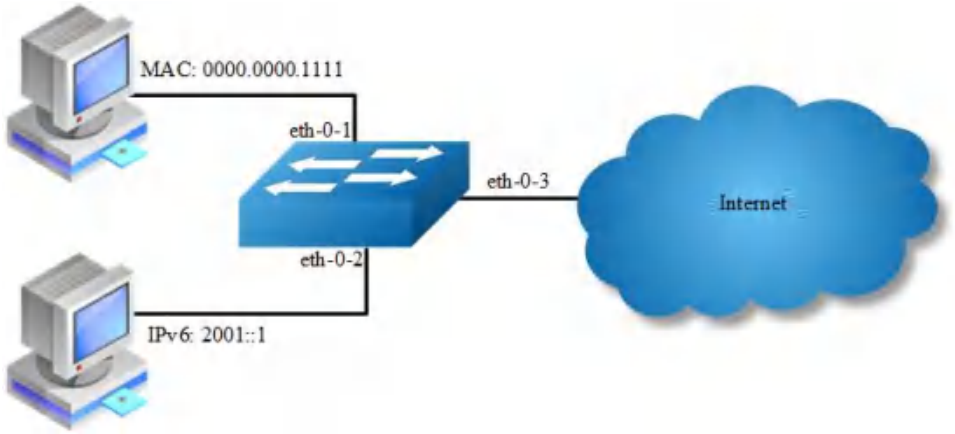


Figure 8-10 ipv6 acl

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable IPv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create access list

mac access list:

```
Switch(config)# mac access-list mac
Switch(config-mac-acl)# permit src-mac host 0000.0000.1111 dest-mac any
Switch(config-mac-acl)# deny src-mac any dest-mac any
Switch(config-mac-acl)# exit
```

ipv6 access list:

```
Switch(config)# ipv6 access-list ipv6
Switch(config-ipv6-acl)# permit any 2001::/64 any
Switch(config-ipv6-acl)# deny any any any
Switch(config-ipv6-acl)# exit
```

step 4 Create class-map, and bind the access list

```
Switch(config)# class-map cmap1
Switch(config-cmap)# match access-group mac
Switch(config-cmap)# exit

Switch(config)# class-map cmap2
Switch(config-cmap)# match access-group ipv6
Switch(config-cmap)# exit
```

step 5 Create policy-map and bind the class map

```
Switch(config)# policy-map pmap1
Switch(config-pmap)# class cmap1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

Switch(config)# policy-map pmap2
Switch(config-pmap)# class cmap2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

step 6 Apply the policy to the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy input pmap1
Switch(config-if)# exit

Switch(config-if)# interface eth-0-2
Switch(config-if)# service-policy input pmap2
Switch(config-if)# exit
```

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

If IPv6 is enabled globally, the IPv6 packet will not obey the MAC ACL rules:

```
Switch# show running-config
mac access-list mac
 10 permit src-mac host 0000.0000.1111 dest-mac any
 20 deny src-mac any dest-mac any
!
ipv6 access-list ipv6
 10 permit any 2001::/64 any
 20 deny any any any
!
class-map match-any cmap1
 match access-group mac
!
class-map match-any cmap2
 match access-group ipv4
```

```
!
policy-map pmap1
 class cmap1
!
policy-map pmap2
 class cmap2
!
interface eth-0-1
 service-policy input pmap1
!
interface eth-0-2
 service-policy input pmap2
!
```

8.6.3 Application cases

N/A

8.7 Configuring Port-Group

8.7.1 Overview

Function Introduction

Port-group is designed to implement a port group based on ACL rules. Multiple interfaces can be added to the port group, supporting physical interfaces and aggregation interfaces. When the user applies ACL policy to the port group, there's only one rule and the action of ACL has a aggregate effect.

Principle Description

N/A

8.7.2 Configuration

Create a port group

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a port group and add member interfaces

```
Switch(config)# port-group port_group_1
Switch(config-port-group)# member interface eth-0-1
Switch(config-port-group)# member interface agg 1
Switch(config-port-group)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
DUT1# show running-config port-group
port-group port_group_1
  member interface eth-0-1
  member interface agg1
```

8.7.3 Application cases

N/A

8.8 Configuring Vlan-Group

8.8.1 Overview

Function Introduction

Vlan-group is designed to implement a vlan group based on ACL rules. Multiple vlan can be added to the vlan group. When the user applies ACL policy to the vlan group, there's only one rule and the action of ACL has a aggregate effect.

Principle Description

N/A

8.8.2 Configuration

Create a vlan group

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a vlan group and add member vlan

```
Switch(config)# vlan-group vlan_group_1
Switch(config-vlan-group)# member vlan 10
Switch(config-vlan-group)# member vlan 20
Switch(config-vlan-group)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
DUT1# show running-config vlan-group
vlan-group vlan_group_1
  member vlan 10
  member vlan 20
```

8.8.3 Application cases

N/A

8.9 Configuring COPP ACL

8.9.1 Overview

Function Introduction

COPP is mainly used to diacard or limit the rate of the packets which is transmitted to cpu. It guarantees that cpu can deal with traffic normally. In the base of original exception, copp can make a careful control of the packets transmitted to cpu.

Principle Description

The following terms and concepts are used to describe ACL： - **Access control entry (ACE)**: Each ACE includes an action element (permit or deny) and a series of filter element based on criteria such as source address, destination address, protocol, and protocol-specific parameters. - **COPP ACL**:COPP ACL deals with packets according to their exceptions, the system can support the following exceptions: any,ipda, fwd-to-cpu, slow-protocol, bpdu, erps, eapol, smart-link, dhcp, rip,ospf, pim, bgp, vrrp, ldp, ptp, rsvp, icmp-redirect, mcast-rpf-fail,macsa-mismatch,vlan-security-discard, post-security-discard, ip-option,udld,dot1x-mac-bypass, 12protocol-tunnel, arp, igmp, ssh, telnet, mlag. COPP only deals with the packets transmitted to cpu, it will not handle the forwarding packets. - **Time Range**: Time range can define a period of time only between which the ACE can be valid if the ACE is associated to the time range.

8.9.2 Configuration

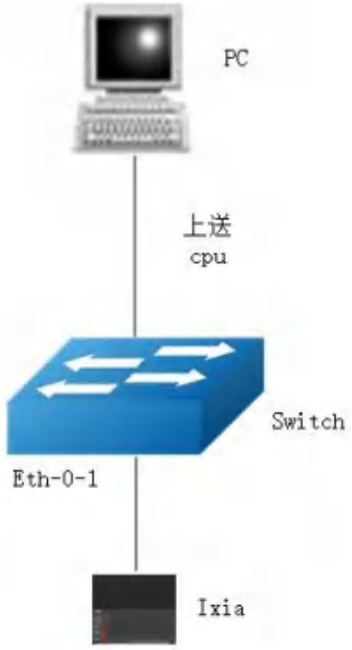


Figure 8-11 copp_acl

In this example, use COPP ACL on interface eth-0-1, to discard the packets with arp exception transmitted to cpu. In the first place, you can use ixia to create a packet, Destination Address:001E.0811.065D, Source Address:0000.0010.0000, the type of arp is arp-request, Sender Hardware Address:0000.0000.0000, Target Protocol Address:10.0.0.1,the rest configuration information is as follows.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create copp access list

copp access list:

```
Switch(config)# control-plane access-list test1
Switch(config-cp-acl)# deny exception arp arp-request
Switch(config-cp-acl)# exit
```

step 3 Create class-map, and bind the copp access list

```
Switch(config)# class-map type control-plane cmap1
Switch(config-cmap-cp)# match access-group test1
Switch(config-cmap-cp)# exit
```

step 4 Create policy-map and bind the class map

```
Switch(config)#policy-map type control-plane pmap1
Switch(config-pmap-cp)#class type control-plane cmap1
Switch(config-pmap-cp-c)#exit
Switch(config-pmap-cp)#exit
```

step 5 Apply the policy to the interface

```
Switch(config)#control-plane
Switch(config-control-plane)#service-policy type control-plane input pmap1
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

The result of show running-config is as follows:

```
Switch# show running-config
control-plane access-list test1
 10 deny exception arp arp-request
!
class-map type control-plane cmap1
 match access-group test1
!
policy-map type control-plane pmap1
 class type control-plane cmap1
!
control-plane
 service-policy type control-plane input pmap1
```

The result of show cpu traffic-statistics receive is as follows:

```
Switch# show cpu traffic-statistics receive

statistics rate time is 5 second(s)

reason      count(packets)      rate(pps)

arp         1029059             0

total       1029059             0
```

8.9.3 Application cases

N/A

8.10 Configuring dot1x

8.10.1 Overview

Function Introduction

IEEE 802 Local Area Networks are often deployed in environments that permit unauthorized devices to be physically attached to the LAN infrastructure, or Permit unauthorized users to attempt to access the LAN through equipment already attached.

Port-based network access control makes use of the physical access characteristics of IEEE 802 LAN infrastructures in order to provide a means of authenticating and authorizing devices attached to a LAN port that has point-to-point connection characteristics, and of preventing access to that port in cases in which the authentication and authorization process fails.

With 802.1X port-based authentication, the devices in the network have specific roles:

- Client: the device (PC) that requests access to the LAN and switch services and responds to requests from the switch. The client software with support the follow the 802.1X standard should run on the PC. For linux system, we recommend the application which named "xsupplicant".
- Authentication server: performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. In this release, the Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server. RADIUS operates in a client/server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

- Switch (edge switch or wireless access point): controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulation the EAP frames and Interacting with the authentication server. When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is re-encapsulated in the RADIUS format. The EAP Frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client. We can enable dot1x on routed port and access port.

Principle Description

Reference to IEEE Std 802.1X- 2004

8.10.2 Configuration

Basic dot1x configuration

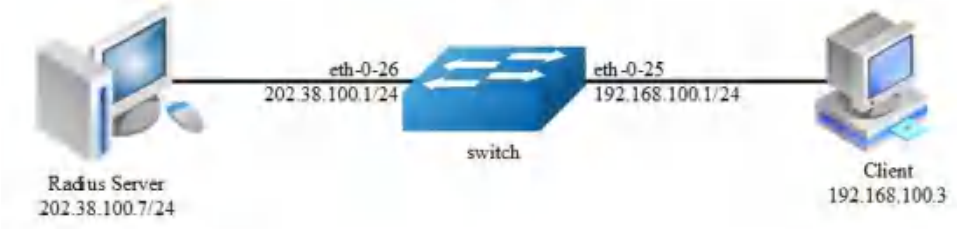


Figure 8-12 dot1x

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable dot1x globally

```
Switch(config)# dot1x system-auth-ctrl
```

step 3 Enter the interface configure mode, set the attributes of the interface and enable dot1x

```
Switch(config)# interface eth-0-25
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface vlan 1
Switch(config-if)# ip address 192.168.100.1/24
Switch(config-if)# exit
```

step 4 Set the attributes of Layer 3 interface and set the Radius server

```
Switch(config)# interface eth-0-26
Switch(config-if)# no switchport
Switch(config-if)# ip address 202.38.100.1/24
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# radius-server host 202.38.100.7
Switch(config)# radius-server host 2001:1000::1
Switch(config)# radius-server key test
Switch(config)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show dot1x
802.1X Port-Based Authentication Enabled
 RADIUS server address: 2001:1000::1:1812
 Next radius message ID: 0
 RADIUS server address: 202.38.100.7:1812
 Next radius message ID: 0
Switch# show dot1x interface eth-0-25
802.1X info for interface eth-0-25
 portEnabled          : true
 portControl          : Auto
 portMode             : Port based
 portStatus           : Authorized
 Mac Auth bypass      : disabled
 reAuthenticate       : disabled
 reAuthPeriod         : 3600
 Max user number      : 255
 Current session number : 1
 Accept user number    : 1
 Reject user number    : 0
 Guest VLAN           : N/A
 Assign VLAN          : N/A
 QuietPeriod          : 60
 ReqMax               : 2
 TxPeriod             : 30
 SuppTimeout          : 30
 ServerTimeout        : 30
 CD: adminControlledDirections : in
 CD: operControlledDirections  : in
 CD: bridgeDetected          : false
========================================

session 1: 1 - 0011.0100.0001
----------------------------------------
 user name : admin
 abort:F fail:F start:F timeout:F success:T
 PAE: state: Authenticated - portMode: Auto
 PAE: reAuthCount: 0 - rxRespId: 0
 BE: state: Idle - reqCount: 0 - idFromServer: 5
```

Enable dot1x on routed port

The example above describes how to enable dot1x on access port. This function can also enable on routed port. The following example shows how to change eth-0-25 to a routed port and enable dot1x.

```
Switch(config)# interface eth-0-25
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.168.100.1/24
Switch(config-if)# dot1x port-control auto
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Using force mode

Dot1x port control mode can be force-authorized or force-unauthorized.

force-authorized:

```
Switch(config)# interface eth-0-25
Switch(config-if)# dot1x port-control force-authorized
Switch(config-if)# exit
```

force-unauthorized:

```
Switch(config)# interface eth-0-25
Switch(config-if)# dot1x port-control force-unauthorized
Switch(config-if)# exit
```

User can choose port control mode as force-authorized,force-unauthorized or auto. The final configuration should over write the previous one.

Enable dot1x accounting

Dot1x accounting can be used to keep track of network usage after user is authenticated. Dot1x accounting is disabled by default, you can enable it on globally configure mode.

Enable dot1x accounting:

```
Switch(config)# dot1x accounting-mode radius
```

Device sends accounting start request to server after user is authenticated when dot1x accounting is enabled, if no corresponding response is received, start-fail policy is needed :

- **online**：In order to avoid the impact of network failure on users, online policy can be configured to allow users to be online.
- **offline**：If dot1x accounting start fail, offline policy can be configured to reject users to be online。

```
Switch(config)# dot1x accounting start-fail online
```

User can configure realtime accounting to make device send realtime accounting request to server periodically. Server keeps accounting users only when received realtime accounting request, so that abnormal accounting can be avoided when server can not receive accounting stop packet from device.

Meanwhile, user can configure max times of realtime accounting with no response and the action when realtime accounting fails. By default, the max times of realtime accounting with no response is set to 3, and user is allowed to be online after realtime accounting failure.

```
Switch(config)# dot1x accounting realtime 60
Switch(config)# dot1x accounting interim-fail max-times 2 offline
```

dot1x optional parameter

Timer for Radius server: Set the wait time for re-activating RADIUS server; Set the maximum failed RADIUS requests sent to server; Set the timeout value for no response from RADIUS server.

```
Switch(config)# radius-server deadtime 10
Switch(config)# radius-server retransmit 5
Switch(config)# radius-server timeout 10
```

Interface attributes: Specify the number of reauthentication attempts before becoming unauthorized; Set the protocol version; Specify the quiet period in the HELD state; Enable reauthentication on a port; Specify the seconds between reauthorization attempts; Specify the authentication server response timeout; Specify the supplicant response timeout; Specify the Seconds between successive request ID attempts;

Enable dot1x handshake with client on a port; Specify the handshake period.

```
Switch(config)# interface eth-0-25
Switch(config-if)# dot1x  max-req  5
Switch(config-if)# dot1x protocol-version 1
Switch(config-if)# dot1x quiet-period 120
Switch(config-if)# dot1x reauthentication
Switch(config-if)# dot1x timeout re-authperiod 1800
```

```
Switch(config-if)# dot1x timeout server-timeout 60
Switch(config-if)# dot1x timeout supp-timeout 60
Switch(config-if)# dot1x timeout tx-period 60
Switch(config-if)# dot1x handshake
Switch(config-if)# dot1x timeout handshake-period 1
Switch(config-if)# exit
```

8.10.3 Application cases

Radius server configuration (Using WinRadius for example)

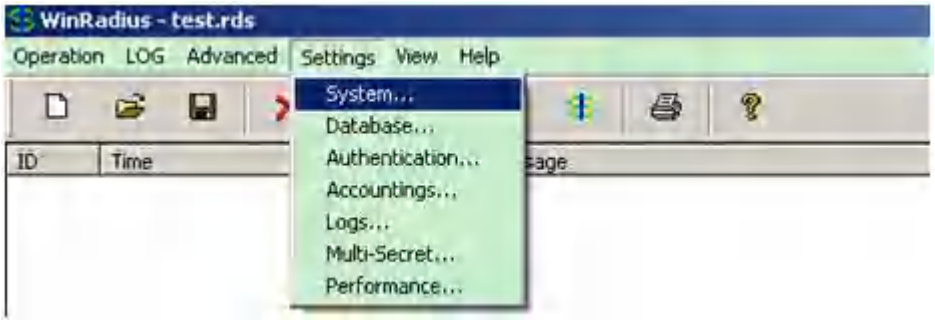


Figure 8-13 Select "Setting-> System"



Figure 8-14 Configure the shared-key, authorization port and account port

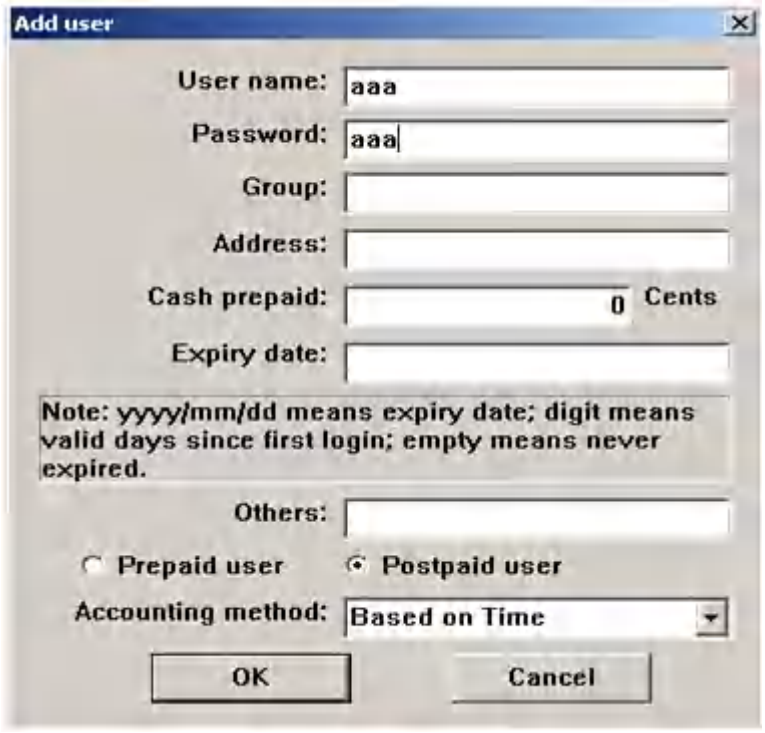


Figure 8-15 Add user name and password on the server

8.11 Configuring Guest VLAN

8.11.1 Overview

Function Introduction

You can configure a guest VLAN for each 802.1x port on the switch to provide limited services to clients (for example, how to download the 802.1x client). These clients might be upgrading their system for 802.1x authentication, and some hosts, such as Windows 98 systems, might not be 802.1x-capable.

When the authentication server does not receive a response to its EAPOL request/identity frame, clients that are not 802.1x-capable are put into the guest VLAN for the port, if one is configured. However, the server does not grant 802.1x-capable clients that fail authentication access to the network. Any number of hosts is allowed access when the switch port is moved to the guest VLAN.

The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.



Guest VLAN is supported on access port, and not supported on routed port or trunk port.

Principle Description

无

8.11.2 Configuration

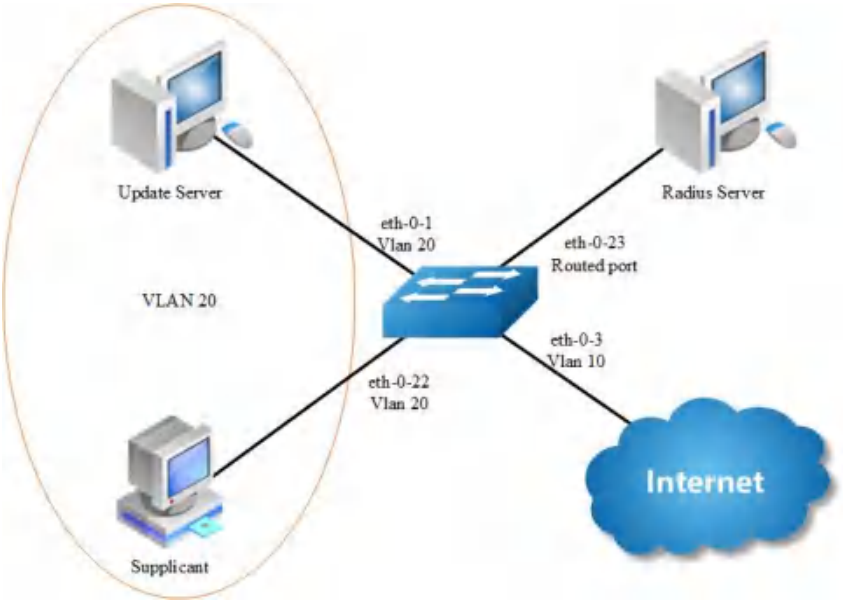


Figure 8-16 Guest vlan: before authenticated

In the above topology, eth-0-22 is an IEEE 802.1X enabled port, and it is in the native VLAN 10, the configured guest VLAN for this port is VLAN 20. So clients that are not 802.1X capable will be put into VLAN 20 after the authenticator had send max EAPOL request/identity frame but got no response.

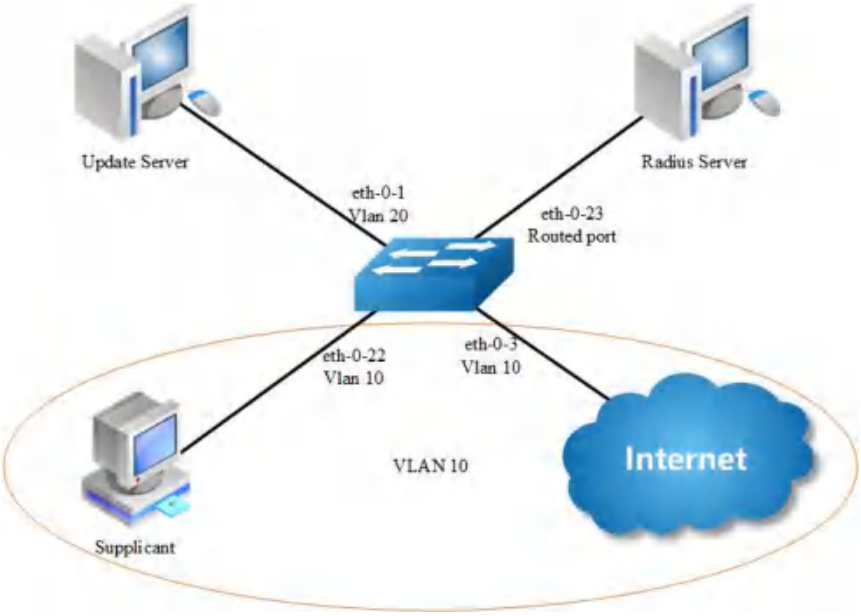


Figure 8-17 Guest vlan: after authenticated

We use remote linux Radius server as authenticate server, the server's address is 202.38.100.7, and the IP address for the connected routed port eth-0-23 is 202.38.100.1. When the client is authenticated by the radius server, then it can access the public internet which is also in VLAN 10.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# vlan 20
Switch(config-vlan)# exit
```

step 3 Enable dot1x globally

```
Switch(config)# dot1x system-auth-ctrl
```

step 4 Enter the interface configure mode, set the attributes of the interface and enable dot1x and set guest vlan

```
Switch(config)# interface eth-0-22
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# dot1x port-control auto
Switch( config-if)# no shutdown
Switch(config-if)# dot1x guest vlan 20
Switch(config-if)# exit
```

step 5 Set the attributes of Layer 3 interface and set the Radius server

```
Switch(config)# interface eth-0-23
Switch(config-if)# no switchport
Switch(config-if)# ip address 202.38.100.1/24
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# radius-server host 202.38.100.7
Switch(config)# radius-server key test
Switch(config)#end
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Init state:

```
Switch# show running-config
dot1x system-auth-ctrl
radius-server host 202.38.100.7 key test
vlan database
 vlan 10,20
!
interface eth-0-22
 switchport access vlan 10
 dot1x port-control auto
 dot1x guest-vlan 20
!
interface eth-0-23
 no switchport
 ip address 202.38.100.1/24
!

Switch# show dot1x interface eth-0-22
802.1X info for interface eth-0-22
```

```
portEnabled          : true
portControl          : Auto
portMode             : Port based
portStatus           : Unauthorized
Mac Auth bypass      : disabled
reAuthenticate       : disabled
reAuthPeriod         : 3600
Max user number      : 255
Current session number : 0
Accept user number   : 0
Reject user number   : 0
Guest VLAN           : 20
Assign VLAN          : N/A
QuietPeriod          : 60
ReqMax               : 2
TxPeriod             : 30
SuppTimeout          : 30
ServerTimeout        : 30
CD: adminControlledDirections : in
CD: operControlledDirections  : in
CD: bridgeDetected           : false
=======================================

Switch# show vlan brief
VLAN ID  Name         State   STP ID  DSCP    Member ports
                               (u)-Untagged, (t)-Tagged
======= ================ ======= ======= ======= ========================
1     default        ACTIVE  0     Disable eth-0-1(u) eth-0-2(u)
                                   eth-0-3(u) eth-0-4(u)
                                   eth-0-5(u) eth-0-6(u)
                                   eth-0-7(u) eth-0-8(u)
                                   eth-0-9(u) eth-0-10(u)
                                   eth-0-11(u) eth-0-12(u)
                                   eth-0-13(u) eth-0-14(u)
                                   eth-0-15(u) eth-0-16(u)
                                   eth-0-17(u) eth-0-18(u)
                                   eth-0-19(u) eth-0-20(u)
                                   eth-0-21(u) eth-0-24(u)
                                   eth-0-25(u) eth-0-26(u)
                                   eth-0-27(u) eth-0-28(u)
                                   eth-0-29(u) eth-0-30(u)
                                   eth-0-31(u) eth-0-32(u)
                                   eth-0-33(u) eth-0-34(u)
                                   eth-0-35(u) eth-0-36(u)
                                   eth-0-37(u) eth-0-38(u)
                                   eth-0-39(u) eth-0-40(u)
                                   eth-0-41(u) eth-0-42(u)
                                   eth-0-43(u) eth-0-44(u)
                                   eth-0-45(u) eth-0-46(u)
                                   eth-0-47(u) eth-0-48(u)
10    VLAN0010       ACTIVE  0     Disable eth-0-22(u)
20    VLAN0020       ACTIVE  0     Disable
```

After configure the guest vlan:

unauthorized:

```
Switch# show dot1x interface eth-0-22
802.1X info for interface eth-0-22
 portEnabled          : true
 portControl          : Auto
 portMode             : Port based
 portStatus           : Unauthorized
 Mac Auth bypass      : disabled
 reAuthenticate       : disabled
 reAuthPeriod         : 3600
 Max user number      : 255
 Current session number : 1
 Accept user number   : 0
 Reject user number   : 1
 Guest VLAN           : 20(Port Authorized by guest vlan)
 Assign VLAN          : N/A
 QuietPeriod          : 60
 ReqMax               : 2
 TxPeriod             : 30
 SuppTimeout          : 30
 ServerTimeout        : 30
 CD: adminControlledDirections : in
 CD: operControlledDirections  : in
 CD: bridgeDetected           : false
=======================================

session 1: 1 - 0011.0100.0001
---------------------------------------
 user name : admin
 abort:F fail:T start:F timeout:F success:F
 PAE: state: Held - portMode: Auto
 PAE: reAuthCount: 1 - rxRespId: 0
 BE: state: Idle - reqCount: 0 - idFromServer: 92

Switch# show vlan brief
VLAN ID  Name          State   STP ID  DSCP    Member ports
                                 (u)-Untagged, (t)-Tagged
======= ================ ======= ======= ======= ========================
1      default        ACTIVE  0      Disable eth-0-1(u) eth-0-2(u)
                                    eth-0-3(u) eth-0-4(u)
                                    eth-0-5(u) eth-0-6(u)
                                    eth-0-7(u) eth-0-8(u)
                                    eth-0-9(u) eth-0-10(u)
                                    eth-0-11(u) eth-0-12(u)
                                    eth-0-13(u) eth-0-14(u)
                                    eth-0-15(u) eth-0-16(u)
                                    eth-0-17(u) eth-0-18(u)
                                    eth-0-19(u) eth-0-20(u)
                                    eth-0-21(u) eth-0-24(u)
                                    eth-0-25(u) eth-0-26(u)
                                    eth-0-27(u) eth-0-28(u)
                                    eth-0-29(u) eth-0-30(u)
                                    eth-0-31(u) eth-0-32(u)
                                    eth-0-33(u) eth-0-34(u)
                                    eth-0-35(u) eth-0-36(u)
                                    eth-0-37(u) eth-0-38(u)
```

```
                                eth-0-39(u) eth-0-40(u)
                                eth-0-41(u) eth-0-42(u)
                                eth-0-43(u) eth-0-44(u)
                                eth-0-45(u) eth-0-46(u)
                                eth-0-47(u) eth-0-48(u)
10    VLAN0010       ACTIVE  0      Disable
20    VLAN0020       ACTIVE  0      Disable eth-0-22(u)
  Client is authenticated
```

authorized:

```
Switch# show dot1x interface eth-0-22
802.1X info for interface eth-0-22
 portEnabled          : true
 portControl          : Auto
 portMode             : Port based
 portStatus           : Authorized
 Mac Auth bypass      : disabled
 reAuthenticate       : disabled
 reAuthPeriod         : 3600
 Max user number      : 255
 Current session number : 1
 Accept user number    : 1
 Reject user number    : 0
 Guest VLAN           : 20
 Assign VLAN          : N/A
 QuietPeriod          : 60
 ReqMax               : 2
 TxPeriod             : 30
 SuppTimeout          : 30
 ServerTimeout        : 30
 CD: adminControlledDirections : in
 CD: operControlledDirections  : in
 CD: bridgeDetected           : false
==========================================

session 1: 1 - 0011.0100.0001
------------------------------------------
 user name : admin
 abort:F fail:F start:F timeout:F success:T
 PAE: state: Authenticated - portMode: Auto
 PAE: reAuthCount: 0 - rxRespId: 0
 BE: state: Idle - reqCount: 0 - idFromServer: 207

Switch# show vlan brief
VLAN ID  Name          State   STP ID  DSCP    Member ports
                                 (u)-Untagged, (t)-Tagged
======= ================ ======== ======= ======= ========================
1     default        ACTIVE  0      Disable eth-0-1(u) eth-0-2(u)
                                eth-0-3(u) eth-0-4(u)
                                eth-0-5(u) eth-0-6(u)
                                eth-0-7(u) eth-0-8(u)
                                eth-0-9(u) eth-0-10(u)
                                eth-0-11(u) eth-0-12(u)
                                eth-0-13(u) eth-0-14(u)
                                eth-0-15(u) eth-0-16(u)
```

```
                          eth-0-17(u) eth-0-18(u)
                          eth-0-19(u) eth-0-20(u)
                          eth-0-21(u) eth-0-24(u)
                          eth-0-25(u) eth-0-26(u)
                          eth-0-27(u) eth-0-28(u)
                          eth-0-29(u) eth-0-30(u)
                          eth-0-31(u) eth-0-32(u)
                          eth-0-33(u) eth-0-34(u)
                          eth-0-35(u) eth-0-36(u)
                          eth-0-37(u) eth-0-38(u)
                          eth-0-39(u) eth-0-40(u)
                          eth-0-41(u) eth-0-42(u)
                          eth-0-43(u) eth-0-44(u)
                          eth-0-45(u) eth-0-46(u)
                          eth-0-47(u) eth-0-48(u)
10   VLAN0010     ACTIVE  0     Disable eth-0-22(u)
20   VLAN0020     ACTIVE  0     Disable

Switch# show dot1x
802.1X Port-Based Authentication Enabled
 RADIUS server address: 202.38.100.7:1812
 Next radius message ID: 0
Switch# show dot1x statistics
=====================================
802.1X statistics for interface eth-0-22
 EAPOL Frames Rx: 52 - EAPOL Frames Tx: 4270
 EAPOL Start Frames Rx: 18 - EAPOL Logoff Frames Rx: 2
 EAP Rsp/Id Frames Rx: 29 - EAP Response Frames Rx: 3
 EAP Req/Id Frames Tx: 3196 - EAP Request Frames Tx: 3
 Invalid EAPOL Frames Rx: 0 - EAP Length Error Frames Rx: 0
 EAPOL Last Frame Version Rx: 2 - EAPOL Last Frame Src: ae38.3288.f046
```

8.11.3 Application cases

N/A

8.12 Configuring ARP Inspection

8.12.1 Overview

Function Introduction

ARP inspection is a security feature that validates ARP packets in a network. ARP inspection intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from some man-in-the-middle attacks. ARP inspection ensures that only valid ARP requests and responses are relayed. The switch performs these activities:

Intercept all ARP requests and responses on untrusted ports.

Verify that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination.

Drop invalid ARP packets.

ARP inspection determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a trusted database, the DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the switch. If the ARP packet is received on a trusted interface, the switch forwards the packet without any checks. On entrusted interfaces, the switch forwards the packet only if it is valid.

Principle Description

Following is a brief description of terms and concepts used to describe the ARP Inspection:

- **DHCP Snooping**: DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers. This feature builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- **Address Resolution Protocol (ARP)**: ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A , but it does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of Host A. All hosts within the broadcast domain receive the ARP request, and Host A responds with its MAC address.

8.12.2 Configuration

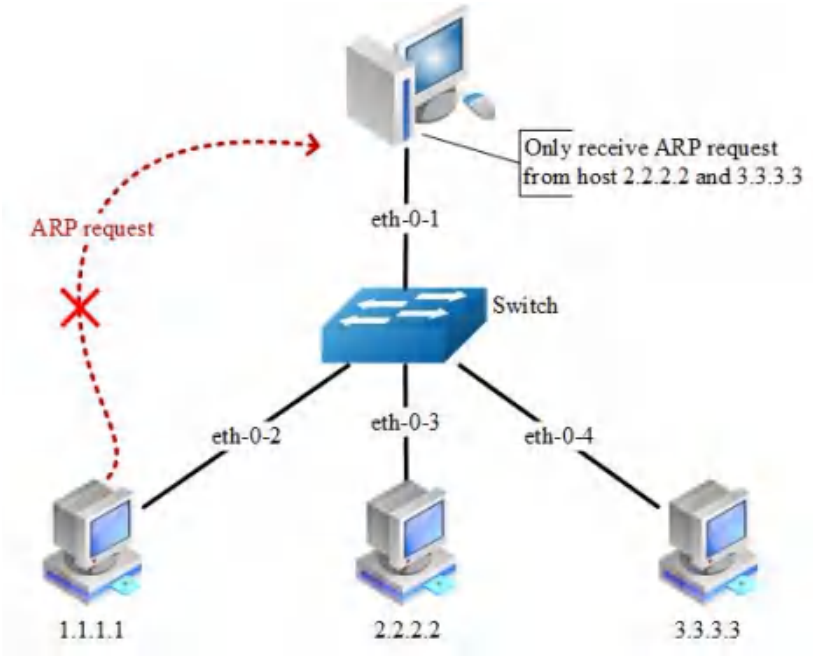


Figure 8-18 arp inspection

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2
Switch(config-vlan)# exit
Switch(config)# exit
```

step 3 Enter the interface configure mode, add the interface into the vlan

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit

Switch(config)# interface eth-0-3
```

```
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit

Switch(config)# interface eth-0-4
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit
```

step 4 Configure arp inspection

```
Switch(config)# interface eth-0-1
Switch(config-if)# ip arp inspection trust
Switch(config-if)# exit
Switch(config)# ip arp inspection vlan 2
Switch(config)# ip arp inspection validate src-mac ip dst-mac
```

step 5 Configure arp access list

```
Switch(config)# arp access-list test
Switch(config-arp-acl)# deny request ip host 1.1.1.1 mac any
Switch(config-arp-acl)# exit
Switch(config)# ip arp inspection filter test vlan 2
```

step 6 Exit the configure mode

```
Switch(config)# exit
```

step 7 Validation

Check the configuration of ARP Inspection on switch:

```
Switch# show ip arp inspection
Source Mac Validation      : Enabled
Destination Mac Validation  : Enabled
IP Address Validation      : Enabled
Vlan    Configuration    ACL Match    Static ACL
================================================================
2       enabled          test
Vlan    ACL Logging    DHCP Logging
================================================================
2       deny           deny
Vlan    Forwarded     Dropped    DHCP Drops    ACL Drops
================================================================
2       0            0          0           0
Vlan    DHCP Permits    ACL Permits    Source MAC Failures
================================================================
2       0              0              0
Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
================================================================
2      0               0                    0
```

Show the log information of ARP Inspection on switch:

```
Switch# show ip arp inspection log
Total Log Buffer Size : 32
Syslog rate : 5 entries per 1 seconds.
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
```

```
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
1970-01-02 00:30:47 : Drop an ARP packet by ACL on vlan 2
```

8.12.3 Application cases

N/A

8.13 Configuring DHCP Snooping

8.13.1 Overview

Function Introduction

DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers.

The DHCP snooping feature performs the following activities:

- Validate DHCP messages received from untrusted sources and filters out invalid messages.
- Build and maintain the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Utilize the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Other security features, such as dynamic ARP inspection (DAI), also use information stored in the DHCP snooping binding database. DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs. The DHCP snooping feature is implemented in software basis. All DHCP messages are intercepted in the BAY and directed to the CPU for processing.

Principle Description

N/A

8.13.2 Configuration

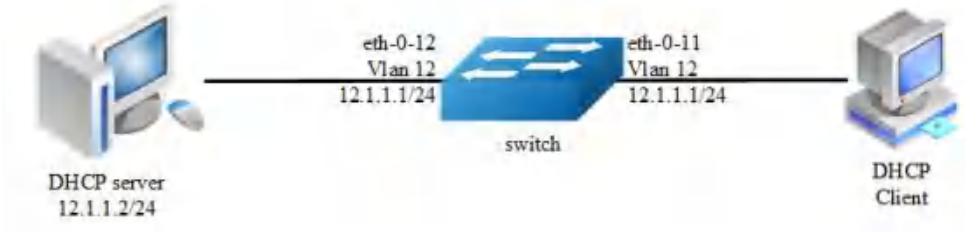


Figure 8-19 dhcp snooping

This figure is the networking topology for testing DHCP snooping functions. We need two Linux boxes and one switch to construct the test bed.

- Computer A is used as a DHCP server.
- Computer B is used as a DHCP client.
- Switch is used as a DHCP Snooping box.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 12
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode, add the interface into the vlan

```
Switch(config)# interface eth-0-12
Switch(config-if)# switchport
Switch(config-if)# switchport access vlan 12
Switch(config-if)# dhcp snooping trust
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-11
Switch(config-if)# switchport
Switch(config-if)# switchport access vlan 12
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface vlan 12
Switch(config-if)# ip address 12.1.1.1/24
Switch(config-if)# exit
```

step 4 Set DHCP attributes

```
Switch(config)# dhcp snooping verify mac-address
Switch(config)# service dhcp enable
Switch(config)# dhcp snooping
Switch(config)# dhcp snooping vlan 12
```

step 5 Exit the configure mode

```
Switch(config)# exit
```

step 6 Validation

Check the interface configuration.

```
Switch(config)# show running-config interface eth-0-12
!
interface eth-0-12
 dhcp snooping trust
 switchport access vlan 12
!

Switch(config)# show running-config interface eth-0-11
!
interface eth-0-11
 switchport access vlan 12
!
```

Check the dhcp service status.

```
Switch# show services
Networking services configuration:
Service Name       Status
============================================================
dhcp               enable
```

Print dhcp snooping configuration to check current configuration.

```
Switch# show dhcp snooping config
dhcp snooping service: enabled
dhcp snooping switch: enabled
Verification of hwaddr field: enabled
Insertion of relay agent information (option 82): disable
Relay agent information (option 82) on untrusted port: not allowed
dhcp snooping vlan 12
```

Show dhcp snooping statistics.

```
Switch# show dhcp snooping statistics
DHCP snooping statistics:
============================================================
DHCP packets                    17
BOOTP packets                   0
Packets forwarded               30
Packets invalid                 0
Packets MAC address verify failed   0
Packets dropped                 0
```

Show dhcp snooping binding information.

```
Switch# show dhcp snooping binding all
DHCP snooping binding table:
VLAN MAC Address    Interface  Lease(s)   IP Address
============================================================
12   0016.76a1.7ed9 eth-0-11   691190     12.1.1.65
```

8.13.3 Application cases

N/A

8.14 Configuring IP source guard

8.14.1 Overview

Function Introduction

IP source guard prevents IP spoofing by allowing only the IP addresses that are obtained through DHCP snooping on a particular port. Initially, all IP traffic on the port is blocked except for the DHCP packets that are captured by DHCP snooping. When a client receives a valid IP address from the DHCP server, an access control list (ACL) is installed on the port that permits the traffic from the IP address. This process restricts the client IP traffic to those source IP addresses that are obtained from the DHCP server; any IP traffic with a source IP address other than that in the ACL's permit list is filtered out. This filtering limits the ability of a host to attack the network by claiming a neighbor host's IP address.

IP source guard uses source IP address filtering, which filters the IP traffic that is based on its source IP address. Only the IP traffic with a source IP address that matches the IP source binding entry is permitted. A port's IP source address filter is changed when a new DHCP-snooping binding entry for a port is created or deleted. The port ACL is modified and reapplied in the hardware to reflect the IP source binding change. By default, if you enable IP source guard without any DHCP-snooping bindings on the port, a default ACL that denies all IP traffic is installed on the port. When you disable IP source guard, any IP source filter ACL is removed from the port.

Also IP source guard can use source IP and MAC address Filtering. When IP source guard is enabled with this option, IP traffic is filtered based on the source IP and Mac addresses. The switch forwards traffic only when the source IP and MAC addresses match an entry in the IP source binding table. If not, the switch drops all other types of packets except DHCP packet.

The switch also supports to have IP, MAC and VLAN Filtering. When IP source guard is enabled with this option, IP traffic is filtered cased on the source IP and MAC addresses. The switch forwards traffic only when the source IP, MAC addresses and VLAN match an entry in the IP source binding table.

When ip source guard is enabled on an interface, enable arp as-layer-3 in global configuration mode so that arp packets matching ip source guard entries can be forwarded normally.If it is disable, the system is discarded, affecting network communication.

Principle Description

The following terms and concepts are used to describe the IP source guard:

- **Dynamic Host Configuration Protocol (DHCP)**: Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway.
- **DHCP Snooping**: DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers. This feature builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- **ACL**: Access control list.

8.14.2 Configuration

Configure ip source guard

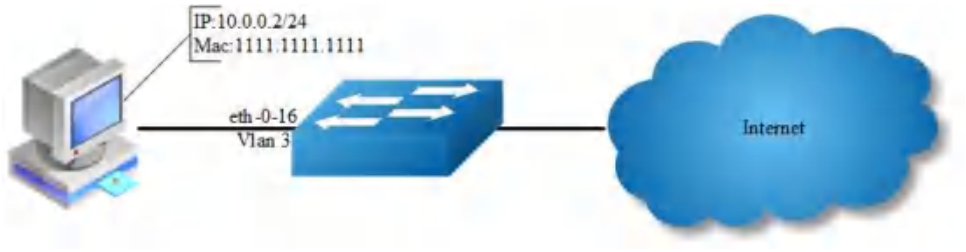


Figure 8-20 ip source guard

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable arp as-layer-3

```
Switch(config)# arp as-layer-3 enable
```

step 3 Enter the vlan configure mode and create vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 3
Switch(config-vlan)# exit
```

step 4 Enter the interface configure mode and set the attributes

```
Switch(config)# interface eth-0-16
Switch(config-if)# switchport
Switch(config-if)# no shutdown
Switch(config-if)# switchport access vlan 3
Switch(config-if)# exit
```

step 5 Add IP source guard entries

```
Switch(config)# ip source maximal binding number per-port 15
Switch(config)# ip source binding mac 1111.1111.1111 vlan 3 ip 10.0.0.2 interface eth-0-16
```

step 6 Enable IP source guard on the interface

```
Switch(config)# interface eth-0-16
Switch(config-if)# ip verify source ip
Switch(config-if)# exit
```

step 7 Exit the configure mode

```
Switch(config)# exit
```

step 8 Validation

```
Switch#show running-config interface eth-0-16
!
interface eth-0-16
 ip verify source ip
 switchport access vlan 3
```

Remove ip source guard entries

Remove by entry：

```
Switch(config)# no ip source binding mac 1111.1111.1111 vlan 3 ip 10.0.0.2 interface eth-0-16
```

Remove by interface：

```
Switch(config)# no ip source binding entries interface eth-0-16
```

Remove by vlan:

```
Switch(config)# no ip source binding entries vlan 3
```

Remove all:

```
Switch(config)# no ip source binding entries
```

8.14.3 Application cases

N/A

8.15 Configuring Private-vlan

8.15.1 Overview

Function Introduction

Private-vlan a security feature which is used to prevent from direct l2 communication among a set of ports in a vlan.

It can provide a safer and more flexible network solutions by isolating the ports which in the same VLAN.

Principle Description

N/A

8.15.2 Configuration

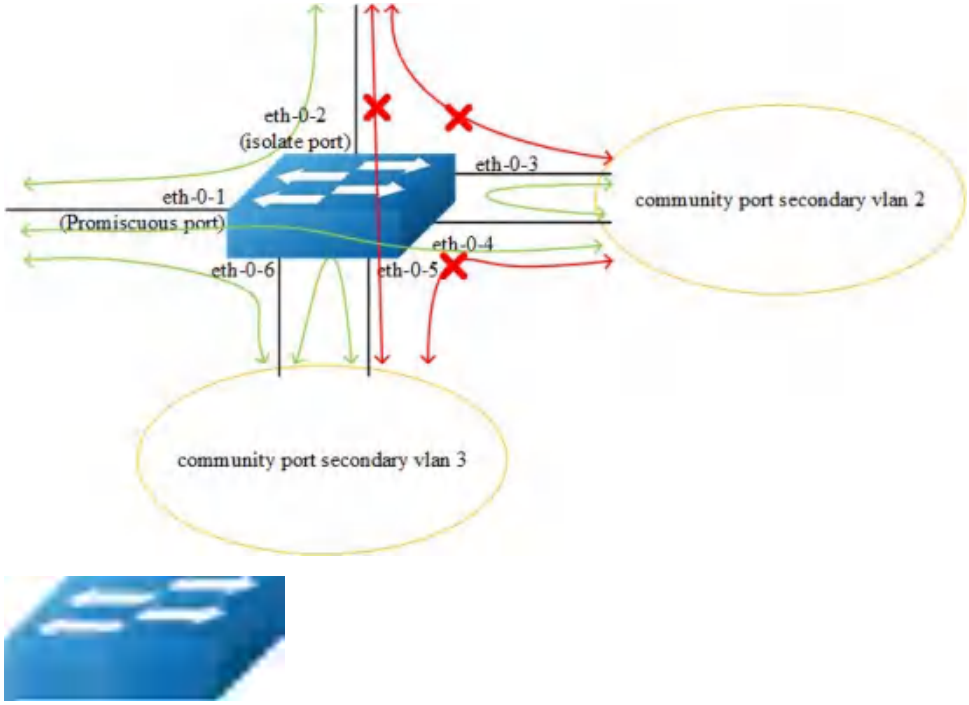


Figure 8-21 private vlan

As the figure above shows:

- All ports are in a same primary vlan.
- Port 1 is promiscuous port; it can communicate with all other ports.
- Port 2 is isolate port; it cannot communicate with all other ports except for the promiscuous port (port 1).
- Port 3 and port 4 are community ports in secondary vlan 2; they can communicate with each other. They cannot communicate with all other ports except for the promiscuous port.
- Port 5 and port6 are community ports in secondary vlan 3; they can communicate with each other. They cannot communicate with all other ports except for the promiscuous port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan

```
Switch (config)# vlan database
Switch (config-vlan)# vlan 2
Switch (config-vlan)# quit
```

step 3 Enter the interface configure mode and set the attributes

Promiscuous port： promiscuous port in pvlan can communicate with any other ports in this pvlan

```
Switch (config)# interface eth-0-1
Switch (config-if)# switchport mode private-vlan promiscuous
Switch (config-if)# switchport private-vlan 2
Switch (config-if)# quit
```

Isolate port： isolate port in pvlan can only communicate with promiscuous port in this pvlan

```
Switch (config)# interface eth-0-2
Switch (config-if)# switchport mode private-vlan host
Switch (config-if)# switchport private-vlan 2 isolate
Switch (config-if)# quit
```

Community port： community port in pvlan can communicate with promiscuous port and community ports with same community-vlan id in this pvlan

```
Switch (config)# interface eth-0-3
Switch (config-if)# switchport mode private-vlan host
Switch (config-if)# switchport private-vlan 2 community-vlan 2
Switch (config-if)# quit

Switch (config)# interface eth-0-4
Switch (config-if)# switchport mode private-vlan host
Switch (config-if)# switchport private-vlan 2 community-vlan 2
Switch (config-if)# quit

Switch (config)# interface eth-0-5
Switch (config-if)# switchport mode private-vlan host
Switch (config-if)# switchport private-vlan 2 community-vlan 3
Switch (config-if)# quit

Switch (config)# interface eth-0-6
Switch (config-if)# switchport mode private-vlan host
Switch (config-if)# switchport private-vlan 2 community-vlan 3
Switch (config-if)# quit
```

step 4 Exit the configure mode

```
Switch(config)# exit
```

step 5 Validation

The result of show private-vlan is as follows：

```
switch # show private-vlan
Primary  Secondary Type        Ports
------------------------------------------------------------------------------
2      N/A     promiscuous   eth-0-1
2      N/A     isloate       eth-0-2
2      2       community     eth-0-3  eth-0-4
2      3       community     eth-0-5  eth-0-6
```

8.15.3 Application cases

N/A

8.16 Configuring AAA

8.16.1 Overview

Function Introduction

Authentication verifies users before they are allowed access to the network and network services. System can use AAA authentication methods and Non-AAA authentication methods. RADIUS Authentication is one of AAA authentication methods. RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS is widely used protocol in network environments. It is commonly used for embedded network devices such as routers, modem servers, switches, etc. RADIUS clients run on support routers and switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

Principle Description

N/A

8.16.2 Configuration

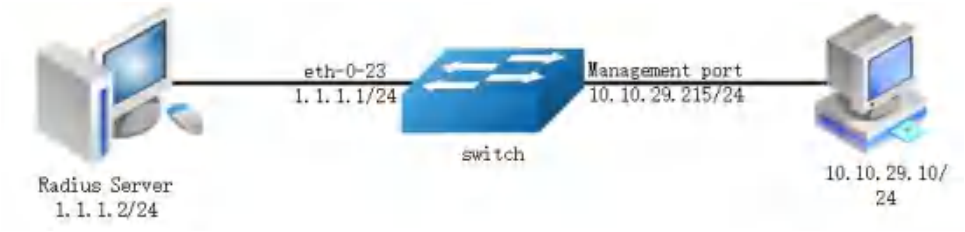


Figure 8-22 private vlan

The figure above is the networking topology for RADIUS authentication functions. We need one Switch and two computers for this test.

One computer as RADIUS server, it ip address of the eth0 interface is 1.1.1.2/24.

Switch has RADIUS authentication function. The ip address of interface eth-0-23 is 1.1.1.1/24. The management ip address of switch is 10.10.29.215, management port is connected the PC for test login, PC's ip address is 10.10.29.10.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable AAA

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login radius-login radius local
```

step 3 Configure Radius server

```
Switch(config)# radius-server host 1.1.1.2 auth-port 1819 key keyname
Switch(config)# radius-server host 2001:1000::1 auth-port 1819 key keyname
```

step 4 Configure a layer 3 interface and set ip address

```
Switch(config)# interface eth-0-23
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# quit
```

step 5 set authentication mode

```
Switch(config)# line vty 0 7
Switch(config-line)#login authentication radius-login
Switch(config-line)#privilege level 4
Switch(config-line)#no line-password
```

step 6 Exit the configure mode

```
Switch(config-line)# end
```

step 7 Validation

You can use command show authentication status in switch:

```
Switch# show aaa status
aaa status:
   Authentication enable
```

You can use command show keys in switch:

```
Switch# show aaa method-lists authentication
authen queue=AAA_ML_AUTHEN_LOGIN
  Name = default  state = ALIVE :  local
Name = radius-login  state = ALIVE :  radius  local
```

Telnet output:

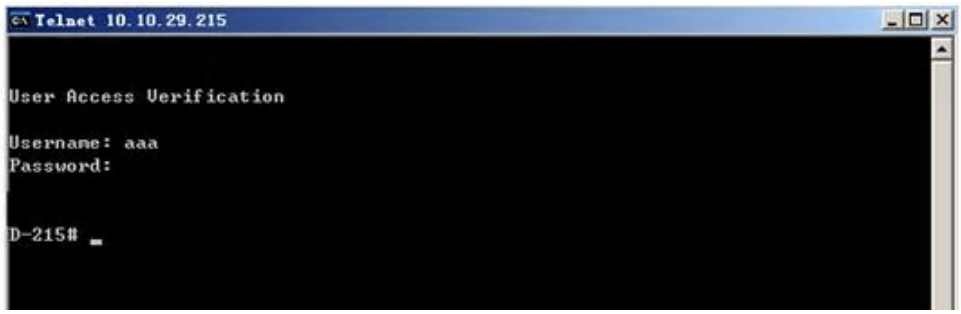


Figure 8-23 Telnet connecting test

Don't forget to turn RADIUS authentication feature on.

Make sure the cables is linked correctly You can use command to check log messages if Switch can't do RADIUS authentication:

```
Switch# show logging buffer
```

8.16.3 Application cases

Radius server configuration (Using WinRadius for example)

Set ip address for PC:

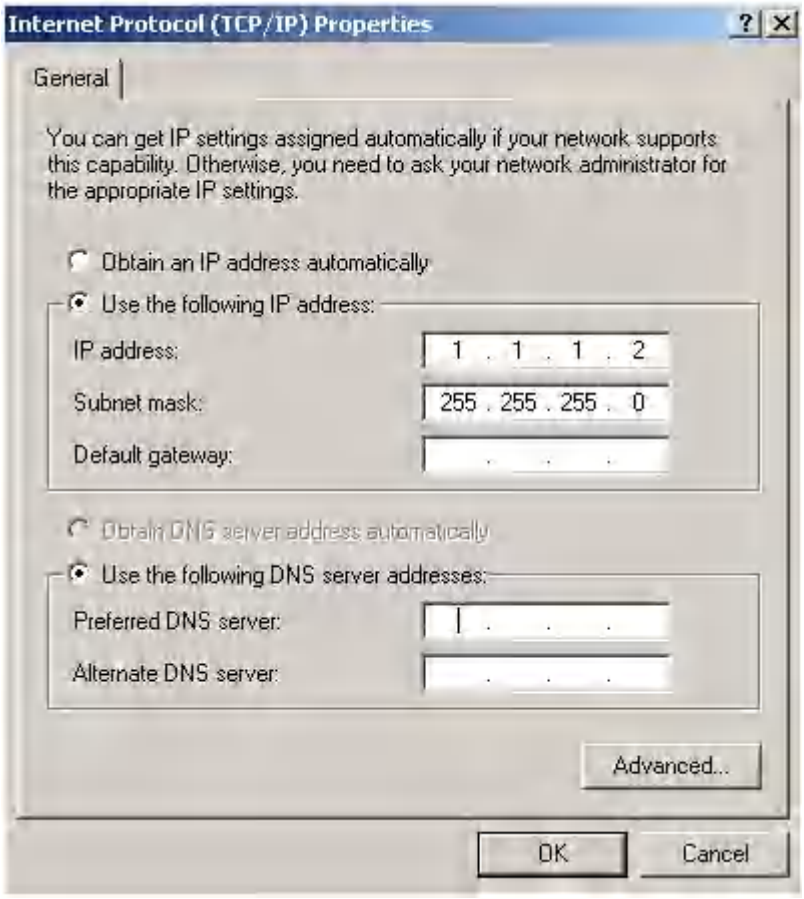


Figure 8-24 Set IP address for PC

Connectivity test between server and switch:

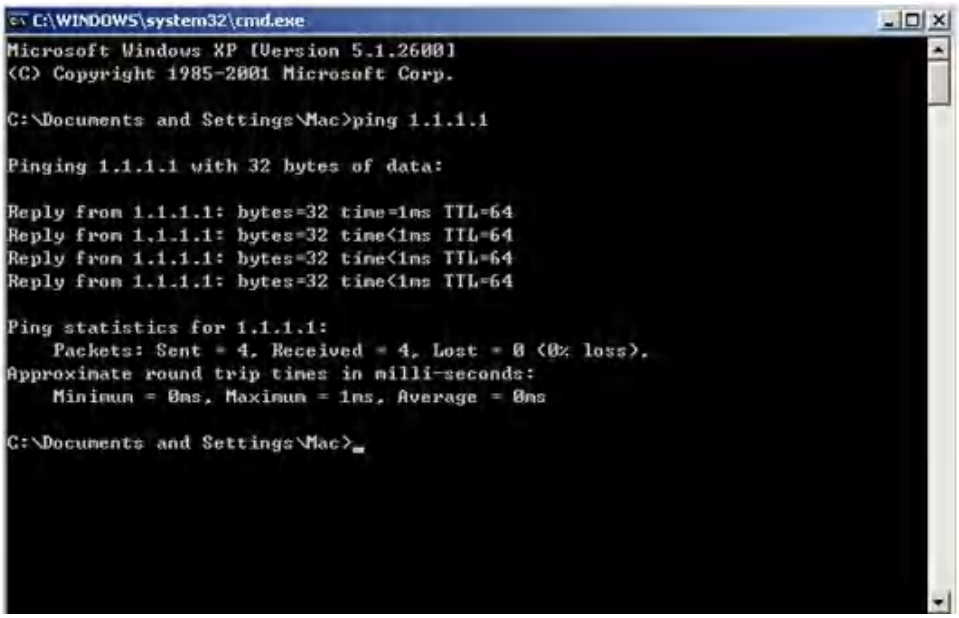


Figure 8-25 Connectivity test

Open winRadius:



Figure 8-26 WinRadius

Configurations for winRadius:

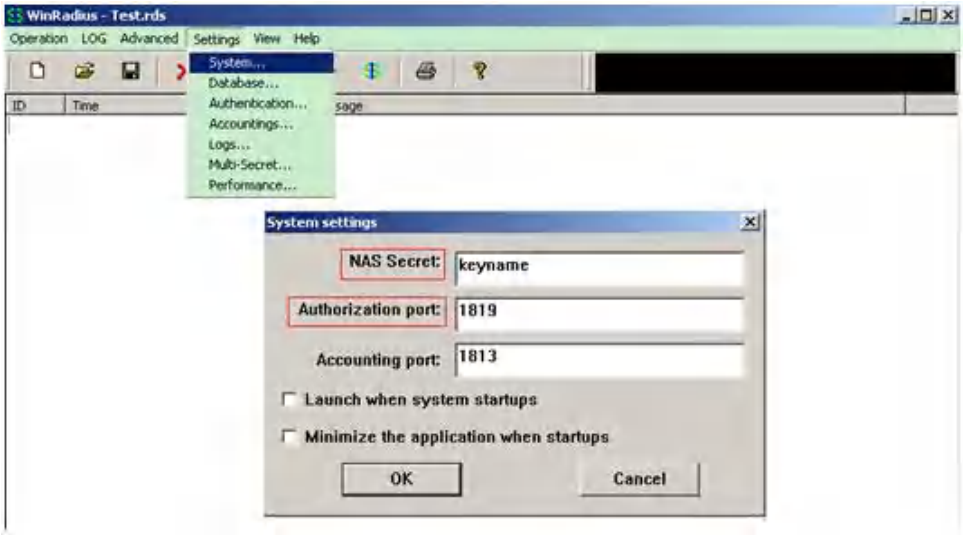


Figure 8-27 WinRadius

Add user and password:

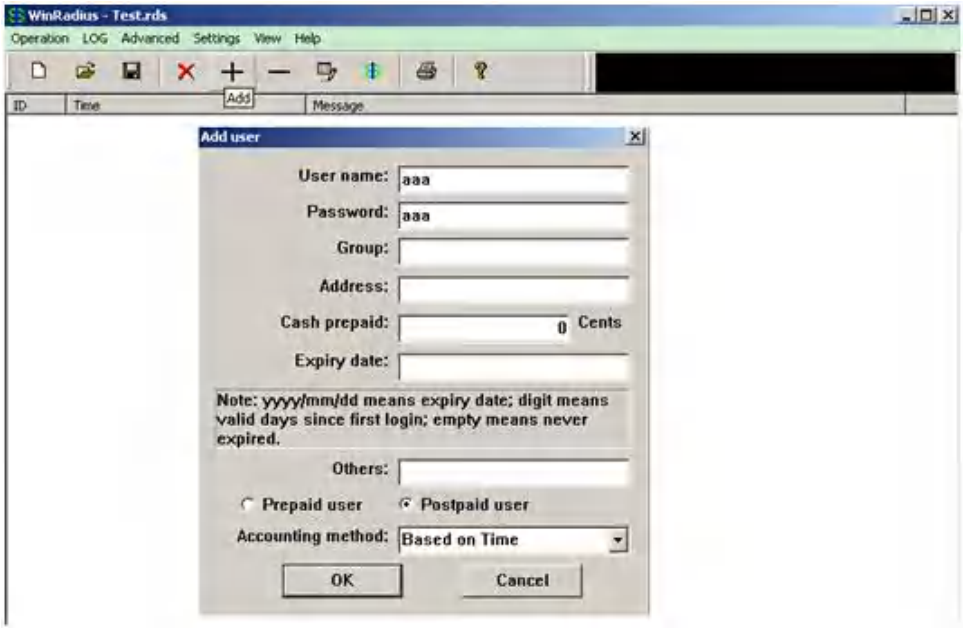


Figure 8-28 Add user and password

Connectivity test between client and switch:

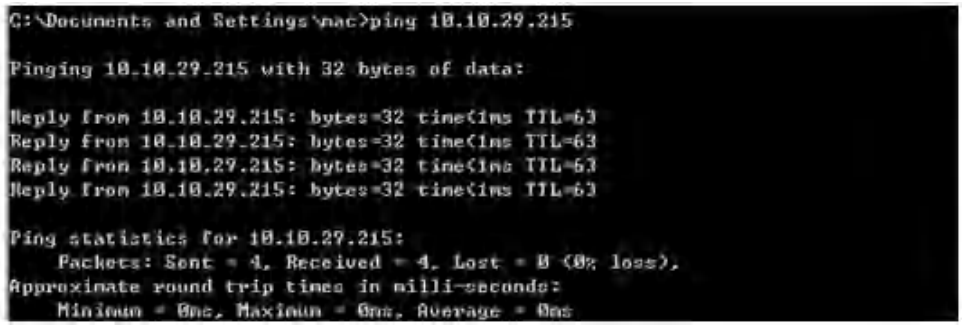


Figure 8-29 Connectivity test

Note:

If local authorization is used, different privilege levels can be configured for corresponding users, for example:

```
Switch(config)# username test1 privilege 3 password 12345678
```

If you are using RADIUS Server authorization, the server may support more privilege levels (e.g., Cisco: 0–15). You can use the command show aaa privilege mapping to view the mapping between privileges and Cisco levels 0–15. You can also use the command aaa privilege mapping to modify this mapping.

8.17 Configuring TACACS+

8.17.1 Overview

Function Introduction

Authentication verifies users before they are allowed access to the network and network services. System can use AAA authentication methods and Non-AAA authentication methods. TACACS+ Authentication is one of AAA authentication methods. TACACS+ is a distributed client/server system that secures networks against unauthorized access. TACACS+ is widely used protocol in network environments. It is commonly used for embedded network devices such as routers, modem servers, switches, etc. TACACS+ clients run on support routers and switches. Clients send authentication requests to a central TACACS+ server, which contains all user authentication and network service access information.

Principle Description

N/A

8.17.2 Configuration

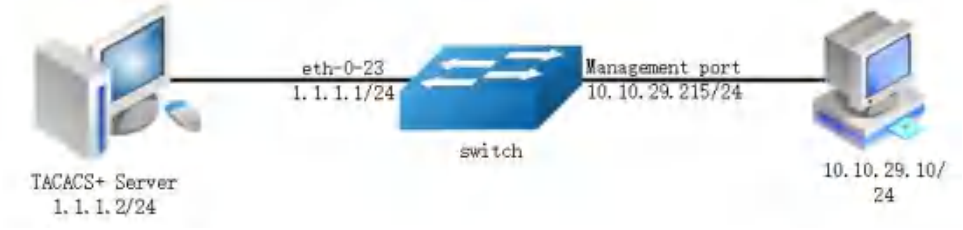


Figure 8-30 TACACS+

The figure above is the networking topology for TACACS+ authentication functions. We need one Switch and two computers for this test. One computer as TACACS+ server, it ip address of the eth0 interface is 1.1.1.2/24. Switch has TACACS+ authentication function. The ip address of interface eth-0-23 is 1.1.1.1/24. The management ip address of switch is 10.10.29.215, management port (only in-band management port) is connected the PC for test login, PC's ip address is 10.10.29.10

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable AAA

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# aaa authentication login tac-login tacacs-plus local
Switch(config)# aaa authorization exec default tacacs-plus
Switch(config)# aaa accounting exec default start-stop tacacs-plus
Switch(config)# aaa accounting commands default tacacs-plus
```

step 3 Configure tacacs+ server

```
Switch(config)# tacacs-server host 1.1.1.2 port 123 key keyname primary
```

step 4 Configure a layer 3 interface and set ip address

```
Switch(config)# interface eth-0-23
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# quit
```

step 5 set authentication mode

```
Switch(config)# line vty 0 7
Switch(config-line)#login authentication tac-login
Switch(config-line)#privilege level 4
Switch(config-line)#no line-password
```

step 6 Exit the configure mode

```
Switch(config-line)# end
```

step 7 Validation

You can use command show authentication status in switch：

```
Switch# show aaa status
aaa stats:
   Authentication enable
```

You can use command show keys in switch:

```
Switch# show aaa method-lists authentication
authen queue=AAA_ML_AUTHEN_LOGIN
   Name = default  state = ALIVE :   local
Name = tac-login  state = ALIVE :   tacacs-plus local
```

Telnet output:



Figure 8-31 Telnet connecting test

8.17.3 Application cases

Radius server configuration

Download TACACS+ server code, DEVEL.201105261843.tar.bz2.

Build the TACACS+ server.

Add username and password in configure file.

```
#!../obj.linux-2.6.9-89.29.1.elsmp-x86_64/tac_plus
id = spawnd {
    listen = { port = 49 }
    spawn = {
        instances min = 1
        instances max = 10
    }
    background = no
}
    user = aaa {
        password = clear bbb
        member = guest
    }
```

Run TACACS+ server:

```
[disciple: ~]$ ./tac_plus ./tac_plus.cfg.in -d 1
```

Use Ping command for test on PC:

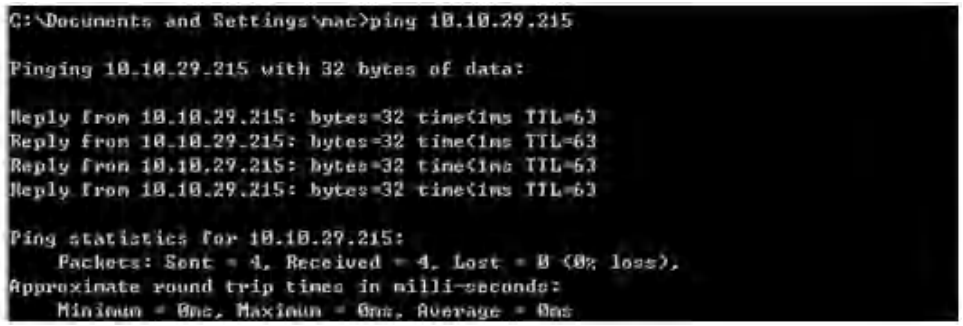


Figure 8-32 Connectivity test

8.18 Configuring Port Isolate

8.18.1 Overview

Function Introduction

Port-isolation a security feature which is used to prevent from direct l2/l3 communication among a set of ports.

It can provide a safer and more flexible network solutions by isolating the ports which in the same VLAN.

Generally, it's used as an access device for user isolation.

Principle Description

N/A

8.18.2 Configuration

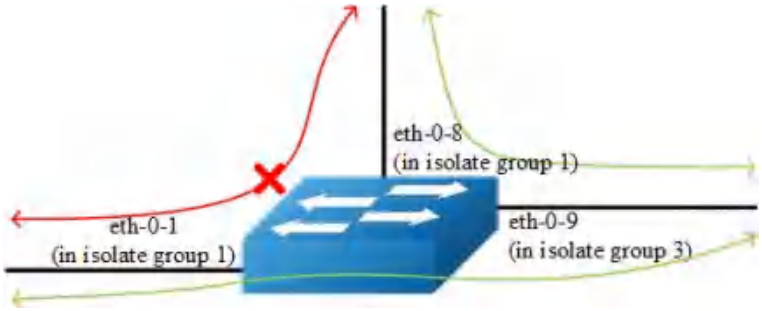


Figure 8-33 Port Isolate

The figure above is the basic topology for port-isolate.

Port 1 and port 8 are in the same isolate group 1, they are isolated. So port1 can not communicate with port 8. Port 9 is in a different isolate group 3, so port 9 can communicate with port 1 and port 8.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the port isolate mode globally

The mode "l2" means only layer 2 packets are isolated. The mode "all" means all packet are isolated include the packets forward according to layer 3 routes.

```
Switch(config)# port-isolate mode l2
```

step 3 Enter the interface configure mode and set isolate group

```
Switch(config-if)# interface eth-0-1
Switch(config-if)# port-isolate group 1
Switch(config-if)# exit

Switch(config)# interface eth-0-8
Switch(config-if)# port-isolate group 1
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# port-isolate group 3
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Use the following command to display the port isolate groups:

```
switch# show port-isolate
------------------------------------------------------------------
 Port Isolate Groups:
------------------------------------------------------------------
 Groups ID: 1
eth-0-1，eth-0-8
------------------------------------------------------------------
```

```
Groups ID: 3
eth-0-9
```

8.18.3 Application cases

N/A

8.19 Configuring DDoS

8.19.1 Overview

Function Introduction

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts of a person or people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root name servers. The term is generally used with regards to computer networks, but is not limited to this field, for example, it is also used in reference to CPU resource management.

DDoS prevent is a feature which can protect our switch from follow kinds of denial-of-service attack and intercept the attack packets.

The flowing types are supported：

- ICMP flood: attackers overwhelm the victim with ICMP packets.
- Smurf attack: attackers flood a target system via spoofed broadcast ping messages.
- SYN flood: attackers send a succession of SYN requests to a target's system.
- UDP flood: attackers send a large number of UDP packets to random ports on a remote host.
- Fraggle attack:attackers send a large number of UDP echo traffic to IP broadcast addresses, all fake source address.
- Small-packet: attackers send a large number of small packets to the system utill the resource exhaust.
- bad mac intercept: attackers send packets with same source and destination MAC address.
- bad ip equal: attackers send packets with same source and destination IP address.

Principle Description

N/A

8.19.2 Configuration

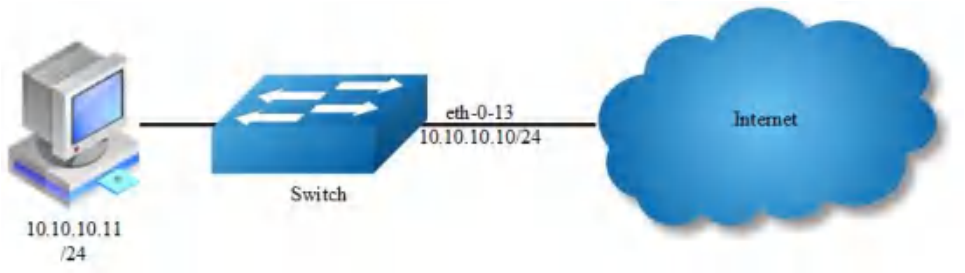


Figure 8-34 Topology for DDoS test

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set DDoS

Enable ICMP flood intercept and set the max received ICMP packet rate 100 packets per-second

```
Switch(config)# ip icmp intercept maxcount 100
```

Enable UDP flood intercept and set the max received UDP packet rate 100 packets per-second

```
Switch(config)# ip udp intercept maxcount 100
```

Enable Smurf attack intercept

```
Switch(config)# ip smurf intercept
```

Enable SYN flood intercept and set the max received SYN packet rate 100 packets per-second

```
Switch(config)# ip tcp intercept maxcount 100
```

Enable Fraggle attack intercept

```
Switch(config)# ip fraggle intercept
```

Enable Small-packet attack intercept and set the received packet length is be more than or equal to 32

```
Switch(config)# ip small-packet intercept maxlength 32
```

Enable packet source IP equals destination IP intercept

```
Switch(config)# ip ipeq intercept
```

Enable packet source MAC equals destination MAC intercept

```
Switch(config)# ip maceq intercept
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ip-intercept config
Current DDoS Prevent configuration:
============================================================
ICMP Flood Intercept          :Enable  Maxcount:500
UDP Flood Intercept           :Enable  Maxcount:500
SYN Flood Intercept           :Enable  Maxcount:500
Small-packet Attack Intercept   :Enable  Packet Length:45
Smurf Attack Intercept        :Enable
Fraggle Attack Intercept      :Enable
MAC Equal Intercept           :Enable
IP Equal Intercept           :Enable
Switch# show ip-intercept statistics
Current DDoS Prevent statistics:
============================================================
Resist Small-packet Attack packets number   : 1730
Resist ICMP Flood packets number          : 0
Resist SYN Flood packets number           : 0
Resist Fraggle Attack packets number      : 0
Resist UDP Flood packets number           : 0

Current DDoS Prevent mgmt-if statistics:
============================================================
Resist ICMP Flood packets number          : 0
Resist SYN Flood packets number           : 0
Resist Fraggle Attack packets number      : 0
Resist UDP Flood packets number           : 0
```

8.19.3 Application cases

N/A

8.20 Configuring Key Chain

8.20.1 Overview

Function Introduction

Keychain is a common method of authentication to configure shared secrets on all the entities, which exchange secrets such as keys before establishing trust with each other. Routing protocols and network applications often use this authentication to enhance security while communicating with peers.

The keychain by itself has no relevance; therefore, it must be used by an application that needs to communicate by using the keys (for authentication) with its peers. The keychain provides a secure mechanism to handle the keys and rollover based on the lifetime.

If you are using keys as the security method, you must specify the lifetime for the keys and change the keys on a regular basis when they expire. To maintain stability, each party must be able to store and use more than one key for an application at the same time. A keychain is a sequence of keys that are collectively managed for authenticating the same peer, peer group, or both. Keychain groups a sequence of keys together under a keychain and associates each key in the keychain with a lifetime.

Principle Description

N/A

8.20.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create key chain and set key

```
Switch(config)# key chain test
Switch(config-keychain)# key 1
Switch(config-keychain-key)# key-string ##test_keystring_1##
Switch(config-keychain-key)# accept-lifetime 0:0:1 1 jan 2012 infinite
Switch(config-keychain)# key 2
Switch(config-keychain-key)# key-string ##test_keystring_2##
Switch(config-keychain-key)# send-lifetime 0:0:1 2 jan 2012 infinite
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

To display the keychain configuration, use the command show key chain in the privileged EXEC mode“

```
Switch # show key chain
key chain test:
  key 1 -- text "key-string ##test_keystring_1##"
    accept-lifetime <00:00:01 Jan 01 2012> - <infinite>
    send-lifetime <always valid> - <always valid> [valid now]
  key 2 -- text "key-string ##test_keystring_2##"
    accept-lifetime <always valid> - <always valid> [valid now]
    send-lifetime <00:00:01 Jan 02 2012> - <infinite>
```

8.20.3 Application cases

N/A

8.21 Configuring Port-Block

8.21.1 Overview

Function Introduction

By default, the switch floods packets with unknown destination MAC addresses out of all ports. If unknown unicast and multicast traffic is forwarded to a protected port, there could be security issues. To prevent unknown unicast or multicast traffic from being forwarded from one port to another, you can block a port (protected or unprotected) from flooding unknown unicast or multicast packets to other ports.

Principle Description

N/A

8.21.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and block unknown unicast

```
Switch(config)# interface eth-0-1
Switch(config-if)# port-block unknown-unicast
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

To display the port-block configuration, use the command show port-block in the privileged EXEC mode：

```
Switch # show port-block interface eth-0-1
Known unicast blocked: Enabled
Known multicast blocked: Disabled
Unknown unicast blocked: Disabled
Unknown multicast blocked: Disabled
Broadcast blocked: Disabled
```

8.21.3 Application cases

N/A

8.22 Configuring Global ACL

8.22.1 Overview

Brief Introduction

System will support the configurations of global acl, when port acl, vlan/vlanif acl and global acl existed at the same time, the priority is port acl > vlan/vlanif acl> global acl.

The specific performance is: When the device receives or sends packets, the system will find the acl rule by the priority, once the rule has hit, the packets will forward immediately. If the upper acl has not hit, then it will research the next acl.

8.22.2 Configuration Examples

Global ACL Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable Global ACL and take affect until the next reload

```
Switch(config)# global-acl enable
```

step 3 Apply the policy to global acl

```
Switch(config)# service-policy global input pmap
```

step 4 Apply the access group to global acl

```
Switch(config)# access-group global input aaa
```

step 5 Validation

```
DUT1# show running-config
global-acl enable
service-policy global input pmap
access-group global input aaa
```

8.23 Configuring MACsec

8.23.1 Overview

Function Introduction

MACSec (Media Access Control Security) is a secure communication method based on the 802.1AE and 802.1X protocols for local area networks. Ensure the security of Ethernet data frames through identity authentication, data encryption, integrity verification, replay protection, and other functions to prevent devices from processing messages with security threats.

MACSec uses L2 encryption technology to provide secure data transmission for hop by hop devices, suitable for situations that require high data confidentiality, such as bank account information being stolen, tampered with, or subjected to malicious network attacks. After deploying MACsec in the network, MACSec encryption technology can ensure the secure transmission of data on intermediate transmission devices, reducing the risk of information leakage and malicious network attacks.

MACsec Principle Description

Basic Concepts

The establishment and management of MACsec secure channels, as well as the negotiation of keys used by MACsec, are the responsibility of the MKA (MACsec Key Agreement) protocol.

- CA (Secure Connectivity Association) is a secure association established and maintained by a key negotiation protocol, which is a collection of two or more members on a local area network that support MACsec and use the same key and key algorithm suite. The key used by CA members is called CAK (Secure Connectivity Association Key).

  MACsec is mainly used in peer-to-peer networking environments, that is, networking from one device's interface to another device's interface, so paired CAKs are mainly used. Two connected devices form a CA and use the same CAK.

- SA (Secure Association) is a secure relationship that ensures the secure transmission of data frames between CA members. Each SA has one or a set of encryption keys for data frames, called SAK (Secure Association Key). SAK is generated by algorithms based on CAK and is used for encrypting and decrypting data packets.

Security mechanism

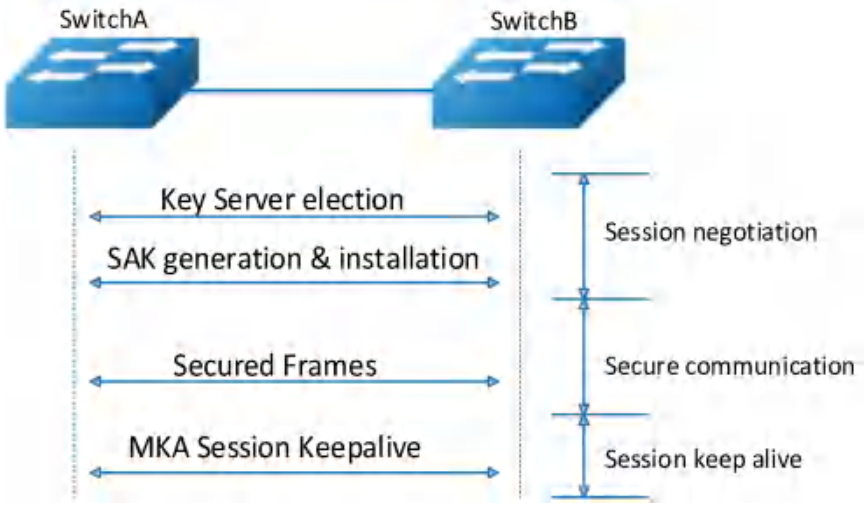
MACsec ensures the secure transmission of user business data in the local area network from the following aspects:

- Data encryption: The sender encrypts the data and transmits it in encrypted form over the local area network link. The receiver decrypts the received encrypted data before performing other processing.

- Integrity verification: The receiver performs integrity verification on the received data to determine whether the data has been tampered with. The sender calculates the Integrity Check Value (ICV) based on the entire data message and encryption algorithm, and attaches it to the message. After receiving the message, the receiver calculates the ICV based on the data message excluding the ICV and the same encryption algorithm, and compares it with the ICV in the message. If the two are the same, the message is considered complete and the verification passes; Otherwise, the verification will not pass.
- Replay protection: To prevent malicious users from launching network attacks by repeatedly sending captured data packets, the receiver will discard duplicate or out of order data packets by default. When data packets are transmitted in the network, there may be a rearrangement of the packet order. The replay protection mechanism allows data frames to have a certain degree of out of order, and these out of order frames can be legally received within the user specified window range. Frames that exceed the window will be discarded.

Sperating mechanism

The MACsec interaction process is mainly divided into three stages: session negotiation, secure communication, and session keep alive：



1. Session negotiation

Devices use configured Pre-Shared Keys (PSK) as CAK to initiate negotiation sessions through EAPOL-MKA messages. The device interface with high priority will be elected as the key server, which generates SAK for encrypting data packets based on the statically configured CAK and distributes it to the peer devices. Devices notify each other of their capabilities and various parameters required to establish a session, such as priority and whether encrypted sessions are expected, through the MKA protocol.

Users can configure the priority value of the interface. The lower the value, the higher the priority. The device interface with higher priority will be elected as the Key Server. When both devices have the same priority, the SCI (Secure Channel Identifier) value of the interface is compared. SCI consists of the MAC address and Port ID of the interface, and the interface with a lower SCI value will be selected as the key server.

2. Secure communication

The sender encrypts the data message using SAK, and the receiver decrypts the data message using SAK. Both end devices can act as both senders and receivers, and the communication process is protected by MACsec. To ensure the security of data packets, when the number of encrypted packets using the same SAK exceeds a certain amount or when the same SAK is used for more than a certain period of time, the SAK needs to be replaced.

3. Session keep alive

The MKA protocol defines an MKA session keep alive timer to specify the timeout period for MKA sessions. After successful MKA session negotiation, both devices will confirm the existence of the connection through interactive MKA protocol messages. After receiving the MKA protocol message from the other end, the device starts the timer. If an MKA protocol message is received from the other end within the timeout period, restart the timer. If no MKA protocol message is received from the other end within the timeout period, it is considered that the connection is no longer secure, the other end device is deleted, and MKA negotiation is resumed.

Parameter Introduction

CKN (secure Connectivity Association Key Name): The name of CAK, which must be configured with the same CKN on the interfaces of both devices.

CAK（secure Connectivity Association Key）： The same CAK must be configured on the interfaces of both devices, which is not directly used for encrypting data packets, but is derived from it and CKN to use SAK for data encryption.

SAK (Secure Association Key): Generated by a key server based on CAK and CKN, used for encrypting and decrypting data messages.

KEK (Key Encryption Key): Both devices generate the same KEK based on the same CAK and CKN, which is used for encrypting and decrypting SAK to prevent leakage during the transmission process of publishing SAK.

ICV (Integrity Check Value): The sender calculates the ICV based on the message and places it at the end of the message. The receiver uses the same algorithm to calculate the ICV and compare it with the ICV carried in the message. If these two ICVs are the same, it means that the message has not been modified and the verification has passed; Otherwise, the message will be considered modified and discarded.

ICK (ICV Key, Integrity Verification Key): Both devices generate the same ICK based on the same CAK and CKN, which is used to calculate the ICV of MKA protocol messages. ICK is only required for ICV calculation of MKA protocol messages, and ICK is not required for ICV calculation of data messages.

mka policy: There exists a default MKA policy named "default-policy", with parameter values corresponding to the default values on the interface, and this policy cannot be deleted or modified. Use "show mka policy" to display the existing policies.

confidentiality-offset: The number of bytes offset from the beginning of the user data frame before encryption begins. The protocol provides three offsets of 0, 30, and 50 for the user to use (default is 0, indicating encryption of the entire data frame). If the configurations on both ends are inconsistent, the key server configuration shall prevail.

icv-mode: The ICV verification mode can be configured as normal or integrity-only. normal： Perform both data encryption and integrity verification (default is normal). integrity-only： Only perform integrity verification without encrypting data.

replay-protection: configurable to enable/disable replay-protection and replay-protection window size. Replay-protection is enabled by default. Assuming the configured replay protection window-size is a, if a message with sequence number X is received, the sequence number of the next allowed received message must be greater than or equal to X+1-a (the default replay protection window-size is 0 data frames).

validation-mode: Supports 2 validation modes: check:which means only performing verification without discarding illegal data frames (default is check mode). strict:which means verifying received data frames and discarding illegal data.

cipher-suite: Both ends encrypt, verify, and recover data frames based on the configured encryption suite algorithm. The currently supported encryption suites are gcm-aes-128 and gcm-aes-256 (default is gcm-aes-128).

mka priority: Used for negotiating and electing key servers, with lower values indicating higher priority (default value is 0).

mka psk (mka Pre-shared key): Configure cipher suite, CKN, and CAK, with no default PSK.

macsec include-sci SCI (Secure Channel Identifier): Used to identify the source of a message, consisting of an interface MAC address and an interface index. Whether the MACsec frame header of both devices contains SCI needs to be consistent. Due to the current device only supporting device to device MACsec, where the interfaces of both devices correspond one-to-one, SCI does not need to be included in the MACsec frame header. By default, the MACsec frame header contains SCI.

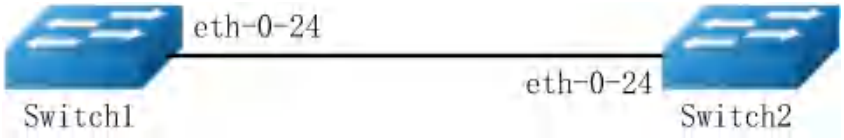
mka apply: by default, there is no policy applied on the port.

mka enable: By default, it is not enabled.

macsec desire (enable MACsec protection): By default, the interface does not provide MACsec protection for data frames sent.

8.23.2 Configuration Examples

Topology



Configuration steps

Configuration on Switch1

step 1 Enter the configure mode

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)#
```

step 2 Configure the IP address of the interface to enable two switches to communicate with each other：

```
Switch1(config)# interface eth-0-24
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.11.1/24
Switch1(config-if)# exit
```

step 3 Configure mka policy (system has default policy)

```
Switch1(config)# mka policy mkatest
Switch1(config-mka-policy)# confidentiality-offset 30
Switch1(config-mka-policy)# exit
Switch1(config)#
```

step 4 mkaConfigure interface mka and enable it

```
Switch1(config)# interface eth-0-24
Switch1(config-if)# mka apply policy mkatest
Switch1(config-if)# mka psk ckn 112233 cak simple abcde1
Switch1(config-if)# macsec desire
Switch1(config-if)# mka priority 11
Switch1(config-if)# mka enable
```

Configuration on Switch2

step 5 Complete a similar configuration on Switch2, with mka priority remaining at the default value of 0 (the lower the value, the higher the priority)

```
Switch2(config)# interface eth-0-24
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.11.2/24
Switch2(config-if)# exit
Switch2(config)# mka policy mkatest
Switch2(config-mka-policy)# confidentiality-offset 30
Switch2(config-mka-policy)# exit
Switch2(config)# interface eth-0-24
Switch2(config-if)# mka apply policy mkatest
Switch2(config-if)# mka psk ckn 112233 cak simple abcde1
Switch2(config-if)# macsec desire
Switch2(config-if)# mka enable
```

step 6 Validation

Show the negotiation results of the MKA session:

```
Switch1# show mka session interface
Interface eth-0-24:
 Tx-SCI          : 0001 0203 0606 0018
 Priority        : 11
 Capability      : 3
 CKN for participant : 1122 3300 0000 0000 0000 0000 0000 0000
  Key server              : no
  MI (MN)                 : 40C0 4080 0080 0000 C040 0080 (430)
  Principal actor         : yes
  MKA session status      : succeeded
  Icv mode                : normal
  Confidentiality offset  : 30 bytes
  Current SAK status      : rx & tx
  Current SAK AN          : 1
  Current SAK Cipher Suite    : gcm-aes-128
  Current SAK KI (KN)     : 0C44 40C8 044C 0404 C440 C484 0000 0002 (2)
  Previous SAK status     : rx
  Previous SAK AN         : 0
  Previous SAK Cipher Suite   : gcm-aes-128
  Previous SAK KI (KN)    : 1000 1040 5011 4000 1100 4000 0000 0001 (1)
  Peer list:
  MI                  MN      Priority Capability Rx-SCI

  0C44 40C8 044C 0404 C440 C484 428      0      3       D85B 2200 0401 0
018

Switch1#
```

```
Switch2# show mka session interface
Interface eth-0-24:
 Tx-SCI          : D85B 2200 0401 0018
 Priority        : 0
 Capability      : 3
 CKN for participant : 1122 3300 0000 0000 0000 0000 0000 0000
  Key server              : yes
  MI (MN)                 : 0452 0202 4042 5216 1616 5614 (444)
  Principal actor         : yes
  MKA session status      : succeeded
  Icv mode                : normal
  Confidentiality offset  : 30 bytes
  Current SAK status      : rx & tx
  Current SAK AN          : 0
  Current SAK Cipher Suite    : gcm-aes-128
  Current SAK KI (KN)     : 0452 0202 4042 5216 1616 5614 0000 0001 (1)
  Previous SAK status     : N/A
  Previous SAK AN         : N/A
  Previous SAK Cipher Suite   : N/A
  Previous SAK KI (KN)    : N/A
  Peer list:
  MI                  MN      Priority Capability Rx-SCI

  A704 20A2 A203 0287 27A0 A182 414      11      3       0001 0203 0606 0
018
```

Switch2

9 Device Management Configuration Guide

9.1 Configuring STM

9.1.1 Overview

Function Introduction

Switch Table Management (STM) is used to configure system resources in the switch to optimize support for specific features, depending on how the switch is used in the network.

You can select a profile to provide maximum system usage for some functions; for example, use the default profile to balance resources and use vlan profile to obtain max MAC entries.

To allocate ternary content addressable memory (TCAM) resources for different usages, the switch STM profile prioritize system resources to optimize support for certain features. You can select STM templates to optimize these features:

- layer2: The VLAN template supports the maximum number of unicast MAC addresses. It would typically be selected for a Layer 2 switch.
- layer3: The routing template maximizes system resources for unicast routing, typically required for a router or aggregator in the center of a network.
- ipv6: The ipv6 template， support the ipv6 functions.
- mpls: The mpls template supports the maximum number of MPLS/MAC entries.
- default: The default template gives balance to all functions.

When users configured a profile mode which is not exist in the next reboot image, then default hardware configure will be used when system up with the next image. The hardware configure may be different from the default profile.

Principle Description

N/A

9.1.2 Configuration

Follow these guidelines when selecting and configuring STM profiles.

You must reload the switch for the configuration to take effect.

Use the "stm prefer layer2" global configuration command only on switches intended for Layer 2 switching with no routing.

Do not use the layer3 profile if you do not have routing enabled on your switch. The stm prefer layer3 global configuration command prevents other features from using the memory allocated to IPv4 unicast routing in the routing profile.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set STM profile(use layer3 for example)

```
Switch(config)# stm prefer layer3
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

This is an example of an output display for route template:

```
Switch# show stm prefer
Current profile is: default
```

```
Next profile is: layer3
Current profile info is:
  number of vlan instance                    : 4094
  number of vlan stats                       : 256
  number of unicast mac address              : 122880
  number of multicast mac address            : 2048
  number of blackhole mac address            : 128
  number of max applied vlan mapping         : 3072
  number of bfd sessions                     : 256
  number of CFM local&remote MEPs            : 1024
  number of CFM lm                           : 256
  number of CFM lck                          : 24
  number of G8031 groups                     : 256
  number of G8032 rings                      : 128
  number of G8032 member ports               : 256
  number of mac based vlan class             : 512
  number of ipv4 based vlan class            : 448
  number of ipv4 subnet based vlan class     : 128
  number of ipv6 based vlan class            : 32
  number of protocol based vlan class        : 7
  number of dot1x mac based                  : 512
  number of unicast ipv4 host routes         : 16384
  number of unicast ipv4 indirect routes     : 61440
  number of unicast ipv4 policy based routes : 1024
  number of unicast ipv6 host routes         : 4096
  number of unicast ipv6 indirect routes     : 4096
  number of unicast ecmp groups              : 240
  number of unicast ip tunnel peers          : 8
  number of multicast ipv4 routes            : 2048
  number of multicast ipv4 member            : 4096
  number of multicast ipv6 routes            : 256
  number of multicast ipv6 member            : 512
  number of mvr entries                      : 2048
  number of mvr6 entries                     : 256
  number of ipv4 source guard entries        : 2048
  number of ipv6 source guard entries        : 1024
  number of ingress security acl flow entries : 5949
  number of ingress security acl flow stats  : 5949
  number of ingress qos flow entries         : 6126
  number of ingress qos flow stats           : 6126
  number of ingress telnet flow entries      : 64
  number of ingress ssh flow entries         : 64
  number of ingress worm anti-attack flow entries : 32
  number of ingress ipfix flow entries       : 2048
  number of ingress udf acl flow entries     : 1024
  number of ingress udf acl flow stats       : 1024
  number of ingress copp flow entries        : 4072
  number of ingress copp flow stats          : 4072
  number of egress security acl flow entries : 1901
  number of egress security acl flow stats   : 1901
  number of ipfix cache                      : 65536
  number of ifit flow                        : 1024
  number of ifit flow acl                    : 0
```

step 5 Reboot the device

```
Switch# reload
```

9.1.3 Application cases

N/A

9.2 Configuring syslog

9.2.1 Overview

Function Introduction

The system message logging software can save messages in a log file or direct the messages to other devices. The system message logging facility has these features:

- Provides you with logging information for monitoring and troubleshooting.
- Allows you to select the types of logging information that is captured.
- Allows you to select the destination of the captured logging information.

By default, the switch logs normal but significant system messages to its internal buffer and sends these messages to the system console. You can specify which system messages should be saved based on the type of the severity level. The messages are time-stamped to enhance real-time debugging and management.

You can access the logged system messages using the switch command-line interface (CLI) or by saving them to a properly configured log server. The switch software saves the log messages in an internal buffer that can store up to 1000 messages. You can monitor the system messages remotely by accessing the switch through Telnet or the console port, or by viewing the logs on a log server.

Principle Description

Terminology:

Terminology	Description
Logging	Current logging configuration
Show	Show logging configuration
Levels	Severity level information
Enable	Enable write log to local file
Disable	Disable write log to local file

System Message Log Facility Types：

Facility Name	Definition
kern	kernel messages
user	random user-level messages
mail	mail system
daemon	system daemons
auth	security/authorization messages
syslog	messages generated internally by syslogd
lpr	line printer subsystem
news	network news subsystem
uucp	UUCP subsystem

cron	clock daemon
authpriv	security/authorization messages (private)
ftp	ftp daemon

Severity Level Definitions:

Severity Level	Definition
emergency	system is unusable
alert	action must be taken immediately
critical	critical conditions
error	error conditions
warning	warning conditions
notice	normal but significant condition
information	Informational
debug	debug-level messages

9.2.2 Configuration

Configuring Logging server



Figure 9-1 syslog server

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable logging server and set the attributes

```
Switch(config)# logging server enable
Switch(config)# logging server address 1.1.1.1
Switch(config)# logging server address 2001:1000::2
Switch(config)# logging server severity debug
Switch(config)# logging server facility mail
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show logging
Current logging configuration:
```

```
============================================================
logging buffer 500
logging timestamp bsd
logging file enable
logging level file warning
logging level module debug
logging server enable
logging server severity debug
logging server facility mail
logging server address 1.1.1.1
logging server address 2001:1000::2
logging alarm-trap enable
logging alarm-trap level middle
logging merge enable
logging merge fifo-size 1024
logging merge timeout 10
logging operate disable
```

Configuring Logging Buffer Size

By default, the number of messages to log to the logging buffer is 500. If desired, you can set the number between 10 and 1000.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the logging Buffer Size

```
Switch(config)# logging buffer 700
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show logging
Current logging configuration:
============================================================
logging buffer 700
logging timestamp bsd
logging file enable
logging level file warning
logging level module debug
logging server enable
logging server severity debug
logging server facility mail
logging server address 1.1.1.1
logging alarm-trap enable
logging alarm-trap level middle
logging merge enable
logging merge fifo-size 1024
logging merge timeout 10
logging operate disable
```

The following is the information of logging server:

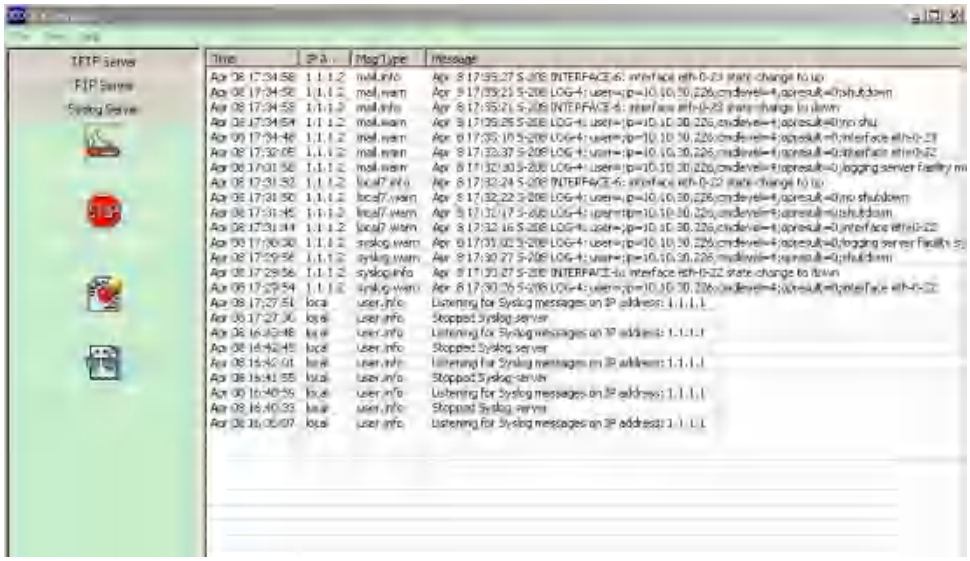


Figure 9-2 syslog on server

You can use command to check showing Logging Information. When configuring the syslog Servers, make sure the cables is linked correctly and two computers can ping each other. Before you can send the system log messages to a log server, you must configure Syslog Software, at the end you can see the log from your software.

9.2.3 Application cases

N/A

9.3 Configuring mirror

9.3.1 Overview

Function Introduction

Mirror function can send one or more copies of packets which are passing through the ports/vlans or sending and receiving by CPU to one or more specified destination ports. It can also send the copies to the CPU and keep in memory or flash files.

The copies of the packets are used for network analyze. The mirror function does not affect the original network traffic.

Principle Description

The following describes concepts and terminology associated with mirror configuration:

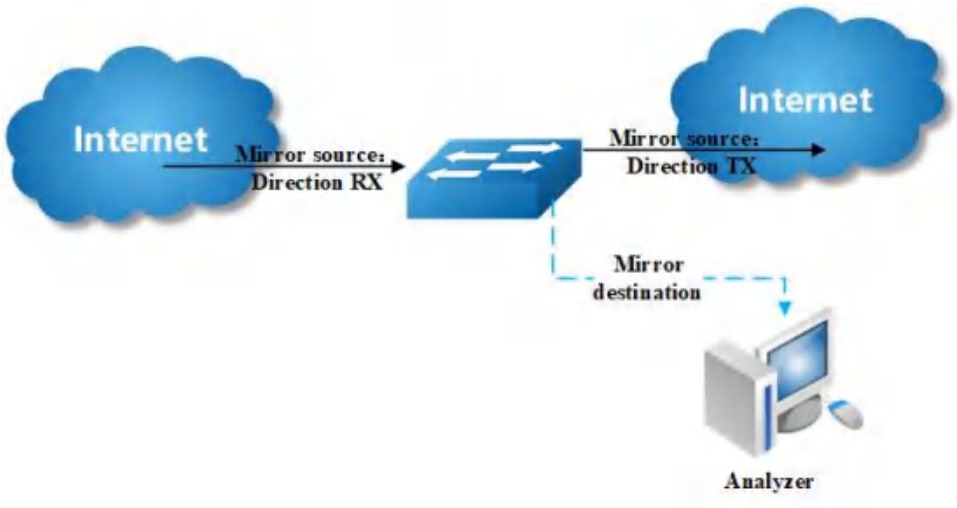


Figure 9-3 Mirror

1.Mirror session

A mirror session is an association of a mirror destination with one or more mirror source. The mirror destination and mirror source will describe later.

The device supports up to 3 mirror sessions.

Mirror sessions do not interfere with the normal operation of the switch. However, an oversubscribed mirror destination, for example, a 10-Gbps port monitoring a 100-Gbps port, results in dropped or lost packets.

2.Mirror direction

The device supports to set the direction of the mirror source, there are 3 options for choose: TX/RX/BOTH.

Receive (RX) mirror: The goal of receive (or ingress) mirror is to monitor as much as possible packets received by the source interface or VLAN before any modification or processing is performed by the switch. A copy of each packet received (except these packets: BPDU, LACPDU, BMGPDU, packets have been discarded by IP-MAC binding check for Vlan_based mirror, CRC error packets for both Port_based and vlan_based mirror) by the source is sent to the destination port for that mirror session. You can monitor a series or range of ingress ports or VLANs in a mirror session. Packets that are modified because of routing are copied without modification; that is, the original packet is copied. Packets that are modified because of quality of service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied with modification. Packets that are modified because of VLAN translation or VLAN classification is copied with the modification. Some features that can cause a packet to be dropped during receive processing have no effect on mirror, the destination port can receive a copy of the packet even if the actual incoming packet is dropped. These features include ingress ACL, VLAN's ingress filter, MAC filter, STP, VLAN tag control, port security, unknown routing packets.

Transmit (TX) mirror: The goal of transmit (or egress) mirror is to monitor as much as possible packets sent by the source interface after all modification and processing is performed by the switch. A copy of each packet (except these packets: packets from CPU port for Vlan_based mirror, mirroring packets for both Port_based and vlan_based mirror) sent by the source is sent to the destination port for that mirror session. Some features that can cause a packet to be dropped during transmit processing might have affect on mirror.

Both: In a mirror session, you can monitor a single port for both received and sent packets.

3.Mirror source

The Mirror source is the original traffic of the network. The types of source are described as following:

Source port: A source port is a layer2 or layer 2 interface which need to be monitored. A physical port or link agg port can be a source port. The member of link agg port is not supported to be a mirror source.

Source VLAN: A source vlan is a vlan which need to be monitored. User should create a vlan interface before set a vlan as mirror source.

CPU:User can set CPU as mirror source to monitor the packets send to or receive from the CPU. The copies of packets send to the mirror destination are before cpu-traffic-limit process. Only session 1 support CPU as mirror source currently.

4.Mirror destination

Mirror function will copy the packets and sent the copies to the mirror destination.

The types of destination are described as following:

Local destination port: The destination port should be a physical port or link agg port, member of link agg port is not supported. The destination port has these characteristics:

- It must reside on the same switch as the source port.
- It should not be in "shutdown" state
- It can participate in only one mirror session at a time (a destination port in one mirror session cannot be a destination port for a second mirror session).
- It cannot be a source port.
- The port does not transmit any traffic except that required for the mirror session.
- It does not participate in spanning tree while the mirror session is active.
- When it is a destination port, all other normal system function of this port should not work until mirror destination configure disabled on this port.
- No address learning occurs on the destination port.
- The real statues of the speed/duplex might not coincide with the values which are displayed.

Multi-destination: The device supports to use a group of destination ports to receive several copies of the traffic. The characteristics of each member in the group of destination ports are same as single destination port.

Remote destination： A remote mirror destination is a remote destination vlan, which has a specified out-going port. The copies of the packets should send to the specified port and add the tag of the remote vlan. A remote destination has these characteristics:

- It is a vlan with a specified out going port.
- The remote VLAN range should be 2 to 4094. If the VLAN isn't created in system, user can not configure this VLAN as mirror remote vlan.
- The out going port should be a physical port. User should manually check if the out going port can transfer mirrored packets.
- Monitor traffic packets are inserted a tag with the remote VLAN ID and directed over the specified out going port to the mirror destination session device.
- It is recommended to configure remote mirror's destination port as switch port. Users should add the destination port to the remote vlan otherwise the mirrored packet can not be transmitted out.

CPU destination： send the copies of packet to the CPU of current device. If there is no analyzer available, user can use CPU as mirror destination and save the result for user or developers analyze packets.

You can analyze network traffic passing through ports or vlans by using mirror function to send a copy of the traffic to another port on the switch that has been connected to a Switch Probe device or other Remote Monitoring (RMON) probe or security device. However, when there is no other monitoring device for capturing packets, normal mirror destination to ports doesn't work. So we can set CPU as mirror destination to send a copy of the traffic to CPU for storing packets. It supports the cli to display the packets of mirror CPU and write the packets in a text file. It is a very functional debug tool. Mirror does not affect the switching of network traffic on source ports or source vlans; a copy of the packets received or sent by the source interfaces are sent to the destination CPU. The cpu-traffic-limit rate can be configured. CPU can participate as a destination in only one mirror session.

9.3.2 Configuration

Configuring Local port mirror

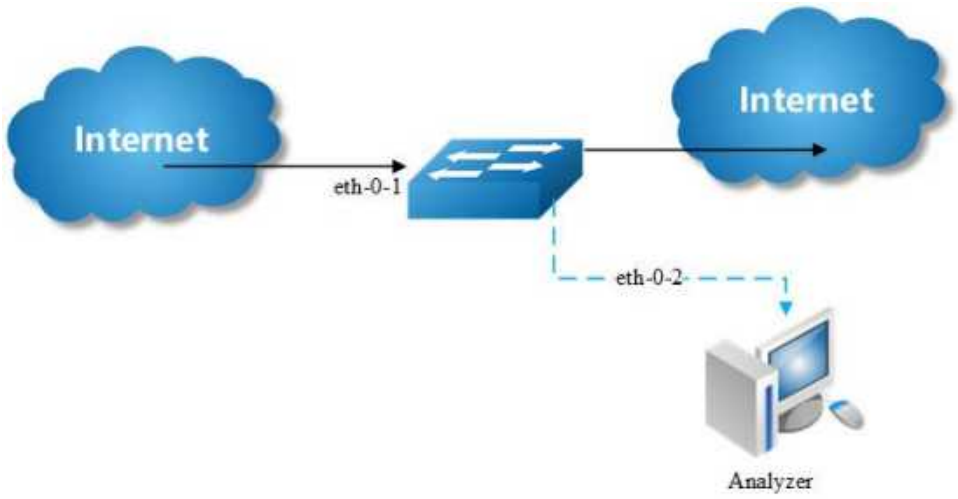


Figure 9-4 port Mirror

Copy the packets of eth-0-1 and send them to eth-0-2

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination of mirror

```
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# monitor session 1 destination interface eth-0-2
```

step 3 Set the source of mirror

```
Switch(config)# monitor session 1 source interface eth-0-1 both
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show monitor session 1
Session  1
----------
Status          : Valid
Type            : Local Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           : eth-0-1
Source VLANs    :
 Receive Only   :
 Transmit Only  :
 Both           :
Destination Port  : eth-0-2
```

Configuring local vlan mirror

Copy the packets from vlan 10 and send them to eth-0-2

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination of mirror

```
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# monitor session 1 destination interface eth-0-2
```

step 3 Enter the vlan configure mode and create a vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# exit
```

step 4 Create a vlan interface

```
Switch(config)# interface vlan10
Switch(config-if)# exit
```

step 5 Set the source of mirror

```
Switch(config)# monitor session 1 source vlan 10 rx
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show monitor session 1
Session  1
----------
Status          : Valid
Type            : Local Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           :
Source VLANs    :
 Receive Only   : 10
 Transmit Only  :
 Both           :
Destination Port  : eth-0-2
```

Configuring CPU as mirror source

Copy the packets from or to CPU and send them to eth-0-2

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination of mirror

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
```

```
Switch(config-if)# exit
Switch(config)# monitor session 1 destination interface eth-0-2
```

step 3 Set the source of mirror

```
Switch(config)# monitor session 1 source cpu both
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
DUT1# show monitor session 1
Session  1
----------
Status          : Valid
Type            : Cpu Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           : cpu
Source VLANs    :
 Receive Only   :
 Transmit Only  :
 Both           :
Destination Port  :eth-0-1
```

Configuring Multi-destination Mirror

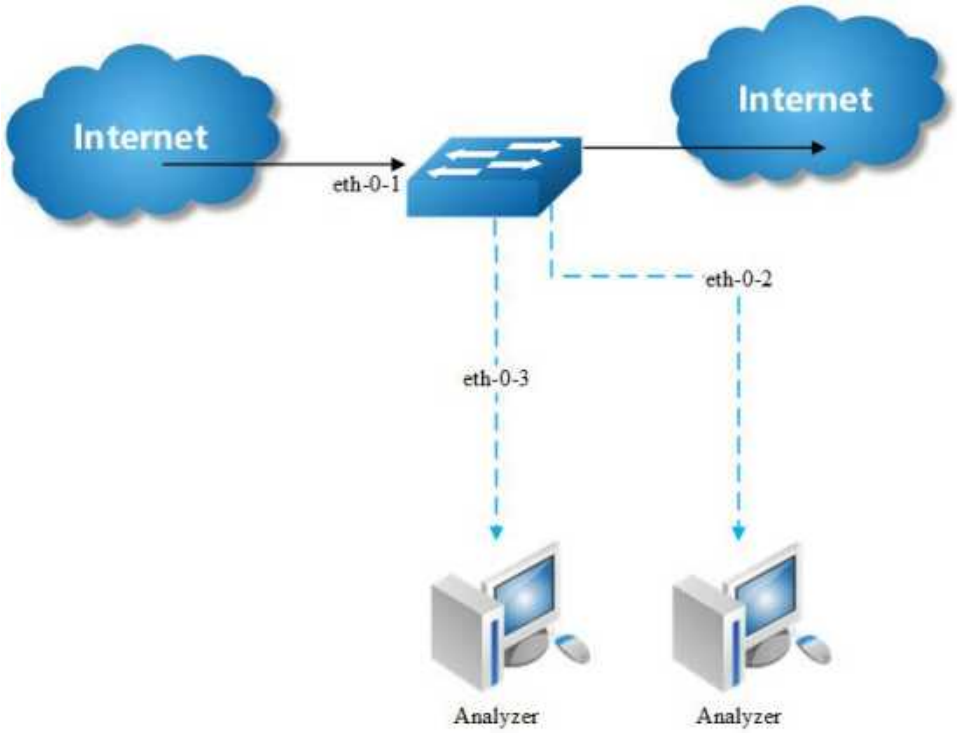


Figure 9-5 Multi-destination Mirror

Copy the packets of eth-0-1 and send them to eth-0-2 and eth-0-3

The rules of mirror source are same as single destination port. The following case use source port for example.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination group of mirror

```
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-3
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# monitor session 1 destination group 1
Switch(config-monitor-d-group)# member eth-0-2
Switch(config-monitor-d-group)# member eth-0-3
Switch(config-monitor-d-group)# exit
```

step 3 Set the source of mirror

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# monitor session 1 source interface eth-0-1
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Session  1
----------
Status          : Valid
Type            : Local Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           : eth-0-1
Source VLANs    :
 Receive Only   :
 Transmit Only  :
 Both           :
Destination Port  : eth-0-2 eth-0-3
```

Configuring Remote Mirror

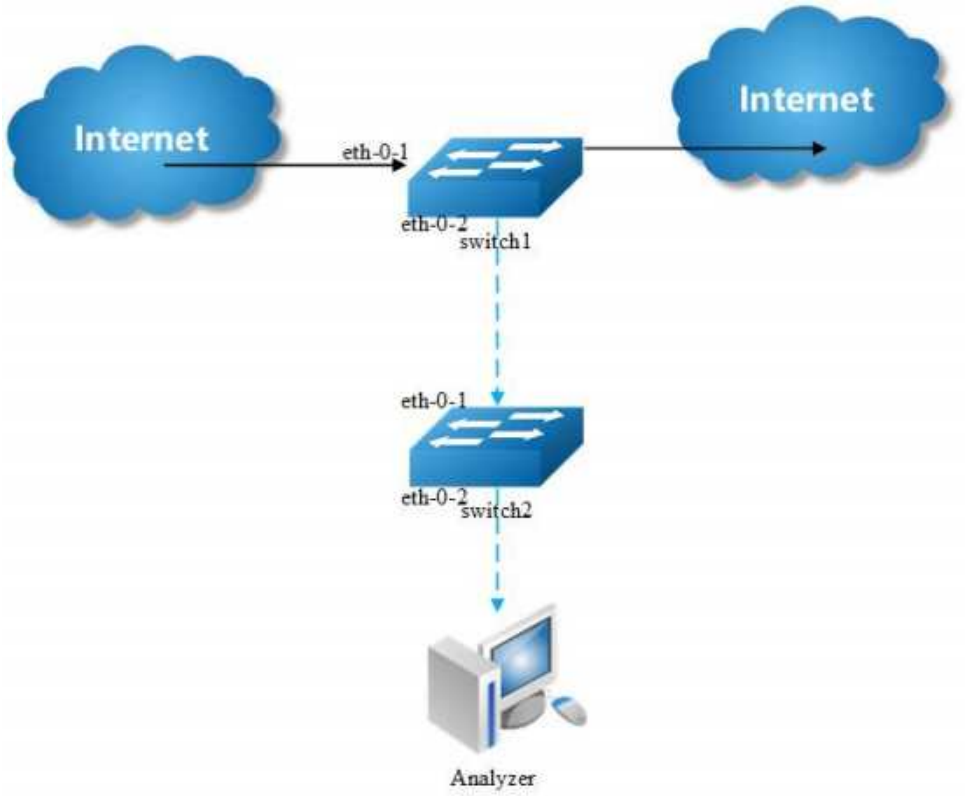


Figure 9-6 Remote Mirror

If local device cannot connect to an analyzer directly, User can choose remote mirror to send the copies of packets with specified vlan tag.

The remote device can pick out the packets with this vlan for analyze.

The following example copies the packets form Switch1's eth-0-1, and send them to Switch2 via Switch1's eth-0-2. Switch2 sends these packets to the analyzer.

The configuration of Switch1:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination of mirror

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 15
Switch(config-vlan)# exit
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# exit

Switch(config)# monitor session 1 destination remote vlan 15 interface eth-0-2
```

step 3 Set the source of mirror

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config)# monitor session 1 source interface eth-0-1 both
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
SwitchA# show monitor session 1
Session  1
----------
Status          : Valid
Type            : Remote Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           : eth-0-1
Source VLANs    :
 Receive Only   :
 Transmit Only  :
 Both           :
Destination Port  : eth-0-2
Destination remote VLAN : 15
```

The configuration of Switch2:

Use these methods on Switch2 to send packets to analyzer via eth-0-2

method 1: use vlan 15 as mirror source，eth-0-2 as mirror destination

```
Switch # configure terminal
Switch (config)# vlan database
Switch (config-vlan)# vlan 15
Switch (config-vlan)# exit

Switch (config)# interface vlan15
Switch (config-if)# exit

Switch (config)# interface eth-0-2
Switch (config-if)# no shutdown

Switch (config)# interface eth-0-1
Switch (config-if)# no shutdown
Switch (config-if)# switchport mode trunk
Switch (config-if)# switchport trunk allowed vlan add 15
Switch (config-if)# exit

Switch (config)# monitor session 1 destination interface eth-0-2
Switch (config)# monitor session 1 source vlan 15 rx
Switch (config)# end
```

method 2: add both ports in to the same vlan (15), and make the packet flood in this vlan

```
Switch# configure terminal

Switch(config)# no spanning-tree enable
```

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 15
Switch(config-vlan)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 15

Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# exit
```

In this configuration vlan tag is stripped because eth-0-2 is access port.

method 3: flood in vlan and keep vlan tag 15

If user needs to keep the vlan tag 15, eth-0-2 should be trunk port: (other configurations are same as method 2)

```
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
```

Configuring CPU Mirror Dest

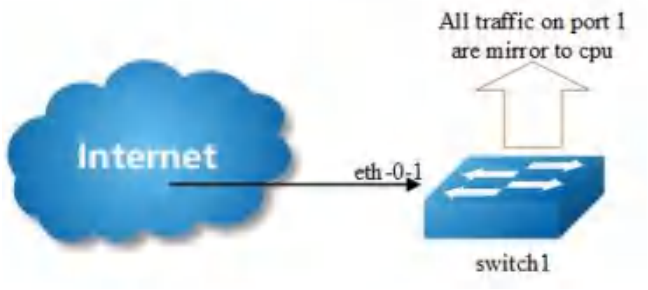


Figure 9-7 Mirror to cpu

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the destination of mirror

```
Switch(config)# monitor session 1 destination cpu
```

Set the buffer size and to cpu rate:

```
Switch(config)# monitor cpu set packet buffer 100
Switch(config)# cpu-traffic-limit reason mirror-to-cpu rate 128
```

step 3 Set the source of mirror

```
Switch(config)# monitor session 1 source interface eth-0-1 both
```

step 4 Exit the configure mode

```
Switch(config)# end
```

Optional steps

Enable or disable to write the packets in to the flash files.

```
Switch#  monitor cpu capture packet start
Switch#  monitor cpu capture packet stop
```

Exchange the files from *.txt to *.pcap

```
Switch# pcap convert flash:/mirror/MirCpuPkt-2016-02-05-18-31-13.txt flash:/MirCpuPkt-2016-02-05.pcap
```

Set the action after the packet buffer is exceeded: “drop” means discard the latest packet; “replace” means discard the oldest packet.

```
Switch(config)# monitor cpu capture strategy drop
Switch(config)# monitor cpu capture strategy replace
```

step 5 Validation

This example shows how to set up a mirror session, session 1, for monitoring source port traffic to a destination cpu. You can use show monitor session to see the configuration.

```
Switch# show monitor session 1
DUT1# show monitor session 1
Session   1
----------
Status          : Valid
Type            : Cpu Session
Source Ports    :
 Receive Only   :
 Transmit Only  :
 Both           : eth-0-1
Source VLANs    :
 Receive Only   :
 Transmit Only  :
 Both           :
Destination Port  : cpu
```

This example shows how to display the mirror cpu packets

```
Switch# show monitor cpu packet all
-----------------show all mirror to cpu packet info-----------------
packet: 1
Source port: eth-0-1
MACDA:264e.ad52.d800, MACSA:0000.0000.1111
vlan tag:100
IPv4 Packet, IP Protocol is 0
IPDA:3.3.3.3, IPSA: 10.0.0.2
Data length: 47
Data:
 264e ad52 d800 0000 0000 1111 8100 0064
 0800 4500 001d 0001 0000 4000 6ad9 0a00
 0002 0303 0303 6365 6e74 6563 796f  75
```

This example shows how to display the mirror buffer size:

```
Switch# show monitor cpu packet buffer
--------------------show packet buffer size ---------------------
The mirror-to-cpu packet buffer size of user set is: 100
```

This example shows how to display the mirror cpu traffic-limit rate:

```
Switch# show cpu traffic-limit | include mirror-to-cpu
mirror-to-cpu           128        0
```

This example shows how to display the files of the flash:

```
Switch# ls flash:/mirror
Directory of flash:/mirror

total 8
-rw-r-----  1 2287 Dec 23 01:16 MirCpuPkt-2016-12-23-01-15-54.txt
-rw-r-----  1 2568 Jan  3 11:41 MirCpuPkt-2017-01-03-11-41-33.txt
14.8T bytes total (7.9T bytes free)

Switch# more flash:/mirror/ MirCpuPkt-2017-01-03-11-41-33.txt
sequence  srcPort
1        eth-0-1
++++++++1483443444:648884
8c 1d cd 93 51 00 00 00 00 00 11 11 08 00 45 00
00 26 00 01 00 00 40 00 72 d0 01 01 01 01 03 03
03 03 63 65 6e 74 65 63 79 6f 75 63 65 6e 74 65
63 79 6f 75
--------
sequence  srcPort
2        eth-0-1
++++++++1483443445:546440
8c 1d cd 93 51 00 00 00 00 00 11 11 08 00 45 00
00 26 00 01 00 00 40 00 72 d0 01 01 01 01 03 03
03 03 63 65 6e 74 65 63 79 6f 75 63 65 6e 74 65
63 79 6f 75
```

This example shows how to display the files of the flash. *.pcap files can open with packets analyzer applications such as wireshark. Please referenc to the "ftp" and "tftp" part to download the files.

```
Switch#ls flash:/mirror
Directory of flash:/mirror

total 12
-rw-r-----  1 2287 Dec 23 01:16 MirCpuPkt-2016-12-23-01-15-54.txt
-rw-r-----  1 2568 Jan  3 11:41 MirCpuPkt-2017-01-03-11-41-33.txt
-rw-r--r--  1  704 Jan  3 13:07 test.pcap
14.8T bytes total (7.9T bytes free)
```

This example shows how to display the actions after the buffer is full

```
Switch# show monitor cpu capture strategy
The capture strategy of cpu mirror is: replace (add new packet and remove oldest
 packet when buffer is full)
```

9.3.3 Application cases

N/A

9.4 Configuring Device Management

9.4.1 Overview

Function Introduction

User can manage the switch through the management port. The switch has two management ports: an Ethernet port and a console port.

9.4.2 Configuration

Configuring console port for management

The default console parameters of switch are:

- Baud rate default is 115200.
- Data bits default is 8.
- Stop bits default is 1.
- Parity settings default is none.

Before you can assign switch information, make sure you have connected a PC or terminal to the console port, and configured the PC or terminal software parameters to match the default console port parameters. After login in the switch, you can modify the console parameters.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter line configuration mode and set the console speed

```
Switch(config)# line console 0
Switch(config-line)# speed 19200
```

step 3 Exit the configure mode

```
Switch(config-line)# end
```

step 4 Validation

After the above setting, console port parameter has been changed, and the PC or terminal can't configure the switch by console port. You must update PC or terminal console speed from 115200 to 19200 to match the new console parameter and can continue configure the switch by console port.

Configuring out band Ethernet port for management

In order to manage device by out band Ethernet port, you should configure management ip address first by console port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure switch management address

IPv4 & IPv6 are both supported, for example:

```
Switch(config)# management ip address 10.10.38.106/24
Switch(config)# management ipv6 address 2001:1000::1/96
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show management ip address
 Management IP address is:  10.10.38.106/24
 Gateway: 0.0.0.0

Switch # show management ipv6 address
```

```
Management IPv6 address is:  2001:1000::1/96
Gateway: ::
```

Configuring Temperature

The switch supports temperature alarm management. You can configure three temperature thresholds: low, high and critical. When switch temperature is lower than low threshold or higher than higher threshold, the switch will be alarm. If the switch temperature is higher than critical threshold, the switch will cut off its power automatically.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configuring temperature threshold

5℃ for low; 70℃ for high; 90℃ for critical.

```
Switch(config)# temperature 5 70 90
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show environment
-------------------------------------------------------
Sensor status (Degree Centigrade):
Index Temperature  Lower_alarm  Upper_alarm Critical_limit
1    50        5         70        90
```

Configuring Fan

The switch supports to manage fan automatically. If the fan is fail or the fan tray is absent, the switch will be alarm. And if the fan tray supports speed-adjust, the switch can adjust the fan speed depending on the real-time temperature. The switch has three temperature thresholds: Tlow=50, Thigh=65 and Tcrit=80 Celsius scales. If Temperature<Tlow, the fan will stall; if Tlow<=Temperature<Thigh, the fan will run on 30% speed rate; if Thigh<=Temperature<Tcrit, the fan will run on 70% speed rate; if Tcrit>=Temperature, the fan will run on 100% speed rate. And there has a temperature hysteresis Thyst=2 Celsius scales. Assuming temperature has previously crossed above Tlow, Thigh or Tcrit, then the temperature must drop below the points corresponding Thyst(Tlow-Thyst, Thigh-Thyst or Tcrit-Thyst) in order for the condition to drive fan speed rate to lower level. For example:

- temperature is 58 Celsius scales, the fan speed rate is 30%; (Tlow<58<Thigh)
- temperature increases to 65 Celsius scales, the fan speed rate is 70%;(Thigh=65)
- temperature decreases to 63 Celsius scales, the fan speed rate is still 70%;(Thigh-Thyst =63)
- temperature decreases to 62 Celsius scales, the fan speed rate is 30%;(62<Thigh-Thyst)

The Tlow, Thigh, Tcrit, Thyst and fan speed rate for each temperature threshold are hard code, and couldn't be modified.

```
Switch# show environment
Fan tray status:
Index    Status
1        PRESENT
FanIndex  Status SpeedRate Mode
1-1       OK     30%     Auto
1-2       OK     30%     Auto
1-3       OK     30%     Auto
1-4       OK     30%     Auto
-------------------------------------------------------
```

Configuring Power

The switch supports to manage power status automatically. If the power is failed or the fan in power is failed, the switch will be alarm. If power is removed or inserted, the switch will notice user also.

User can show the power status to verify the power status.

```
Switch# show environment
---------------------------------------------------------
Power status:
Index   Status   Power   Type    Fans    Control
1       PRESENT  OK      AC      -       -
2       ABSENT   -       -       -       -
3       PRESENT  OK      DC(PoE) -       -
---------------------------------------------------------
```

Configuring Transceiver

The switch supports manage the transceiver information, and the transceiver information includes basic information and diagnostic information. The basic information includes transceiver type, vendor name, PN, S/N, wavelength and link length for supported type. The diagnostic information includes real-time temperature, voltage, current, optical transmit power, optical receive power and the threshold about these parameters. If the transceiver is inserted or removed, the real-time parameter is out of threshold, the switch will notice the users.

User can show the transceiver information to verify this function.

```
Switch# show transceiver detail
Port eth-1-2 transceiver info:
Transceiver Type: 10G Base-SR
Transceiver Vendor Name : OEM
Transceiver PN         : SFP-10GB-SR
Transceiver S/N        : 201033PST1077C
Transceiver Output Wavelength: 850 nm
Supported Link Type and Length:
    Link Length for 50/125um multi-mode fiber: 80 m
    Link Length for 62.5/125um multi-mode fiber: 30 m
-------------------------------------------------------------------------------
Transceiver is internally calibrated.
mA: milliamperes, dBm: decibels (milliwatts), NA or N/A: not applicable.
++ : high alarm, +  : high warning, -  : low warning, -- : low alarm.
The threshold values are calibrated.
-------------------------------------------------------------------------------
          High Alarm   High Warn   Low Warn    Low Alarm
Temperature  Threshold   Threshold   Threshold    Threshold
Port  (Celsius)    (Celsius)    (Celsius)    (Celsius)    (Celsius)
--------- ----------------- ---------- ---------- ---------- ----------
eth-1-2  25.92     95.00       90.00     -20.00      -25.00
-------------------------------------------------------------------------------------------------
             High Alarm   High Warn   Low Warn    Low Alarm
     Voltage    Threshold    Threshold  Threshold    Threshold
Port    (Volts)    (Volts)     (Volts)    (Volts)     (Volts)
--------- ----------------- ---------- ---------- ---------- ----------
eth-1-2   3.32      3.80       3.70       2.90        2.80
-------------------------------------------------------------------------------------------------
             High Alarm   High Warn   Low Warn    Low Alarm
      Current    Threshold    Threshold  Threshold    Threshold
Port   (milliamperes)  (mA)       (mA)       (mA)       (mA)
--------- ----------------- ---------- ---------- ---------- ----------
eth-1-2   6.41      20.00      18.00      1.00        0.50
-------------------------------------------------------------------------------------------------
      Optical     High Alarm   High Warn   Low Warn    Low Alarm
```

```
   Transmit Power   Threshold   Threshold   Threshold    Threshold
Port   (dBm)       (dBm)     (dBm)      (dBm)     (dBm)
--------- ------------------  ----------------  ----------------  ----------------  ----------------
eth-1-2   -2.41      2.01      1.00       -6.99       -7.96
-------------------------------------------------------------------------------------------------------
     Optical     High Alarm   High Warn    Low Warn    Low Alarm
   Receive Power   Threshold   Threshold    Threshold    Threshold
Port   (dBm)       (dBm)     (dBm)      (dBm)     (dBm)
--------- ------------------  ----------------  ----------------  ----------------  ----------------
eth-1-2    -12     -    1.00      0.00      -19.00     -20.00
-------------------------------------------------------------------------------------------------------
```

Upgrade bootrom

The switch supports to upgrade the bootrom image when system is running. And after upgrading, you must reboot the switch to take effect.

step 1 Copy bootrom image file to the flash

```
Switch# copy mgmt-if tftp://10.10.38.160/bootrom.bin flash:/boot/
```

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Upgrade the bootrom

```
Switch(config)# update bootrom flash:/boot/bootrom.bin
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Reboot the system

```
Switch# reboot
```

step 6 Validation

After the above setting, you can show uboot version information of platform:

```
Switch# show version
……
EPLD Version is 1
BootRom Version is 3.0.2
```

Upgrade EPLD

The switch supports to upgrade the EPLD image when system is running. And after upgrading, you must reboot the switch to take effect.

step 1 Copy epld image file to the flash

```
Switch# copy mgmt-if tftp://10.10.38.160/vme_v1.0 flash:/boot/vme_v1.0
```

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Upgrade the epld

```
Switch(config)# update epld flash:/boot/vme_v1.0
```

step 4 Exit the configure mode

```
Switch(config)# exit
```

step 5 Reboot the system

```
Switch# reboot
```

step 6 Validation

After the above setting, then power off and restart the device， you can show epld version information with command:

```
Switch# show version
……
EPLD Version is 1
BootRom Version is 3.0.2
```

9.5 Configuring Bootrom

9.5.1 Overview

Function Introduction

The main function of Bootrom is to initialize the board simply and load the system image to boot. You can use some necessary commands in bootrom mode.

Bootrom can load the system image both from TFTP server and persistent storage like flash. Then you can configure the Switch and TFTP server IP address as environment variables in Bootrom mode for boot the system image.

9.5.2 Configuration

Configuring Boot from TFTP Server

Method 1: Boot the system from TFTP server

Save the configuration and reboot the system:

```
bootrom:> setenv bootcmd boot_tftp OS-ms-v3.1.9.it.r.bin
bootrom:> saveenv
bootrom:> reset
```

Method 2: Method 1:Boot the system from TFTP server without password

Save the configuration and reboot the system:

```
bootrom:> setenv bootcmd boot_tftp_nopass OS-ms-v3.1.9.it.r.bin
bootrom:> saveenv
bootrom:> reset
```

Method 3: Boot the system from TFTP server and reboot automatically

```
bootrom:> boot_tftp OS-ms-v3.1.9.it.r.bin
```

Method 4: Boot the system from TFTP server and reboot automatically without password

```
bootrom:> boot_tftp_nopass OS-ms-v3.1.9.it.r.bin
```

Validation

After the above setting, you can get show information:

```
bootrom:> reset
……..
TFTP from server 10.10.29.160; our IP address is 10.10.29.118
Filename 'OS-ms-v3.1.9.it.r.bin'.
```

```
Load address: 0xaa00000
Loading: octeth0: Up 100 Mbps Full duplex (port  0)
#################################################################
####################
done
Bytes transferred = 12314539 (bbe7ab hex), 1829 Kbytes/sec
```

Configuring Boot from FLASH

step 1 Boot the system from FLASH

Save the configuration and reboot the system:

```
bootrom:> setenv bootcmd boot_flash OS-ms-v3.1.9.it.r.bin
bootrom:> saveenv
bootrom:> reset
```

step 2 Boot the system from without password

Save the configuration and reboot the system:

```
bootrom:> setenv bootcmd boot_flash_nopass OS-ms-v3.1.9.it.r.bin
bootrom:> saveenv
bootrom:> reset
Do you want to revert to the default config file ? [Y|N|E]:Y
```

step 3 Boot the system from FLASH and reboot automatically

```
bootrom:> boot_flash OS-ms-v3.1.9.it.r.bin
```

step 4 Boot the system from FLASH and reboot automatically without password

```
bootrom:> boot_flash_nopass OS-ms-v3.1.9.it.r.bin
Do you want to  revert  to the default config file ? [Y|N|E]:Y
```

step 5 Validation

After the above setting, you can get show information:

```
bootrom:> reset
……
Do you want to  revert  to the default config file ? [Y|N|E]:Y
### JFFS2 loading '/boot/OS-ms-v3.1.9.it.r.bin' to 0xaa00000
Scanning JFFS2 FS: . done.
### JFFS2 load complete: 12314539 bytes loaded to 0xaa00000
## Booting image at 0aa00000 ...
  Verifying Checksum ... OK
  Uncompressing Kernel Image ... OK
……
```

Set boot IP

step 1 Set Switch IP address , details information as follows

```
bootrom:> setenv ipaddr 10.10.29.101
bootrom:> saveenv
```

step 2 Set TFTP server IP address , details information as follows

```
bootrom:> setenv serverip 10.10.29.160
bootrom:> saveenv
```

step 3 validation

After the above setting, you can get show information:

```
bootrom:> printenv
printenv
bootdelay=5
baudrate=9600
download_baudrate=9600
……
stderr=serial
ipaddr=10.10.29.101
serverip=10.10.29.160
Environment size: 856/2044 bytes
```

Upgrade bootrom

step 1 upgrade the Bootrom image from TFTP server

```
bootrom:> upgrade_uboot bootrom.bin
```

step 2 validation

After the above setting, you can get show information:

```
bootrom:> version
version
Bootrom 3.0.3 (Development build) (Build time: Aug  4 2011 - 11:47:06)
```

Set gateway IP

step 1 Set Switch gateway IP address , details information as follows

```
bootrom:> setenv gatewayip 10.10.37.1
bootrom:> saveenv
```

step 2 Set network mask , details information as follows

```
bootrom:> setenv netmask 255.255.255.0
bootrom:> saveenv
```

step 3 validation

After the above setting, you can get show information:

```
bootrom:> printenv
printenv
bootdelay=5
baudrate=9600
download_baudrate=9600
……
stderr=serial
gatewayip=10.10.38.1
netmask=255.255.255.0
Environment size: 856/2044 bytes
```

9.6 Upgrade firmware

9.6.1 Overview

This is often referred to as "firmware upgrade", "version upgrade", or "image upgrade". The firmware of the switch includes all user oriented software functions. During the product maintenance phase, we regularly update the firmware, which usually includes new features, optimizations of existing features, or fixes for bugs discovered during use. Therefore, we usually recommend that users use the latest official version. Firmware upgrade includes two steps: firmware upload and firmware application.

9.6.2 Configuration

Upgrading the firmware version of the switch can be done through web page or CLI operation. Here, we take command line operation and TFTP upgrade as an example.



The above figure shows a Windows system PC connected to the management interface(mgmt-if) of switch. The TFTP service is enabled on the PC as the TFTP Server, and the switch is used as the TFTP Client to download new firmware (uImage) from the PC,then load the new firmware.

The detailed operation is as follows:

Step1 Configure the IP address for the NIC of the PC.

Configure the IP address 192.168.1.101 and mask 255.255.2555.0 for the NIC of the PC.

As long as it can be ensured that the switch and the PC can communicate with each other.

Step2 Enter configuration mode

```
Switch#configure terminal
```

Step3 Configure the IP address of the switch management port

默认情况下交换机的所有端口是 vlan1 的成员端口，我们配置 vlan1 的 IP 地址与 PC 对应网卡的 IP 地址在同一网段即可互通

```
Switch(config)# management ip address 192.168.1.1/24
```

Step4 Test Connectivity

```
Switch# ping mgmt-if 192.168.1.101
ping: Warning: source address might be selected on device other than: MANG_VRF
PING 192.168.1.101 (192.168.1.101) from 192.168.1.1 MANG_VRF: 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=0.607 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=0.839 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=128 time=0.288 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=128 time=0.790 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=128 time=0.818 ms

--- 192.168.1.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 0.288/0.668/0.839/0.207 ms
Switch#
```

开启 PC 的 TFTP 服务

在 PC 运行 TftpServer 工具（比如“Tftpd32”）使 PC 对外提供 TFTP 服务，将需要加载到交换机的固件（镜像文件）放在 TFTP 服务指定的目录下。



下载固件到交换机

```
Switch# copy mgmt-if tftp://192.168.1.101/hos-vastlakes-v3.0.20.bin flash:/boot/hos-vastlakes-v3.0.20.bin
```

注意：固件（镜像）必须保存 flash/boot 目录下。

指定交换机使用新的固件

```
Switch # boot system flash:/boot/hos-vastlakes-v3.0.20.bin
```

执行回车会提示确认，再次回车即可。

```
Are you sure to use flash:/boot/hos-vastlakes-v3.0.20.bin as the next boot image? [confirm]
```

重启交换机，交换机将使用新加载固件（镜像）。

```
switch5# reboot
Building configuration...
Reboot system? [confirm]
```

9.7 Configuring SmartConfig

9.7.1 Overview

Function Introduction

SmartConfig is a smart method of switch initial configuration. After enabling SmartConfig, switch will start to download configuration file or image file from tftp server， if not finding startup-config file at startup. Then switch will install these file， and it will reboot itself if had downloaded image file.

Note that we use deploy file to control the configuration file and image file downloaded by switch. Switch fetch these file according the deploy file, which is a XML-formatted file. The deploy file named smartdeploy.xml , while its content like below:

```
<SmartDeploy>
<ftype>init</ftype>
```

```
<hostprefix>Bruce</hostprefix>
<defItem>
<option>enable</option>
<image>def.bin</image>
<config>def.cfg</config>
</defItem>
<groups>
<Item>
<type>MAC</type>
<value>001e.0808.9100</value>
<image>switchOs.bin</image>
<config>startup.cfg</config>
</Item>
<Item>
<type>productid</type>
<value>09SWITCH-E48-10</value>
<image>productid.bin</image>
<config>productid.cfg</config>
</Item>
<Item>
<type>SN</type>
<value>E054GD116004</value>
<image>sn.bin</image>
<config>sn.cfg</config>
</Item>
</groups>
</SmartDeploy>
```

There are three kind of item used by switch to find out image file and configuration file fit itself. Switch will search fit item according sequence like MAC, SN , product-id。We just specify the file name in the deploy file, and place all these file on tftp server.

Principle Description

N/A

9.7.2 Configuration

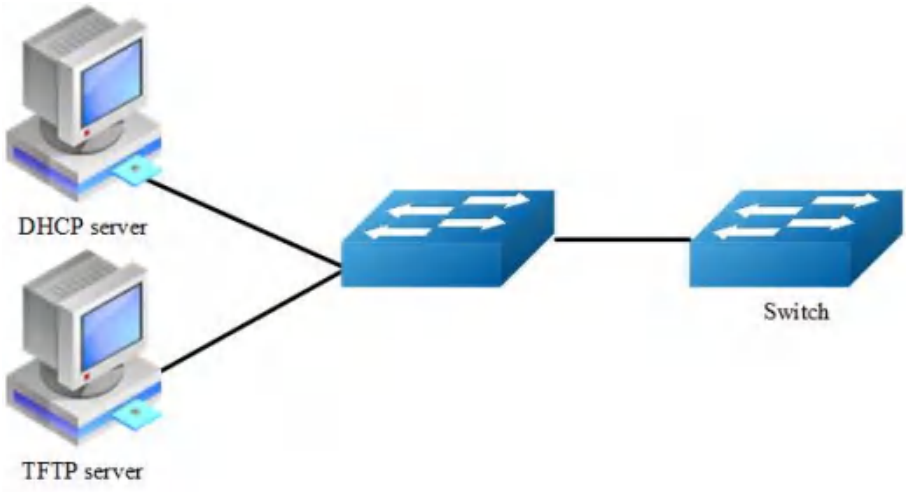


Figure 9-8 smart config

This figure is the network topology of testing SmartConfig function， We need two switches and two linux boxes to construct the test bed。"switch" in the figure is the switch we enable SmartCofng on. Note that the address of TFTP server provided by DHCP server can be used by switch to connect to TFTP server directly or via routes.

Enable smartConfig

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enable smartConfige

```
Switch(config)#smart-config initial-switch-deployment
```

step 3 Exit the configure mode

```
Switch (config)#exit
```

step 4 Validation

Use this command to check the smart-config settings:

```
Switch# show smart-config config
Smart-Config config:
 initial-switch-deployment: on
 hostname-prefix: on
 Send log message to console: on
```

Using smartConfig

SmartConfig was enable default，so we just make sure there is no startup-config.conf file. Then switch will start SmartConfig next boot. And we can delete startup-config.conf manually, so that Smartconfig will work after reboot. Procedure of configure SmartConfig as fallow：

step 1:

Configure smartdeploy.xml file， and place it with image file， configuration file to tftp server. The directory must be like this (Configuration files should be in conf directory and images should be in images directory.)：

```
smartconfig/
  |--conf/
  |--images/
  |--smartdeploy.xml
```

step 2:

Configure DHCP server，tftp server address option must be set；

step 3:

Make sure there is no startup-config.conf file;

step 4:

boot or reboot the system.

9.8 Reboot Logs

9.8.1 Overview

Function Introduction

Switch support display reboot logs. Depend on these logs, user can judge the reboot reasons of a switch. The reboot reasons include Manual Reboot, Power Off or Other Reasons.



User can find no more than 50 reboot logs through this command.

Detail about the show result as following:

Reboot Type	Description
POWER	Power outages
MANUAL	Cli "reboot/reload" is used
HIGH-TMPR	Reboot for abnormal high temperature
BHMDOG BHM	watchdog, monitor functional module
LCMDOG LCM	watchdog, monitor each LC
SCHEDULE	Schedule reboot
SNMP-RELOAD	SNMP reboot
HALFAIL	Reboot for HAGT communicate with HSRV failed， need stack enable
ABNORMAL	Unusual reboot, include reboot under shell
CTCINTR	Button reboot
LCATTACH	Reboot for LC attach CHSM failed
OTHER	Other reboot

Principle Description

N/A

9.8.2 Configuration

Reboot logs are enabled by default. User can display the logs as the following examples:

step 1 Display the logs

```
Switch# show reboot-info
Times    Reboot Type    Reboot Time (UTC)
---------+--------------+--------------------
1        MANUAL         2023-01-04 20:54:22
2        MANUAL         2023-01-04 21:00:30
3        MANUAL         2023-01-04 21:06:53
4        MANUAL         2023-01-04 21:10:13
5        MANUAL         2023-01-04 21:14:16
6        MANUAL         2023-01-04 21:20:50
```

9.9 Upload config

9.9.1 Overview

Function Introduction

It is possible to configure periodic uploading of running configurations to designated servers, so that users can compare configuration changes at different times to trace network issues. Or more timely detection of illegal configuration modifications.

Principle Description

N/A

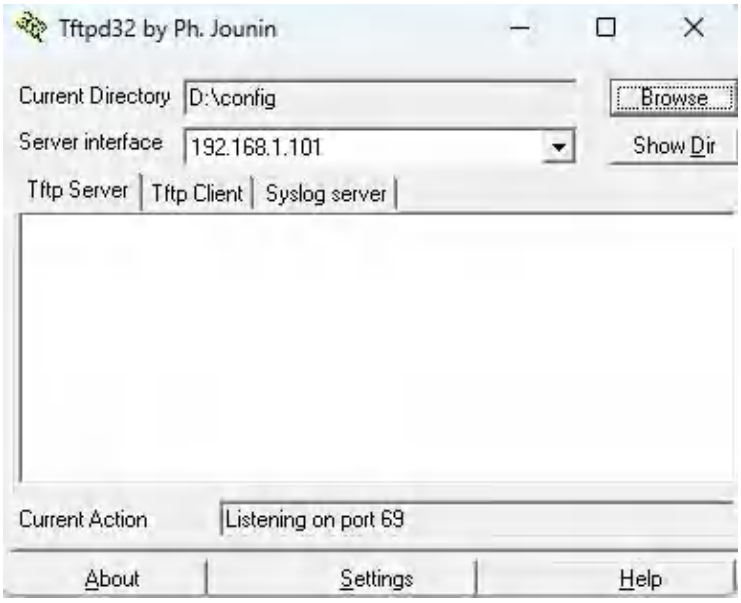
9.9.2 Configuration

The following example illustrates how to configure the periodic upload of switch configurations:



Figure 9-9 update configuration

step 1 Run TFTP APP on a CP and use it as TFTP Server



step 2 Complete the necessary network configuration on the switch to enable interoperability between the switch and the server.

```
Switch(config)# management ip address 192.168.1.7/24
Switch(config)# end
Switch# ping mgmt-if 192.168.1.101
ping: Warning: source address might be selected on device other than: MANG_VRF
PING 192.168.1.101 (192.168.1.101) from 192.168.1.7 MANG_VRF: 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=2.77 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=4.40 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=128 time=1.55 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=128 time=1.55 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=128 time=1.67 ms

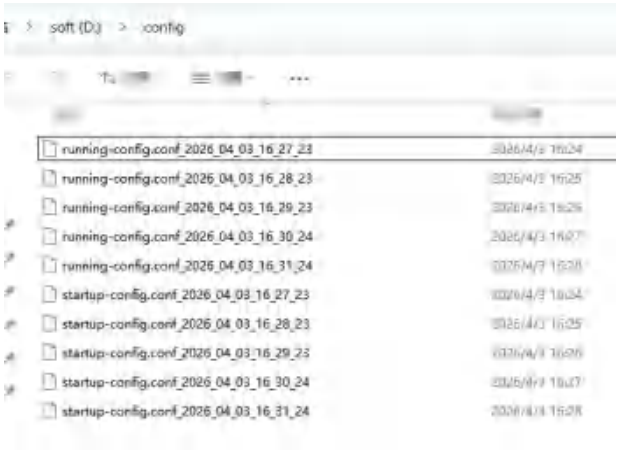
--- 192.168.1.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
```

```
rtt min/avg/max/mdev = 1.545/2.386/4.398/1.106 ms
Switch#
```

step 3 Configure server address and time interval for uploading configuration files.

```
Switch(config)# upload config tftp mgmt-if server-ip 192.168.1.101
```

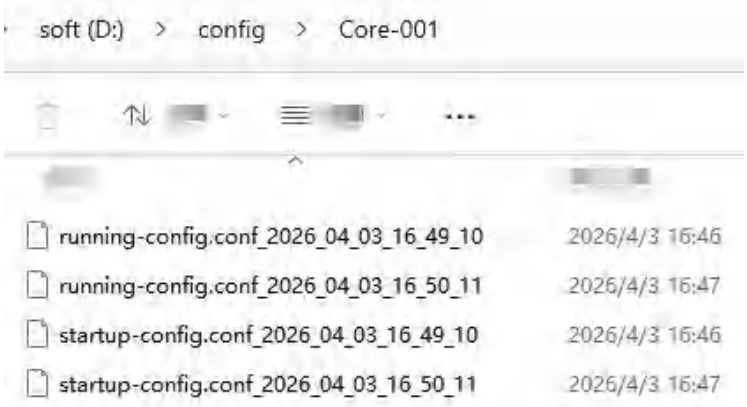
step 4 verify



If you need to store it in a folder under the server path, you need to specify it in the configuration in the server specified path.

For example, upload the configuration file to the "Core-001" folder in the server directory every 20 seconds

```
Switch(config)# upload config tftp mgmt-if time 20 server-ip 192.168.1.101 remote-path Core-001
```



10 Network Management Configuration Guide

10.1 Configuring Network Diagnosis

10.1.1 Overview

Function Introduction

Ping is a computer network administration utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer. The name comes from active sonar terminology.

Ping operates by sending Internet Control Message Protocol (ICMP) echo request packets to the target host and waiting for an ICMP response. In the process it measures the time from transmission to reception (round-trip time) [1] and records any packet loss. The results of the test are printed in form of a statistical summary of the response packets received, including the minimum, maximum, and the mean round-trip times, and sometimes the standard deviation of the mean.

Traceroute is a computer network tool for measuring the route path and transit times of packets across an Internet Protocol (IP) network.

Traceroute sends a sequence of Internet Control Message Protocol (ICMP) packets addressed to a destination host. Tracing the intermediate routers traversed involves control of the time-to-live (TTL) Internet Protocol parameter. Routers decrement this parameter and discard a packet when the TTL value has reached zero, returning an ICMP error message (ICMP Time Exceeded) to the sender.

Principle Description

N/A

10.1.2 Configuration

Ping IP with in-band port

```
Switch# ping 10.10.29.247
Switch# ping ipv6 2001:1000::1
```

Ping IP with management port

```
Switch# ping mgmt-if 10.10.29.247
Switch# ping mgmt-if ipv6 2001:1000::1
```

Ping IP with VRF instance

```
Switch# ping vrf vrf1 10.10.10.1
```

Traceroute IP with inner port

```
Switch# traceroute 1.1.1.2
Switch# traceroute ipv6 2001:1000::1
```

10.1.3 Application cases

Example for Ping

```
Switch # ping mgmt-if 192.168.100.101
PING 192.168.100.101 (192.168.100.101) 56(84) bytes of data.
64 bytes from 192.168.100.101: icmp_seq=0 ttl=64 time=0.092 ms
64 bytes from 192.168.100.101: icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from 192.168.100.101: icmp_seq=2 ttl=64 time=0.693 ms
64 bytes from 192.168.100.101: icmp_seq=3 ttl=64 time=0.071 ms
64 bytes from 192.168.100.101: icmp_seq=4 ttl=64 time=1.10 ms
--- 192.168.100.101 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4054ms
rtt min/avg/max/mdev = 0.071/0.408/1.104/0.421 ms, pipe 2
```

Example for traceroute

```
Switch# traceroute 1.1.1.2
traceroute to 1.1.1.2 (1.1.1.2), 30 hops max, 38 byte packets
 1  1.1.1.2 (1.1.1.2)  112.465 ms  102.257 ms  131.948 ms
Switch # ping mgmt-if ipv6 2001:1000::1
PING 2001:1000::1(2001:1000::1) 56 data bytes
64 bytes from 2001:1000::1: icmp_seq=1 ttl=64 time=0.291 ms
64 bytes from 2001:1000::1: icmp_seq=2 ttl=64 time=0.262 ms
64 bytes from 2001:1000::1: icmp_seq=3 ttl=64 time=0.264 ms
64 bytes from 2001:1000::1: icmp_seq=4 ttl=64 time=0.270 ms
64 bytes from 2001:1000::1: icmp_seq=5 ttl=64 time=0.274 ms
--- 2001:1000::1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.262/0.272/0.291/0.014 ms
```

10.2 Configuring NTP

10.2.1 Overview

Function Introduction

NTP is a tiered time distribution system with redundancy capability. NTP measures delays within the network and within the algorithms on the machine on which it is running. Using these tools and techniques, it is able to synchronize clocks to within milliseconds of each other when connected on a Local Area Network and within hundreds of milliseconds of each other when connected to a Wide Area Network. The tiered nature of the NTP time distribution tree enables a user to choose the accuracy needed by selecting a level (stratum) within the tree for machine placement. A time server placed higher in the tree (lower stratum number), provides a higher likelihood of agreement with the UTC time standard.

Some of the hosts act as time servers, that is, they provide what they believe is the correct time to other hosts. Other hosts act as clients, that is, they find out what time it is by querying a time server. Some hosts act as both clients and time servers, because these hosts are links in a chain over which the correct time is forwarded from one host to the next. As part of this chain, a host acts first as a client to get the correct time from another host that is a time server. It then turns around and functions as a time server when other hosts, acting as clients, send requests to it for the correct time.

Principle Description

N/A

10.2.2 Configuration

Configuring Client/Server mode connecting with in-band interface

Before configuring NTP client, make sure that NTP service is enabled on Server.

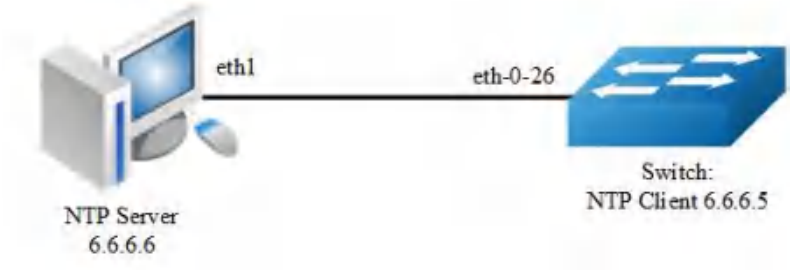


Figure 10-1 NTP

step 1 Enter the configure mode

```
Switch#configure terminal
```

step 2 Enter the vlan configure mode and create a vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and join the vlan

```
Switch(config)# interface eth-0-26
Switch(config-if)# switch access vlan 10
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 create a vlan interface and set the IP address

```
Switch(config)# interface vlan10
Switch(config-if)# ip address 6.6.6.5/24
Switch(config-if)# exit
```

step 5 Set the attributes of NTP client

Enable a trustedkey; Configure the IP address of the NTP server; Enable authentication; Once you have enabled authentication, the client switch sends the time-of-day requests to the trusted NTP servers only; Configure ntp ace.

```
Switch(config)# ntp key 1 serverkey
Switch(config)# ntp server 6.6.6.6 key 1
Switch(config)# ntp authentication enable
Switch(config)# ntp trustedkey 1
Switch(config)# ntp ace 6.6.6.6 none
```

step 6 Exit the configure mode

```
Switch(config)# end
```

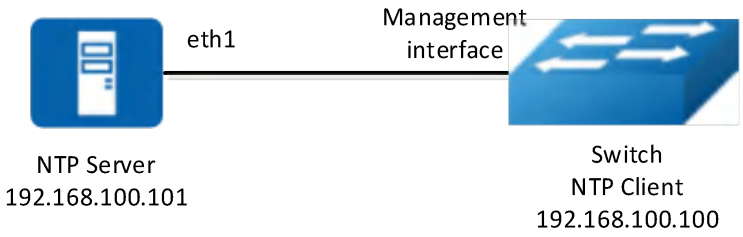
step 7 Validation

```
Switch# show ntp
Current NTP configuration:
============================================================
NTP access control list:
 6.6.6.6 mask 255.255.255.255 none
Unicast peer:
Unicast server:
 6.6.6.6 key 1
Authentication: enabled
Local reference clock:
Disable management interface

Switch# show ntp status
Current NTP status:
============================================================
clock is synchronized
stratum:       7
reference clock: 6.6.6.6
frequency:     17.365 ppm
precision:     2**20
```

```
reference time:  d14797dd.70b196a2  ( 1:54:37.440 UTC  Thu Apr  7 2011)
root delay:     0.787 ms
root dispersion: 23.993 ms
peer dispersion: 57.717 ms
clock offset:   -0.231 ms
stability:      6.222 ppm
Switch# show ntp associations
Current NTP associations:
  remote         refid     st   when poll reach   delay   offset  disp
==============================================================================
*6.6.6.6        127.127.1.0   6  50  128  37     0.778   -0.234  71.945
synchronized, + candidate, # selected, x falsetick, . excess, - outlier
```

Configuring Client/Server mode connecting with management interface



step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the attributes of NTP client

```
Switch(config)# ntp key 1 serverkey
Switch(config)# ntp server mgmt-if 192.168.100.101 key 1
Switch(config)# ntp authentication enable
Switch(config)# ntp trustedkey 1
Switch(config)# ntp ace 192.168.100.101 none
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ntp
Current NTP configuration:
============================================================
NTP access control list:
 192.168.100.101 mask 255.255.255.255 none
Unicast peer:
Unicast server:
 192.168.100.101(mgmt-if) key 1
Authentication: enabled
Local reference clock:
Only management interface
Switch# show ntp associations
Current NTP associations:
remote         refid     st when poll reach   delay   offset  disp
==============================================================================
```

```
*192.168.100.101 127.127.1.0     3  27  64   1    1.328   2.033 433.075
* sys.peer, + candidate, # selected, x falsetick, . excess, - outlyer
```

10.2.3 Application cases

Configuring NTP Server (Use the ntpd of linux system for example）

We use the "ntpd" of linux system for example to introduce how to set NTP server in the network.

Step 1 Display eth1 ip address

```
[root@localhost octeon]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:08:C7:89:4B:AA
          inet addr:6.6.6.6  Bcast:6.6.6.255  Mask:255.255.255.0
          inet6 addr: fe80::208:c7ff:fe89:4baa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3453 errors:1 dropped:0 overruns:0 frame:1
          TX packets:3459 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:368070 (359.4 KiB)  TX bytes:318042 (310.5 KiB)
```

Step 2 Check networks via Ping

```
[root@localhost octeon]# ping 6.6.6.5
PING 6.6.6.5 (6.6.6.5) 56(84) bytes of data.
64 bytes from 6.6.6.5: icmp_seq=0 ttl=64 time=0.951 ms
64 bytes from 6.6.6.5: icmp_seq=1 ttl=64 time=0.811 ms
64 bytes from 6.6.6.5: icmp_seq=2 ttl=64 time=0.790 ms
```

Step 3 Configure ntp.conf

```
[root@localhost octeon]# vi /etc/ntp.conf
server  127.127.1.0 # local clock
fudge   127.127.1.0 stratum 5
#
# Drift file.  Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
#
driftfile /var/lib/ntp/drift
broadcastdelay  0.008
broadcast 6.6.6.255
#
# PLEASE DO NOT USE THE DEFAULT VALUES HERE. Pick your own, or remote
# systems might be able to reset your clock at will. Note also that
# ntpd is started with a -A flag, disabling authentication, that
# will have to be removed as well.
#
#disable auth
keys       /etc/ntp/keys
trustedkey 1
```

Step 4 Configure keys

```
[root@localhost octeon]# vi /etc/ntp/keys
#
# PLEASE DO NOT USE THE DEFAULT VALUES HERE. Pick your own, or remote
```

```
# systems might be able to reset your clock at will. Note also that
# ntpd is started with a -A flag, disabling authentication, that
# will have to be removed as well.
#
1 M serverkey
```

Step 5 Start ntpd service

```
[root@localhost octeon]# ntpd
```

10.3 ConfiguringPTP

10.3.1 Overview

Brief Introduction

PTP (Precision Time Protocol) is a protocol for synchronizing computer clocks, defined by the IEEE 1588 standard. PTP aims to accurately synchronize clocks in distributed systems over a network to achieve high-precision time stamps and time synchronization, and is commonly used in industrial automation, transportation, power systems, and other fields that require high-precision time synchronization.

The PTP protocol is based on a master-slave architecture, where one device acts as the master clock and the rest act as slave clocks. The master clock is the time source in the network, and it synchronizes the clocks of all devices by transmitting time stamps to the slave clocks. The PTP protocol mainly consists of the following important components and steps:

- Synchronization process: The master clock publishes time information in the network. After receiving the time information, the slave clock adjusts its local clock to the time of the master clock.
- ransmission delay measurement: The master clock measures the propagation delay of its own time information to the slave clock, so that the slave clock can correct its own time.
- Time offset calculation: The clock calculates the offset between its own time and the master clock time based on the time information from the master clock and the local time information it receives.
- Time correction: Adjust the local time based on the calculated offset from the clock to keep the local time consistent with the master clock time.

The PTP protocol includes two versions: 1588 and G.8275.1. The 1588 includes 1588v1 and 1588v2. 1588v1 can only achieve sub-millisecond time synchronization accuracy, while 1588v2 can achieve sub-microsecond synchronization accuracy. G.8275.1 standard is a precise time synchronization standard defined by ITUT, which is similar to IEEE 1588 standard, both of which are designed to solve the problem of time synchronization across the entire network.

In wireless networks, clock synchronization between the mobile bearer network and the base station is required. 1588v2 can achieve sub-microsecond time synchronization accuracy with hardware-assisted processing; at the same time, it has lower costs and reduces dependence on GPS.

Background

PTP (Precision Time Protocol) is a protocol used to synchronize computer clocks, defined by the IEEE 1588 standard. The background of PTP lies in the need for high-precision time synchronization in various fields, such as industrial automation, transportation, power systems, and other applications where precise time coordination among distributed devices is essential.

The problem that PTP solves is to provide a high-precision time synchronization solution that ensures that devices in distributed systems maintain consistent time. Compared to previous technologies, PTP has the following advantages:

- Improved time synchronization accuracy: PTP achieves sub-microsecond time synchronization accuracy, which is significantly more precise than the sub-second accuracy of traditional NTP (Network Time Protocol). This high precision better meets the needs of many real-time applications.
- Applicability to various network topologies: PTP is applicable to any network topology, whether star, tree, or mesh, enabling high-precision synchronization. This wide applicability makes PTP suitable for a range of applications.
- Hardware-assisted processing: PTP utilizes hardware assistance to achieve faster time synchronization. This hardware assistance significantly improves synchronization speed, reduces latency, and enhances real-time system performance.

- Scalability: The design of PTP is scalable, facilitating the integration of more devices and sensors. This scalability enables PTP to be used in a wide range of applications such as the Internet of Things and smart homes.
- Independence from specific transmission media: PTP is independent of specific transmission media and can achieve high-precision time synchronization on Ethernet, fiber optic cables, and other transmission media. This flexibility makes PTP more practical for a variety of applications.

In summary, as a high-precision time synchronization protocol, PTP offers improved accuracy, broader applicability, faster processing, scalability, and flexibility compared to previous technologies. These advantages have made PTP an essential time synchronization technology in various fields.

Principle Description

Abbreviation and Terminology

Table 1-1 Abbreviation

Abbreviation	Description
GPS	Global Positioning System
NTP	Network Time Protocol
PTP	Precision Time Protocol
UTC	Coordinated Universal Time
TAI	International Atomic Time

Basic principle

The following are some important concepts in PTP, which can help users understand the function of PTP and configure it based on these understandings.

PTP Domain

The concept of PTP domain refers to the network that applies the Precision Time Protocol, also known as the PTP domain. In a PTP domain, there is only one master clock, and all devices are synchronized with this clock. Devices in the PTP domain can exchange time information through the PTP protocol, and perform time calibration and synchronization.

The supported range of PTP domain is 0-255. The PTP domain can be configured by command. The default domain is 0.

Clock

There are three clock types in the PTP protocol:

- Boundary Clock(BC): In the same PTP domain， the device with two or more ports participating in PTP time synchronization are called Boundary Clocks. The device synchronize time fron upstream devices through one port and distribute time to downstream devices through other ports. If the source clock distributes time to downstream devices through two or more ports, it is also called Boundary Clock.
- Ordinary Clock(OC): In the same PTP domain， the device with only one port participating in PTP time synchronization are called Ordinary Clocks. An ordinary clock either acts as the master clock providing the clock source or serves as the ultimate terminal, acquiring the clock from other clock sources without being able to act as an intermediate node to propagate the clock to other nodes.
- Transparent Clock(TC): As an intermediate node, a transparent clock receives the PTP packet without termination. It has a residence time bridge to calculate the residence time of the packet at this node, which is used to correct the time label before it is passed on.

Port

The ports with PTP protocol enabled are called PTP ports. There are two types in PTP ports:

- Master Port: The ports used to distribute time to downstream devices on BC or OC devices.
- Slave Port: The port used to receives time from upstream device on BC or OC devices.
- Passive Port： The ports does not receive or distribute time on BC devices.

Grandmaster Clock

The Grandmaster Clock (GM) is the highest-level clock in the PTP domain and the reference time for the entire PTP domain. All clock nodes in the PTP domain are organized in a certain hierarchy, and the time of the optimal clock will eventually be synchronized to the entire PTP domain through the interaction of PTP protocol messages between clock nodes. Therefore, it is also known as the clock source. The optimal clock can be statically specified through manual configuration, or dynamically elected through the Best Master Clock (BMC) protocol.

BMC algorithm

In the BMC algorithm, the optimal clock is dynamically elected through the interaction of PTP protocol messages between clock nodes. Each clock node exchanges information such as the optimal clock priority, time level, and time accuracy carried in the Announce message, and finally selects a node as the optimal clock for the PTP domain.

BMC selects Master Clock according to the following fields:

- Priority1： The valid range is 0-255. The smaller number indicates higher priority. The default value is 128.
- Clock-Class: The valid range is 6/7/13/14/52/58/187/193/248. The smaller number indicates higher priority. . The default value is 248.
- Clock-Accuracy: The valid range is 0x20-0x31/0xFE. The smaller number indicates higher priority. . The default value is 0xFE.
- Priority2: The valid range is 0-255. The smaller number indicates higher priority. The default value is 128.

The priority sequence is Priority1＞Clock-Class＞Clock-Accuracy＞Priority2. If Priority1 of two devices are same, it should compare the Clock-Class, and so on. The device with higher Priority/class/accuracy becomes the master clock.

Packet encapsulation

The synchronization process of PTP is based on PTP packets interaction. The downstream device calculates the offset and delay from the master according to the timestamp in the packets, and adjusts the local clock to synchronize with the master clock.

The PTP timestamp is added by ASIC to ensure the precision of PTP time. The ASIC chip has higher performance than the software.

PTP packets support MAC encapsulation and UDP(IPv4/IPv6) encapsulation.

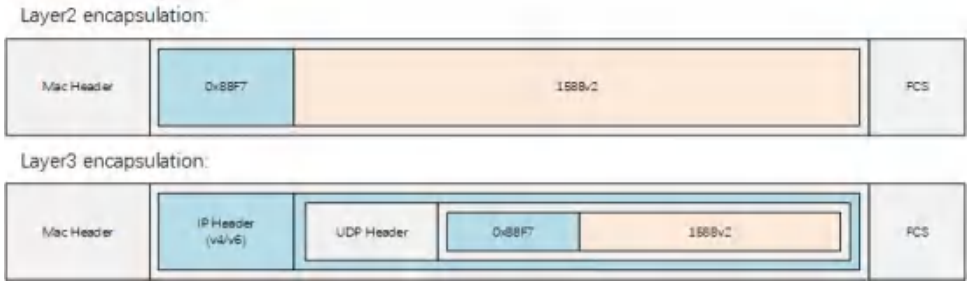


Figure 10-2 PTP packets

Synchronization process

The PTP time synchronization process involves the following steps:

- Selecting master and slave clocks: In the PTP synchronization process, one node is selected as the Master clock, and the other nodes serve as Slave clocks. The Master clock has a high-precision clock source and sends time information to the Slave clocks through the network, while the Slave clocks synchronize their clocks by correcting themselves.
- Clock identification and initialization: Nodes first need to identify each clock and initialize relevant parameters for the Master clock and Slave clocks, including clock frequency, delay, etc.
- Message passing between clocks: The core mechanism of PTP synchronization is to achieve time synchronization through message passing on the network. The Master clock periodically sends Sync messages to the network, and the Slave clocks receive the Sync messages and return corresponding Follow Up messages.
- Delay compensation and correction: During the PTP synchronization process, delay compensation is required due to message transmission delays. By calculating the transmission time of messages, delays can be compensated to ensure time accuracy. Simultaneously, based on the time difference between the Master clock and Slave clocks, the Slave clock corrects its clock.

- Fault tolerance mechanisms and optimization: In practical networks, due to factors such as complex network topologies and unstable link qualities, PTP synchronization may fail or lose precision. Therefore, during the PTP synchronization process, considerations for fault tolerance mechanisms are essential to improve system stability. Additionally, network conditions can be optimized for PTP synchronization, such as selecting better Master clocks or adjusting synchronization periods.
- Verification of time consistency: The verification of time consistency is also essential in the PTP synchronization process. Common verification methods include monitoring the time difference between Master clocks and Slave clocks, as well as assessing synchronization accuracy.

This process can be described in detail through the delay request-response mechanism in the IEEE 1588 standard. The Master clock periodically sends Sync messages and records the precise time t1 when the message leaves; the Slave clock records the precise time t2 when the Sync message arrives; the Master clock sends a Follow_up message containing t1; the Slave clock sends a Delay_Req message, records the precise time t3 when the message leaves; the Master clock records the precise time t4 when the Delay_Req message arrives; subsequently, the Master clock sends a Delay_Resp message containing t4 to the Slave clock. Through this method, Master and Slave clocks can achieve time synchronization.

Default Configuration

Table 1-2 Default Configuration

Item	Default Setting
PTP clock	Disabled
PTP device type	Transparent clock
PTP priority1 and PTP priority2	Default priority number is 128
PTP step mode	Two step
PTP announce interva	1 seconds
PTP announce timeout	3 seconds

10.3.2 Configuration Examples

Configuring Boundary Clock/Ordinary Clock

1. Topology



Figure 10-3 PTP Timing Domain

As shown in the Figure above， Switch1 is the master in the domain, it distributes time to downstream devices. Switch2 receives time from Switch1 and send to Switch3.

2. Configuration Steps

Configuration of Switch1

step 1 Enter the configuration mode

```
Switch1# configure terminal
```

step 2 Enable PTP globally and set priority1

Switch1 is an ordinary clock with only one interface to distribute the time.

Configure priority1 as 0. This will lead the Switch have the highest priority in this PTP timing domain, that means, it is the master clock of this domain.

```
Switch1(config)# ptp priority1 0
```

step 3 Configure device type

```
Switch1(config)# ptp device-type oc
```

Note： The switch defaults to TC, and BC requires at least 2 interfaces to enable PTP. In this example, Switch1 only has one interface to enable PTP and must be configured to OC.

step 4 Configure clock-step(Optional)

Time-stamp in PTP packec support one-step and two-step mode. The default configration is two-step.

- one-step: It add the time-stamp into the Sync packet.
- two-step： It add the time-stamp of the Sync packet into the Follow_Up packet.

It is suggested to use same clock-step in one PTP domain.

```
Switch1(config)# ptp clock-step two-step
```

step 5 Configure the delay mechanism mode(Optional)

There are two mechanism modes: E2E/P2P.

- e2e: use the req-resp delay measurement mechanism.
- p2p: use a node delay measurement mechanism

The default configuration is "e2e" mode.

```
Switch1(config)# ptp delay-mechanism p2p
```

step 6 Enable PTP on the interface

```
Switch1(config)# interface eth-0-9
Switch1(config-if)# ptp enable
```

step 7 Global Enable PTP

```
Switch1(config)# ptp start
```

Configuration of Switch2

step 1 Enter the configuration mode

```
Switch2# configure terminal
```

step 2 Set device type as boundary clock and Enable PTP globally

Switch2 receives time from Switch1 and send time to Switch3. Set the device type to boundary clock.

```
Switch2(config)# ptp device-type bc
```

step 3 Configure the delay mechanism mode(Optional)

Note： Please use same configuration as Switch1.

```
Switch2(config)# ptp delay-mechanism p2p
```

step 4 Enable PTP on the interface

```
Switch2(config)# interface eth-0-9
Switch2(config-if)# ptp enable
Switch2(config-if)# exit
Switch2(config)# interface eth-0-18
Switch2(config-if)# ptp enable
Switch2(config-if)# exit
```

step 5 Global Enable PTP

```
Switch1(config)# ptp start
```

Configuration of Switch3

step 1 Enter the configuration mode

```
Switch3# configure terminal
```

step 2 configure the device-type as OC

```
Switch3(config)# ptp device-type oc
```

step 3 Configure the delay mechanism mode(Optional)

Note： Please use same configuration as Switch2.

```
Switch3(config)# ptp delay-mechanism p2p
```

step 4 Enable PTP on the interface

```
Switch3(config)# interface eth-0-18
Switch3(config-if)# ptp enable
Switch3(config-if)# exit
```

step 5 Global Enable PTP

```
Switch1(config)# ptp start
```

Validation

Verify the PTP global state on Switch1

```
Switch1# show ptp
------------------ Global Config ------------------
PTP State               : enable
Port Number             : 1
Domain                  : 0
Step Mode               : two
Priotity1               : 0
Priotity2               : 128
Clock Type              : ordinary clock
Delay Machanism         : P2P
------------------ Time Status ------------------
    000102.fffe.030d02-0 seq 0 RESPONSE MANAGEMENT TIME_STATUS_NP
master_offset(ns)       0
ingress_time(ns)        0
gmPresent               false
gmIdentity              000102.fffe.030d02
Switch1#
```

Verify the PTP port state on Switch1

```
Switch1# show ptp interface
    000102.fffe.030d02-9 seq 0 RESPONSE MANAGEMENT PORT_DATA_SET
portIdentity        000102.fffe.030d02-9
portState           MASTER
logMinDelayReqInterval  0
peerMeanPathDelay     924
logAnnounceInterval   1
```

```
announceReceiptTimeout  3
logSyncInterval       0
delayMechanism        2
logMinPdelayReqInterval 0
versionNumber         2
Switch1#
```

Verify the PTP global state on Switch2

```
Switch2# show ptp
------------------ Global Config ------------------
PTP State               : enable
Port Number             : 2
Domain                  : 0
Step Mode               : two
Priotity1               : 128
Priotity2               : 128
Clock Type              : binary clock
Delay Machanism         : P2P
------------------ Time Status --------------------
    d85b22.fffe.0bba84-0 seq 0 RESPONSE MANAGEMENT TIME_STATUS_NP
master_offset(ns)       -2
ingress_time(ns)        1732806452900546058
gmPresent               true
gmIdentity              000102.fffe.030d02
Switch2#
```

Verify the PTP port state on Switch2

```
Switch2# show ptp interface
    d85b22.fffe.0bba84-9 seq 0 RESPONSE MANAGEMENT PORT_DATA_SET
portIdentity          d85b22.fffe.0bba84-9
portState             SLAVE
logMinDelayReqInterval  0
peerMeanPathDelay     921
logAnnounceInterval   1
announceReceiptTimeout  3
logSyncInterval       0
delayMechanism        2
logMinPdelayReqInterval 0
versionNumber         2
    d85b22.fffe.0bba84-18 seq 0 RESPONSE MANAGEMENT PORT_DATA_SET
portIdentity          d85b22.fffe.0bba84-18
portState             MASTER
logMinDelayReqInterval  0
peerMeanPathDelay     1282
logAnnounceInterval   1
announceReceiptTimeout  3
logSyncInterval       0
delayMechanism        2
logMinPdelayReqInterval 0
versionNumber         2
Switch2#
```

Verify the PTP global state on Switch3

```
Switch# show ptp
------------------ Global Config ------------------
```

```
PTP State               : enable
Port Number             : 1
Domain                  : 0
Step Mode               : two
Priotity1               : 128
Priotity2               : 128
Clock Type              : ordinary clock
Delay Machanism         : P2P
------------------- Time Status  -------------------
    d85b22.fffe.000401-0 seq 0 RESPONSE MANAGEMENT TIME_STATUS_NP
master_offset(ns)       -7
ingress_time(ns)        1732805307622102495
gmPresent               true
gmIdentity              000102.fffe.030d02
Switch#
```

Verify the PTP port state on Switch3

```
Switch# show ptp interface
    d85b22.fffe.000401-18 seq 0 RESPONSE MANAGEMENT PORT_DATA_SET
portIdentity         d85b22.fffe.000401-18
portState            FAULTY
logMinDelayReqInterval  0
peerMeanPathDelay     1284
logAnnounceInterval    1
announceReceiptTimeout  3
logSyncInterval       0
delayMechanism        2
logMinPdelayReqInterval 0
versionNumber         2
Switch#
```

10.4 Configuring Phy Loopback

10.4.1 Overview

Function Introduction

Phy loopback is a proprietary based loopback. There are 2 types of phy loopback: phy(including internal and external) level loopback and port level loopback.

- If a physical port is configured as "external phy loopback", all packets coming into this port should be loopback back from the port itself at phy level.
- If a physical port is configured as "internal phy loopback", all packets expected out from this port should be looped back to specified physical port.
- If a physical port is configured as "port loopback", all packets coming into this port should be looped back from the port itself, and whether to swap the SMAC with the DMAC should be selectable by users. And if the MAC is swapped, the CRC should be recalculated.

Principle Description

N/A

10.4.2 Configuration

Configuring external phy loopback

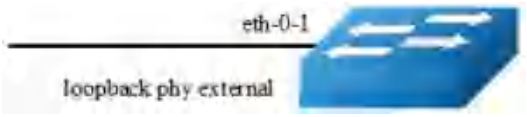


Figure 10-4 external phy topology

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set loopback phy external

```
Switch (config)# interface eth-0-1
Switch (config-if)# no shutdown
Switch (config-if)# loopback phy external
```

step 3 Exit the configure mode

```
Switch (config-if)# end
```

step 4 Validation

```
Switch# show phy loopback
 Interface  Type      DestIntf   SwapMac
 ------------------------------------------------
 eth-0-1    external  -          -
 ------------------------------------------------
```

Configuring internal phy loopback



Figure 10-5 Internal phy topology

step 1 Enter the configure mode

```
Switch # configure terminal
```

step 2 Enter the interface configure mode and set loopback phy internal and specify the destination interface

```
Switch (config)# interface eth-0-2
Switch (config-if)# no shutdown
Switch (config-if)# exit

Switch (config)# interface eth-0-1
Switch (config-if)# no shutdown
Switch (config-if)# loopback phy internal eth-0-2
```

step 3 Exit the configure mode

```
Switch (config-if)# end
```

step 4 Validation

```
Switch# show phy loopback
 Interface  Type      DestIntf   SwapMac
```

```
-----------------------------------------------
eth-0-1   internal   eth-0-2   -
-----------------------------------------------
```

Configuring port level loopback

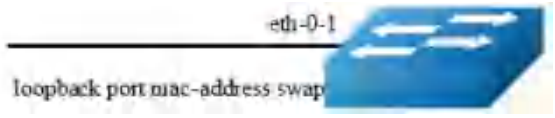


Figure 10-6 Port level topology

step 1 Enter the configure mode

```
Switch # configure terminal
```

step 2 Enter the interface configure mode and set loopback phy mac-address swap

```
Switch (config)# interface eth-0-1
Switch (config-if)# no shutdown
Switch (config-if)# loopback port mac-address swap
```

step 3 Exit the configure mode

```
Switch (config-if)# end
```

step 4 Validation

```
Switch# show phy loopback
 Interface  Type      DestIntf   SwapMac
-----------------------------------------------
eth-0-1   port     -       yes
-----------------------------------------------
```

10.4.3 Application cases

N/A

10.5 Configuring L2 ping

10.5.1 Overview

Function Introduction

The tool L2 ping is a useful application which's purpose is detecting the connection between two switches. The L2 ping tool is not same with the well-known 'ping IP-ADDRESS' in the WINDOWS system. The normal "ping" is realized by the protocol ICMP which is dependent on the IP layer, so it may be inapplicable if the destination device is only Layer 2 switch. But the protocol used by L2 ping is only relying on Layer 2 ethernet packets.

When L2 ping is started, the L2 ping protocol packet (with ether type '36873(0x9009)') is sent from a specified physical port to another specified destination port. At the destination end, the L2 ping protocol will be sent back via non 802.1ag loopback, or via a configuration "l2 ping response". The device which is pinging, will receive the ping response packet, and print the ping result.

Principle Description

N/A

10.5.2 Configuration

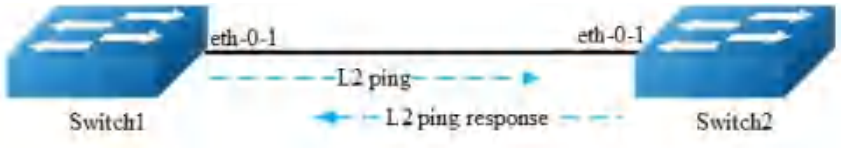


Figure 10-7 ping a switch port

The configurations are almost same on Switch1 and Switch2, except the parts which are specially pointed out.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and turn up the interface

```
Switch (config)# interface eth-0-1
Switch (config-if)# no shutdown
```

step 3 Enable the L2 ping response function

Configure on Switch2:

```
Switch (config-if)# l2 ping response enable
```

step 4 Exit the configure mode

```
Switch (config-if)# end
```

step 5 Using L2 ping

Operate on Switch1:

```
Switch1# l2 ping 001e.0808.58f1 interface eth-0-1 count 10 interval 1000 timeout 2000

Sending 10 L2 ping message(s):
64 bytes from 001e.0808.58f1: sequence = 0, time = 10ms
64 bytes from 001e.0808.58f1: sequence = 1, time = 15ms
64 bytes from 001e.0808.58f1: sequence = 2, time = 13ms
64 bytes from 001e.0808.58f1: sequence = 3, time = 12ms
64 bytes from 001e.0808.58f1: sequence = 4, time = 20ms
64 bytes from 001e.0808.58f1: sequence = 5, time = 21ms
64 bytes from 001e.0808.58f1: sequence = 6, time = 12ms
64 bytes from 001e.0808.58f1: sequence = 7, time = 16ms
64 bytes from 001e.0808.58f1: sequence = 8, time = 14ms
64 bytes from 001e.0808.58f1: sequence = 9, time = 17ms
L2 ping completed.
------------------------------------
10 packet(s) transmitted, 10 received, 0 % packet loss
```

001e.0808.58f1 is the MAC address of the interface on Switch2. It can be gained by command "show interface eth-0-1" on Switch2.

10.6 Configuring RMON

10.6.1 Overview

Function Introduction

RMON is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. You can use the RMON feature with the Simple Network Management Protocol (SNMP) agent in the switch to monitor all the traffic flowing among switched on all connected LAN segments.

RMON is a standard monitoring specification that defines a set of statistics and functions that can be exchanged between RMON-compliant console systems and network probes RMON provides you with comprehensive network-fault diagnosis, planning, and performance-tuning information.

Principle Description

N/A

10.6.2 Configuration

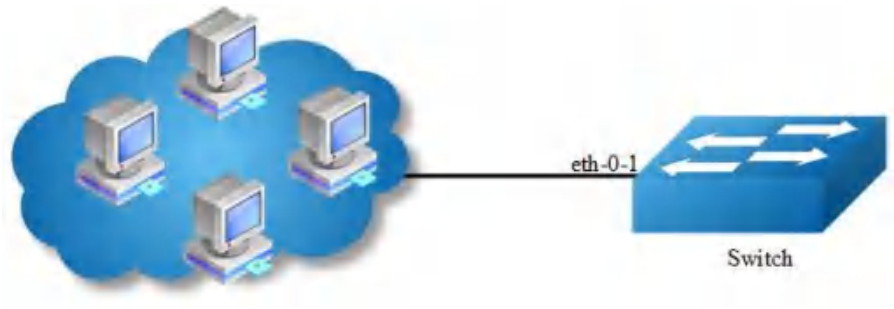


Figure 10-8 rmon

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and create a stats and a history

```
Switch(config)# interface eth-0-1
Switch(config-if)# rmon collection stats 1 owner test
Switch(config-if)# rmon collection history 1 buckets 100 interval 1000 owner test
Switch(config-if)# exit
```

step 3 Create an event with log and trap both set.

```
Switch(config)# rmon event 1 log trap public description test_event owner test
```

step 4 Create a alarm using event 1 we created before and monitor the alarm on ETHERSTATSBROADCASTPKTS on eth-0-1

```
Switch(config)# rmon alarm 1 etherStatsEntry.6.1 interval 1000 delta rising-threshold 1000 event 1 falling-threshold 1 event 1
owner test
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show rmon statistics
 Rmon collection index 1
    Statistics ifindex = 1, Owner: test
    Input packets 0, octets 0, dropped 0
    Broadcast packets 0, multicast packets 0, CRC alignment errors 0, collisions 0
    Undersized packets 0, oversized packets 0, fragments 0, jabbers 0
    # of packets received of length (in octets):
    64: 0,  65-127: 0, 128-255: 0
```

```
    256-511: 0,  512-1023: 0, 1024-max: 0

Switch# show rmon history
 History index = 1
    Data source ifindex = 1
    Buckets requested = 100
    Buckets granted = 100
    Interval = 1000
    Owner: test

Switch# show rmon event
 Event Index = 1
    Description: test_event
    Event type Log & Trap
    Event community name: public
    Last Time Sent = 00:00:00
    Owner: test

Switch# show rmon alarm
 Alarm Index = 1
 Alarm status = VALID
 Alarm Interval = 1000
 Alarm Type is Delta
 Alarm Value = 00
 Alarm Rising Threshold = 1000
 Alarm Rising Event = 1
 Alarm Falling Threshold = 1
 Alarm Falling Event = 1
 Alarm Owner is test
```

10.7 Configuring SNMP

10.7.1 Overview

Function Introduction

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a MIB. The SNMP manager can be part of a network management system (NMS). The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent. The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data. An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Error user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events may send a trap.

Principle Description

SNMP module is based on the following RFC draft:

- SNMPv1: Defined in RFC 1157.
- SNMPv2C: Defined in RFC 1901.
- SNMPv3: Defined in RFC 2273 to 2275.

Following is a brief description of terms and concepts used to describe the SNMP protocol:

- **Agent**: A network-management software module, an agent has local knowledge of management information and translates that information into a form compatible with SNMP.
- **Management Information Base (MIB)**: Management Information Base, collection of information is organized hierarchically.

- **Engine ID**: A unique ID for a network's node.
- **Trap**: Used by managed devices to asynchronously report events to the NMS.

10.7.2 Configuration

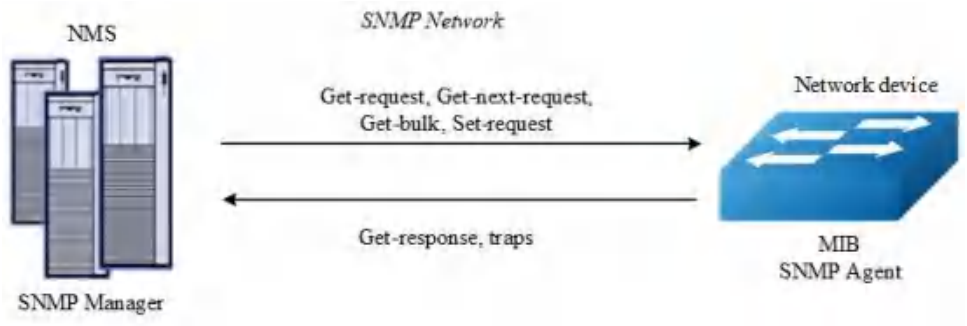


Figure 10-9 snmp

As shown in the figure SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in get-request, get-next-request, and set-request format.

Enable SNMP

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable SNMP globally

```
Switch(config)# snmp-server enable
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config
snmp-server enable
```

Configuring community string

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

Beginning in privileged EXEC mode, follow these steps to configure a community string on the switch.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configuring community string

Configure a view named "DUT"(optional); Configure a community named "public" with read access and view "DUT".

```
Switch(config)# snmp-server view DUT included 1
Switch(config)# snmp-server community public read-write (view DUT)
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config
snmp-server enable
snmp-server view DUT included .1
snmp-server community public read-only view DUT
```

Configuring SNMPv3 Groups, Users and Accesses

You can specify an identification name (engine ID) for the local SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, you can add new users to the SNMP group, and you can add access for the SNMP group.

Beginning in privileged EXEC mode, follow these steps to configure SNMP on the switch.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the globle configurations for SNMP

Set engineID; Set the user name, password, and authentication type; Create SNMP server; Set the authority for the group member.

```
Switch(config)# snmp-server engineID 8000123456
Switch(config)# snmp-server usm-user usr1 authentication md5 mypassword privacy des yourpassword
Switch(config)# snmp-server group grp1 user usr1 security-model usm
Switch(config)# snmp-server access grp1 security-model usm noauth
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config
snmp-server engineID 8000123456
snmp-server usm-user usr1 authentication md5 mypassword privacy des yourpassword
snmp-server group grp1 user usr1 security-model usm
snmp-server access grp1 security-model usm noauth
```

SNMPv1 and SNMPv2 notifications configure

Beginning in privileged EXEC mode, follow these steps to configure SNMP on the switch.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the global configurations for SNMP

Enable all supported traps; Configure a remote trap manager which IP is “10.0.0.2”; Configure a remote trap manager which IPv6 address is “2001:1000::1”.

```
Switch(config)# snmp-server trap enable all
Switch(config)# snmp-server trap target-address 10.0.0.2 community public
Switch(config)# snmp-server trap target-address 2001:1000::1 community public
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config
snmp-server trap target-address 10.0.0.2 community public
snmp-server trap target-address 2001:1000::1 community public
snmp-server trap enable vrrp
snmp-server trap enable igmp snooping
snmp-server trap enable ospf
snmp-server trap enable pim
snmp-server trap enable stp
snmp-server trap enable system
snmp-server trap enable coldstart
snmp-server trap enable warmstart
snmp-server trap enable linkdown
snmp-server trap enable linkup
```

Configuring SNMPv3 notifications

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the global configurations for SNMP

Enable all supported traps; Configure a trap notify item for SNMPv3; Configure a remote trap manager's IP address; Configure a remote trap manager's IPv6 address; Add a local user to SNMPv3 notifications.

```
Switch(config)# snmp-server trap enable all
Switch(config)# snmp-server notify notif1 tag tmptag trap
Switch(config)# snmp-server target-address targ1 param parm1 10.0.0.2 taglist tmptag
Switch(config)# snmp-server target-address t1 param p1 2001:1000::1 taglist tag1
Switch(config)# snmp-server target-params parm1 user usr1 security-model v3 message-processing v3 noauth
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config
snmp-server notify notif1 tag tmptag trap
snmp-server target-address t1 param p1  2001:1000::1 taglist tag1
snmp-server target-address targ1 param parm1  10.0.0.2 taglist tmptag
snmp-server target-params parm1 user usr1 security-model v3 message-processing v3 noauth
snmp-server trap enable vrrp
snmp-server trap enable igmp snooping
snmp-server trap enable ospf
snmp-server trap enable pim
snmp-server trap enable stp
snmp-server trap enable system
snmp-server trap enable coldstart
snmp-server trap enable warmstart
snmp-server trap enable linkdown
snmp-server trap enable linkup
```

10.8 Configuring SFLOW

10.8.1 Overview

Function Introduction

sFlow is a technology for monitoring traffic in data networks containing switches and routers. In particular, it defines the sampling mechanisms implemented in a sFlow Agent for monitoring traffic, and the format of sample data used by the sFlow Agent when forwarding data to a central data collector.

The architecture and sampling techniques used in the sFlow monitoring system are designed to provide continuous site-wide (and network-wide) traffic monitoring for high speed switched and routed networks.

The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched flows, and time-based sampling of network interface statistics.

Default Configuration for sflow：

Feature	Default Setting
global sflow	disabled
sflow on port	disable
collector udp port	6343
counter interval time	20 seconds

Principle Description

N/A

10.8.2 Configuration

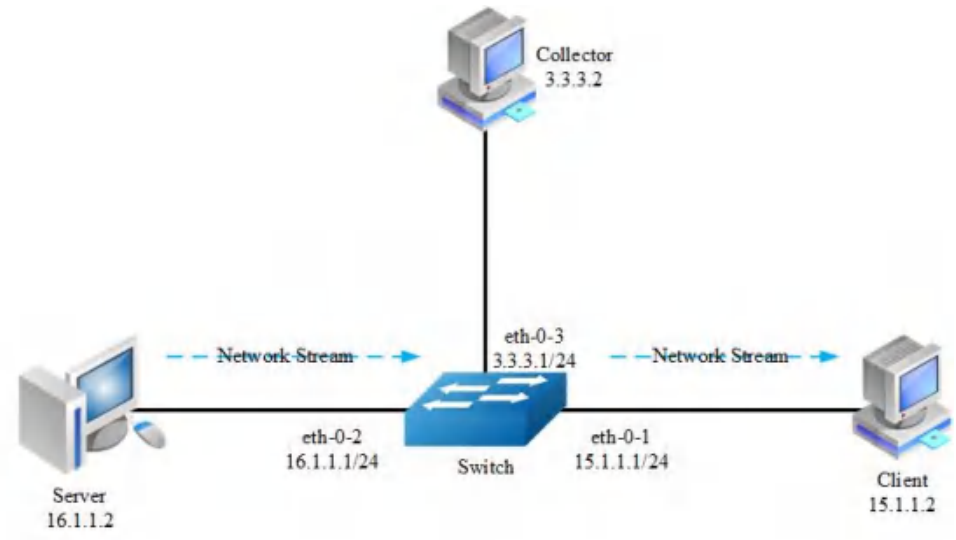


Figure 10-10 sflow

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable sflow globally

```
Switch(config)# sflow enable
```

step 3 Set the global attribute for sflow

Set the agent IP address， set the collector IP address and udp port. If the udp port is not specified, it means default port 6364.

```
Switch(config)# sflow agent ip 3.3.3.1
Switch(config)# sflow collector 3.3.3.2 6342
```

Set the agent and collector with IPv6:

```
Switch(config)# sflow agent ipv6 2001:2000::2
Switch(config)# sflow collector 2001:2000::1
```


At list one Agent and one collector must be configured for sflow. User can use IPv4 or IPv6.

Set the interval to send interface counter information (optional):

```
Switch(config)# sflow counter interval 15
```

step 4 Enter the interface configure mode and set the attributes of the interfaces

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 15.1.1.1/24
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)#no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 16.1.1.1/24
Switch(config-if)# exit

Switch(config)# interface eth-0-3
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 3.3.3.1/24
Switch(config-if)# exit
```

step 5 Enable sflow for input packets on eth-0-1

```
Switch(config)# interface eth-0-1
Switch(config-if)# sflow flow-sampling rate 8192
Switch(config-if)# sflow flow-sampling enable input
Switch(config-if)# sflow counter-sampling enable
Switch(config-if)# exit
```

step 6 Validation

To display the sflow configuration, use following command：

```
Switch# show sflow
sFlow Version: 5
sFlow Global Information:
 Agent IPv4 address          : 3.3.3.1
 Agent IPv6 address          : 2001:1000::2
 Counter Sampling Interval    : 15 seconds
 Collector 1:
  IPv4 Address: 3.3.3.2
  vrf: N/A
```

```
 Port: 6342
Collector 2:
 IPv6 Address: 2001:1000::1
 vrf: N/A
 Port: 6343

sFlow Port Information:
                    Flow-Sample   Flow-Sample
Port      Counter  Flow     Direction    Rate
---------------------------------------------------------
eth-0-1    Enable   Enable   Input       8192
```

10.8.3 Application cases

N/A

10.9 Configuring LLDP

10.9.1 Overview

Function Introduction

LLDP（Link Layer Discovery Protocol）is the discovery protocol on link layer defined as standard in IEEE 802.1ab. Discovery on Layer 2 can locate interfaces attached to the devices exactly with connection information on layer 2, such as VLAN attribute of port and protocols supported, and present paths among client, switch, router, application servers and other network servers. This detailed description is helpful to get useful information for diagnosing network fast, like topology of devices attached, conflict configuration between devices, and reason of network failure.

Principle Description

N/A

10.9.2 Configuration

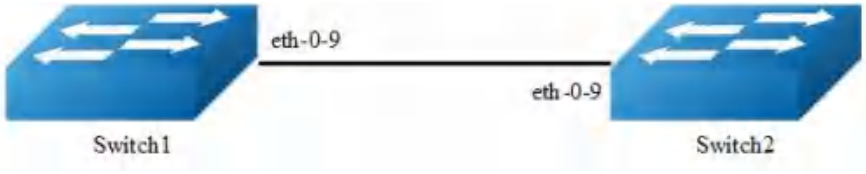


Figure 10-11 lldp

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable SNMP globally

```
Switch(config)# lldp enable
```

step 3 Enter the interface configure mode and set the attributes of LLDP on the interface

```
Switch(config)# interface eth-0-9
Switch(config)# no shutdown
Switch(config-if)# no lldp tlv 8021-org-specific vlan-name
Switch(config-if)# lldp tlv med location-id ecs-elin 1234567890
Switch(config-if)# lldp enable txrx
Switch(config-if)# exit
```

step 4 Set LLDP timers (optional)

Configure the transmitting interval of LLDP packet to 40 seconds; Configure the transmitting delay of LLDP packet to 3 seconds; Configure the reinit delay of LLDP function to 1 second.

```
Switch(config)# lldp timer msg-tx-interval 40
Switch(config)# lldp timer tx-delay 3
Switch(config)# lldp timer reinitDelay 1
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

To display the LLDP configuration, use following command:

```
Switch# show lldp local config
LLDP global configuration:
============================================================
LLDP function global enabled : YES
LLDP msgTxHold    : 4
LLDP msgTxInterval : 40
LLDP reinitDelay   : 1
LLDP txDelay      : 3
Switch# show lldp local config interface eth-0-9
LLDP configuration on interface eth-0-9 :
============================================================
LLDP admin status : TXRX
Basic optional TLV Enabled:
  Port Description TLV
  System Name TLV
  System Description TLV
  System Capabilities TLV
  Management Address TLV
IEEE 802.1 TLV Enabled:
  Port Vlan ID TLV
  Port and Protocol Vlan ID TLV
  Protocol Identity TLV
IEEE 802.3 TLV Enabled:
  MAC/PHY Configuration/Status TLV
  Power Via MDI TLV
  Link Aggregation TLV
  Maximum Frame Size TLV
LLDP-MED TLV Enabled:
  Med Capabilities TLV
  Network Policy TLV
  Location Identification TLV
  Extended Power-via-MDI TLV
  Inventory TLV
Switch# show running-config
!
lldp enable
lldp timer msg-tx-interval 40
lldp timer reinit-delay 1
lldp timer tx-delay 3
!
interface eth-0-9
```

```
lldp enable txrx
 no lldp tlv 8021-org-specific vlan-name
 lldp tlv med location-id ecs-elin 1234567890
!
Switch# show lldp neighbor
Local Port eth-0-1 has 0 neighbor(s)

Local Port eth-0-2 has 0 neighbor(s)
…
Local Port eth-0-9 has 2 neighbor(s)

Remote LLDP Information of port eth-0-9
============================================================
Neighbor Index : 1
Chassis ID type: Mac address
Chassis ID    : 48:16:be:a4:d7:09
Port ID type   : Interface Name
Port ID       : eth-0-9
TTL : 160
Expired time: 134
…
Location Identification :
ECS ELIN: 1234567890
```

10.10 Configuring IPFIX

10.10.1 Overview

Function Introduction

Traffic on a data network can be seen as consisting of flows passing through network elements. For administrative or other purposes, it is often interesting, useful, or even necessary to have access to information about these flows that pass through the network elements. This requires uniformity in the method of representing the flow information and the means of communicating the flows from the network elements to the collection point. This is what IPFIX can do.

Before IPFIX was introduced, there is a Cisco private method NetFlow. IPFIX is similar to NetFlow and is based on NetFlow version 9.

Principle Description

Reference to RFC7011/RFC7012 (IPFIX version 10)

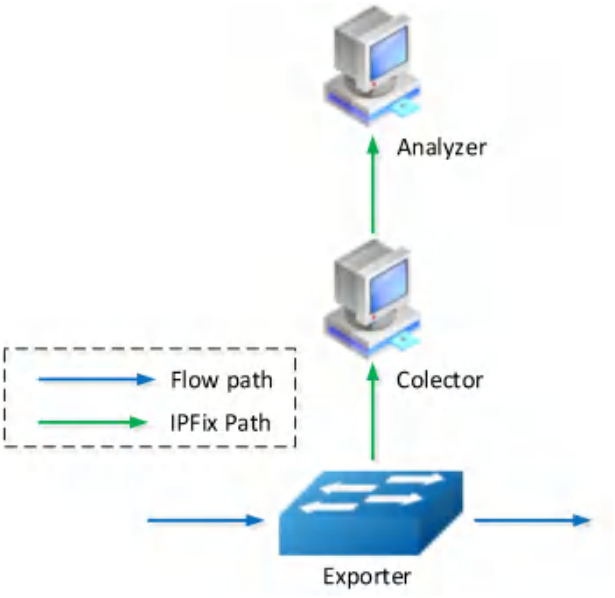


Figure 10-12 IPFIX basic Topology

The IPFIX has three types of devices: Exporter/Collector/Analyzer.

- Exporter analyze the network traffic and send the statistics result to the collector.
- Collector/Analyzer is used to collect and analyze the data, and process the advanced treatment, for example: generating a visual image.

The switch actors as the exporter. The following descriptions are about the exporter.

- Flow establish: the switch supports to recognise and establish a flow. The packets of the flow may have same MAC/IP/VLAN and so on. Which fields of the packets are the same and will be considered as the same flow can be configured by users according to the requirements.
- Flow aging: If there is no traffic for a long time after the flow established, it may occupy the system resources. Therefore system supports to aging the flows without traffic. User can configure the aging time according to their requirements. The default aging time is 1800 seconds which means if there is no packets after the flow established, the flow should be removed after 1800 seconds.
- Information sampling: system supports to set the sample rate, for example: randomly sample one packet from 100 packets. The larger the sampling interval is, the rougher the collected information will be, but the system burden will also be lighter. Conversely, the denser the sampling is, the more comprehensive the information will be, but the system's performance pressure will be greater. Users can choose a moderate sampling interval.
- Information output: System supports to send the information to the collector. User can specify the parameters such as destination IP / output interface / DSCP and so on.

There are two kinds of information to send in IPFIX packets: template and flow information . The template is used to tell the collect and analyzer how to parse the flow information, for example, the first fields is source MAC, the second field is source IP, and so on. The flow information should be filled according to the template. User can set the packet interval of the template packet and flow information packet. The default interval are 600 seconds. User can also specify the conditions to send packet immediately, for example, send the packet when the flow establishing or aging.

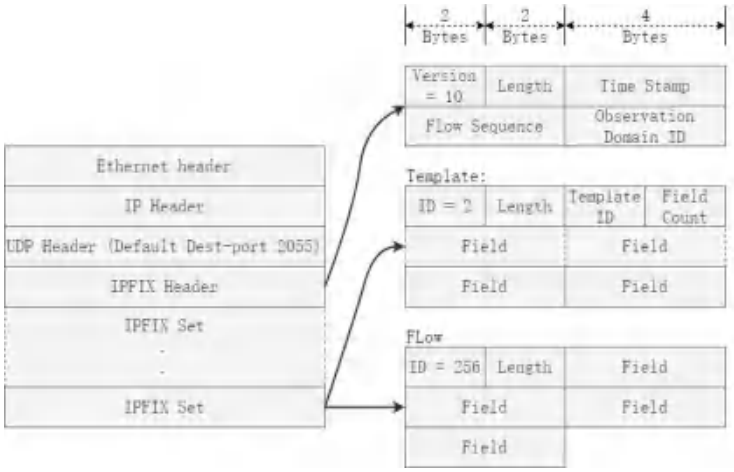
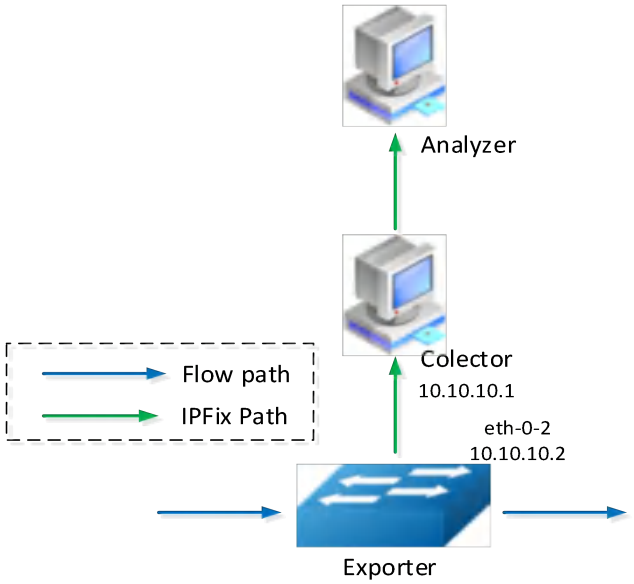


Figure 10-13 IPFIX Packet

As the figure above shows, IPFIX packets are encapsulated in UDP format. The default dest port is 2055. One IPFIX packets can include several sets, the set can be template or flow information.

10.10.2 Configuration



step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the aging time(optional)

Set the aging time as 300 seconds. The aging time is 1800 seconds by default.

```
Switch(config)# ipfix global
Switch(Config-ipfix-global)# flow aging 300
```

step 3 Configuring layer3 interface

```
Switch(config)# interface eth-0-2
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
```

```
Switch(config-if)# ip address 10.10.10.2/24
Switch(config-if)# exit
```

step 3 Configuring recorder

IPFIX can mark a flow with its five-tuple/VLAN/VNI information. User can specify the characteristic to separate the flow in the "Recorder" according to actual demande. In the following example, the packets with same mac source address/ ipv4 source and destination address/vxlan-vni belong to the same flow.

By the way, users should configure the "collect counter". In the following example, "collect counter bytes" and "collect counter packets" are both enabled, which means the bytes and packet should both be recordered.

```
Switch(config)# ipfix recorder recorder1
Switch(Config-ipfix-reocrder)# match mac source address
Switch(Config-ipfix-reocrder)# match ipv4 source address mask 32
Switch(Config-ipfix-reocrder)# match ipv4 destination address mask 32
Switch(Config-ipfix-reocrder)# match vxlan-vni
Switch(Config-ipfix-reocrder)# collect counter bytes
Switch(Config-ipfix-reocrder)# collect counter packets
Switch(Config-ipfix-reocrder)# exit
```

step 4 Configuring sampler

```
Switch(config)# ipfix sampler sampler1
Switch(Config-ipfix-sampler)# 1 out-of 100
Switch(Config-ipfix- sampler)# exit
```

step 5 Configuring exporter

Configure the encapsulation and sending parameters of IPFIX packets.

In the following example, the packets' destination address is 10.10.10.1(same as the address of the collector), sending out from eth-0-2, the packets interval is 200 seconds.

"event flow end timeout" means when the flow aging, the IPFIX packet should send to the collector immediately.

```
Switch(config)# ipfix exporter exporter1
Switch(Config-ipfix-exporter)# destination 10.10.10.1
Switch(Config-ipfix-exporter)# source interface eth-0-2
Switch(Config-ipfix-exporter)# flow data timeout 200
Switch(Config-ipfix-exporter)# event flow end timeout
Switch(Config-ipfix-exporter)# exit
```

step 6 Configuring monitor

```
Switch(config)# ipfix monitor monitor1
Switch(Config-ipfix-monitor)# recorder recorder1
Switch(Config-ipfix-monitor)# exporter exporter1
Switch(Config-ipfix-monitor)# exit
```

step 7 Enter the interface configure mode and apply ipfix

```
Switch(config)# interface eth-0-1
Switch(config-if)# ipfix monitor input monitor1
Switch(config-if)# exit
```

You can specify the monitor and sampler in one command, use the sampler to decide the sample rate. (If the sampler is not specified, the sample rate should be 100%)

```
Switch(config)# interface eth-0-1
Switch(config-if)# ipfix monitor input monitor1 sampler sampler1
Switch(config-if)# exit
```

step 8 Exit the configure mode

```
Switch(config)# end
```

step 9 Validation

Use the following commands to validate the configuration:

```
Switch# show ipfix global
IPFIX global informaition:
   Current flow cache number                  : 0(ingress: 0, egress: 0)
   Flow cache aging interval                  : 300 seconds
   Flow cache export interval                 : 5 seconds
   Flow cache memory usage threshold          : 90%
   Flow cache sampler mode                    : all flow
   Flow cache packet wraparound threshold     : 67108863
   Flow cache byte wraparound threshold       : 4294967295
   Flow cache dropped packet wraparound threshold   : 1023
   Flow cache jitter threshold                : 65535
   Flow cache latency threshold               : 16777215

Switch# show ipfix recorder recorder1
IPFIX recorder information:
  Name        : recorder1
  Description   :
  Match info    :
    match Source Mac Address
    match IPv4 Source Address
    match IPv4 Destination Address
    match Vxlanvni
  Collect info   :
    collect Flow Byte Number
    collect Flow Packet Number

Switch# show ipfix exporter exporter1
IPFIX exporter information:
  Name                    : exporter1
  Description             :
  Exporter Interface      : eth-0-2
  Domain ID               : 0
  Collector Name          : 10.10.10.1
  IPFIX message protocol      : UDP
  IPFIX message destination Port  : 2055
  IPFIX message TTL value     : 255
  IPFIX message DSCP value    : 63
  IPFIX data interval         : 200
  IPFIX template interval     : 1800
  IPFIX exporter events       :
    Flow aging event

Switch# show ipfix sampler sampler1
IPFIX sampler information:
  Name                    : sampler1
```

```
  Description              :
  Rate                     : 100

Switch# show ipfix monitor monitor1
IPFIX monitor information:
  Name                     : monitor1
  Description              :
  Recorder                 : recorder1
  exporter                 : exporter1
  flow mirror packet       : 0
  flow mirror destination  : NA
```

10.10.3 Application cases

N/A

10.11 ConfiguringHardware Monitor

10.11.1 Overview

Brief Introduction

The Hardware Monitor monitors network traffic at the Hardware level. It monitors network congestion in real time and reports the traffic to the network administrator for corresponding traffic management policies. The Hardware Monitor includes two modules: Buffer Monitor and Latency Monitor.

- **Buffer Monitor**: Buffer Monitor monitors the usage of the buffer cell of a port and reports congestion events in real time.
- **Latency Monitor**: Latency Monitor monitors the latency of packets, reports latency events in real time, and discards packets with high latency.

In addition, the Hardware Monitor provides watermark and notification functions.

- **Watermark Function**: The watermark function records the maximum buffer cell usage and packet forwarding delay of a port within a period of time.
- **Notification Function**: The notification function sends data collected by the Hardware Monitor module and reported events to the specified notification server.

Principle Description

The Hardware Monitor is divided into Buffer Monitor module and Latency Monitor module. The principles of these modules are described as follows.

1. Principle Description Of Buffer Monitor

The schematic diagram of the Buffer Monitor module is as follows:

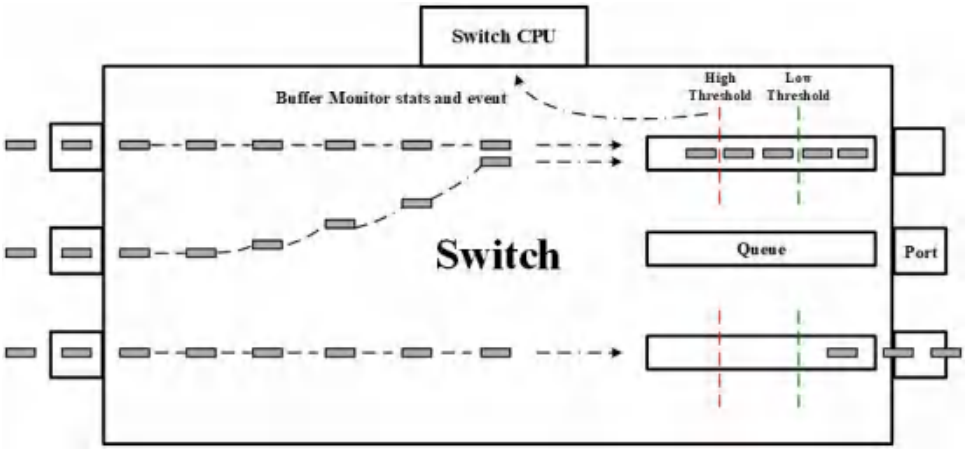


Figure 10-14 Buffer Monitor Principle description

When the traffic forwarded to an interface of the switch exceeds its maximum forwarding capability, the switch caches the packets that cannot be forwarded immediately and forwards them when congestion is relieved or relieved. The minimum unit used by the chip of Shengke switch to cache packets is the buffer cell. Network congestion can be determined by monitoring the occupation of the buffer cell of the port.

On an interface with the Buffer Monitor function enabled, the switch samples the usage of the interface's buffer cell every 10ms and stamps the time. When the number of buffer cells occupied by an interface exceeds the High Threshold, the switch immediately reports a congestion start event to the CPU with the time stamp of the event. When the number of buffer cells occupied by an interface decreases form the High Threshold to the Low Threshold, the switch immediately reports a congestion end event to the CPU with the time stamp of the event. If the notification function of the Buffer Monitor module is enabled, the system sends a Buffer Stats Notify message to the notification server at an interval defined by the administrator. The message contains the name of the monitored port, the usage of the buffer cell, and the timestamp when data is reported. When a congestion event occurs or ends, the system immediately sends a Micro-burst Start Notify and Micro-burst Stop Notify packet to the notification server. The content is also the name of the monitored port, buffer cell usage, and time stamp of the event report.

2. Principle Description Of Latency Monitor

The Latency Monitor module works as follows:

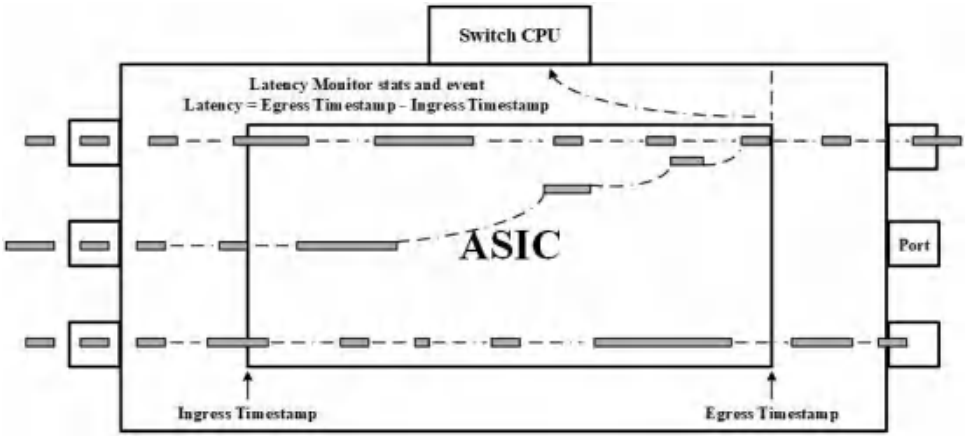


Figure 10-15 Latency Monitor Principle description

The Latency Monitor module calculates the latency of packets forwarded within the switch. You can configure eight latency intervals. On interfaces with Latency Monitor enabled, the switch calculates the number of packets in each latency interval every 10ms.

When the packet forwarding delay exceeds the High Threshold, the switch immediately reports a delay start event to the CPU, along with the timestamp when the event occurs, the packet forwarding delay, and the packet inbound port information. When the packet forwarding delay is lower than the Low Threshold, the switch immediately reports a delay end event to the CPU with the event occurrence time stamp, packet forwarding delay, and packet inbound port information. If the forwarding delay of a packet on a port with high forwarding delay is greater than Drop Threshold, the packet will be discarded. If the Latency Monitor notification function is

enabled, the system sends Latency Stats Notify packets to the notification server at an interval defined by the administartor. The Latency Stats Notify packets contain the name of the monitored port and the number of packets in each interval. When a latency event starts or ends, the system sends a Latency Event Start Notify and Latency Event Stop Notify packet to the notification server. These packets contain the name of the monitoring port and the inbound port name of the packet that triggers the event, the forwarding delay of the packet triggering the event and the timestamp when the event occurred.

- **Configure Buffer Monitor**

The following example shows how to enable the buffer monitor function on port eth-0-9 and the congestion event reporting function of the system. Set the high and low thresholds for reporting congestion events to 32000 and 64 buffer cells.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode

```
Switch(config)# interface eth-0-9
```

step 3 Enable the buffer monitor function of the interface

```
Switch(config-if)# hardware buffer-monitor enable
```

step 4 Exit the interface configure mode

```
Switch(config-if)# exit
```

step 5 Enable the congestion event reporting function of the system

```
Switch(config)# hardware buffer-monitor micro-burst enable
```

step 6 Set the high and low threshold for triggering congestion events

```
Switch(config)# hardware buffer-monitor micro-burst threshold low 64 high 3200
```

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

```
Switch# show hardware buffer-monitor status
Hardware buffer micro-burst event: enable
Hardware buffer micro-burst threshold low: 64    high: 3200
Port      buffer monitor
--------------------------------------------------------------------------------
eth-0-1   disable
eth-0-2   disable
eth-0-3   disable
eth-0-4   disable
eth-0-5   disable
eth-0-6   disable
eth-0-7   disable
eth-0-8   disable
eth-0-9   enable
eth-0-10  disable
eth-0-11  disable
eth-0-12  disable
eth-0-13  disable
eth-0-14  disable
eth-0-15  disable
eth-0-16  disable
```

```
eth-0-17  disable
eth-0-18  disable
eth-0-19  disable
eth-0-20  disable
eth-0-21  disable
eth-0-22  disable
eth-0-23  disable
eth-0-24  disable
eth-0-25  disable
eth-0-26  disable
```

step 9 Display Buffer Monitor statistics

Display the first 20 pieces of data recorded by the Buffer Monitor on port eth-0-9:

```
Switch# show hardware buffer-monitor interface eth-0-9 20
Brief CLI issued at : 08:39:15 UTC Fri Jan 28 2022
Flag (micro-burst)  : S - Start record    E - End record    U - Update record
Flag Time              Total count
-------------------------------------------------------------
-   08:39:11.698066906  58368
-   08:39:11.688067287  58367
-   08:39:11.678067668  58365
-   08:39:11.668068049  58366
-   08:39:11.658068430  58369
-   08:39:11.648068811  58368
-   08:39:11.638069192  58367
-   08:39:11.628069573  58366
-   08:39:11.618069954  58370
-   08:39:11.608070335  58367
-   08:39:11.598070716  58368
-   08:39:11.588071097  58369
-   08:39:11.578071478  58368
-   08:39:11.568071859  58367
-   08:39:11.558072240  58367
-   08:39:11.548072621  58369
-   08:39:11.538073002  58366
-   08:39:11.528073383  58368
-   08:39:11.518073764  58369
-   08:39:11.508074145  58369
```

View congestion events on port eth-0-9:

```
Switch# show hardware buffer-monitor micro-burst interface eth-0-9
Brief CLI issued at : 09:20:11 UTC Fri Jan 28 2022
Flag (micro-burst)  : S - Start record    E - End record
Flag Time(ns)          Total count        Threshold level
-----------------------------------------------------------------------
E   09:20:06.892459921  64                 0
S   09:19:05.087003316  3200               0
```

Configure Latency Monitor

The following example show how to enable the Latency Monitor and high-latency packet discard function on port eth-0-9, configure eight latency ranges, and set the latency event trigger threshold and high-latency packet discard threshold.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode

```
Switch(config)# interface eth-0-9
```

step 3 Enable the Latency Monitor function of the interface

```
Switch(config-if)# hardware latency-monitor enable
```

step 4 (Optional) Enable the function of discarding packets with high forwarding latency for the port

```
Switch(config-if)# hardware latency-monitor drop
```

step 5 (Optional) Set the threshold for the port to discard packets with high forwarding delay

The threshold for discarding packets with high forwarding latency is set to the sixth time range:

```
Switch(config-if)# hardware latency-monitor threshold drop 6
```

step 6 Exit the interface configure mode

```
Switch(config-if)# exit
```

step 7 Configure eight delay intervals for the system

```
Switch(config)# hardware latency-monitor range 640 3200 6400 12800 25600 128000 256000
```

step 8 Set the high and low threshold for triggering delayed events

Set the high threshold for triggering the delay event to 64000ns, and the low threshold to 1280ns:

```
Switch(config)# hardware latency-monitor threshold event low 1280 high 64000
```

step 9 Exit configure mode

```
Switch(config)# end
```

step 10 Validation

```
Switch# show hardware latency-monitor status
Hardware latency monitor range(ns)
Range1:    1-640     Range2:  640-3200     Range3:  3200-6400     Range4:  6400-12800
Range5: 12800-25600    Range6: 25600-128000   Range7:128000-256000   Range8:256000-more
Port    monitor     event low   event high   drop-threshold
--------------------------------------------------------------------------------
eth-0-1  disable    disable    disable    disable
eth-0-2  disable    disable    disable    disable
eth-0-3  disable    disable    disable    disable
eth-0-4  disable    disable    disable    disable
eth-0-5  disable    disable    disable    disable
eth-0-6  disable    disable    disable    disable
eth-0-7  disable    disable    disable    disable
eth-0-8  disable    disable    disable    disable
eth-0-9  enable     1280       64000      6
eth-0-10 disable    disable    disable    disable
eth-0-11 disable    disable    disable    disable
eth-0-12 disable    disable    disable    disable
eth-0-13 disable    disable    disable    disable
eth-0-14 disable    disable    disable    disable
eth-0-15 disable    disable    disable    disable
eth-0-16 disable    disable    disable    disable
eth-0-17 disable    disable    disable    disable
```

```
eth-0-18 disable    disable    disable    disable
eth-0-19 disable    disable    disable    disable
eth-0-20 disable    disable    disable    disable
eth-0-21 disable    disable    disable    disable
eth-0-22 disable    disable    disable    disable
eth-0-23 disable    disable    disable    disable
eth-0-24 disable    disable    disable    disable
eth-0-25 disable    disable    disable    disable
eth-0-26 disable    disable    disable    disable
```

step 11 Display Latency Monitor statistics

Query the first 10 data collected by Latency Monitor on port eth-0-9:

```
Switch# show hardware latency-monitor interface eth-0-9 10
Brief CLI issued at: 19:32:34 UTC Mon Feb 07 2022
Format:  0-640 - the count of packets with latency between 0ns and 640ns
                          0     640   3200   6400   12800    25600  128000  256000
                          -       -      -      -       -        -       -       -
Time                    640   3200   6400   12800   25600  128000  256000    more
09:52:57.366657807       35   2447   4492   3633     994    1256    1207     581
09:52:57.356657807       41   2519   4743   3675    1016    1463    1001     180
09:52:57.346657807       55   2752   5017   3957     961    1213     637      51
09:52:57.336657807       43   2463   4629   3519     877    1266    1123     721
09:52:57.326657807       41   2464   4827   3900     959    1243     805     403
09:52:57.316657807       37   2681   4951   3767    1015    1187     750     256
09:52:57.306657807       49   2411   4614   3541     848    1254    1175     754
09:52:57.296657807       32   2581   4571   3635    1038    1475    1138     167
09:52:57.286657807       41   2666   5036   3875     885    1057     874     207
09:52:57.276657807       38   2506   4802   3792     976    1163    1105     260
```

Check the delay events reported by port eth-0-9:

```
Switch# show hardware latency-monitor event interface eth-0-9
Brief CLI issued at: 10:06:13 UTC Fri Jan 28 2022
Flags          : S - Start record    E - End record
           Time  RangeId  Latency(ns)    Status    Source Port
-------------------------------------------------------------------------------
10:06:13.098854000     6      63994       S       eth-0-3
10:06:13.098849000     2       1280       E       eth-0-3
10:06:13.098845000     6      64032       S       eth-0-3
10:06:13.098840000     2       1282       E       eth-0-3
10:06:13.098836000     6      64036       S       eth-0-3
10:06:13.098832000     2       1304       E       eth-0-1
```

Configure notification function

The system network topology is as follows:

1. Topology

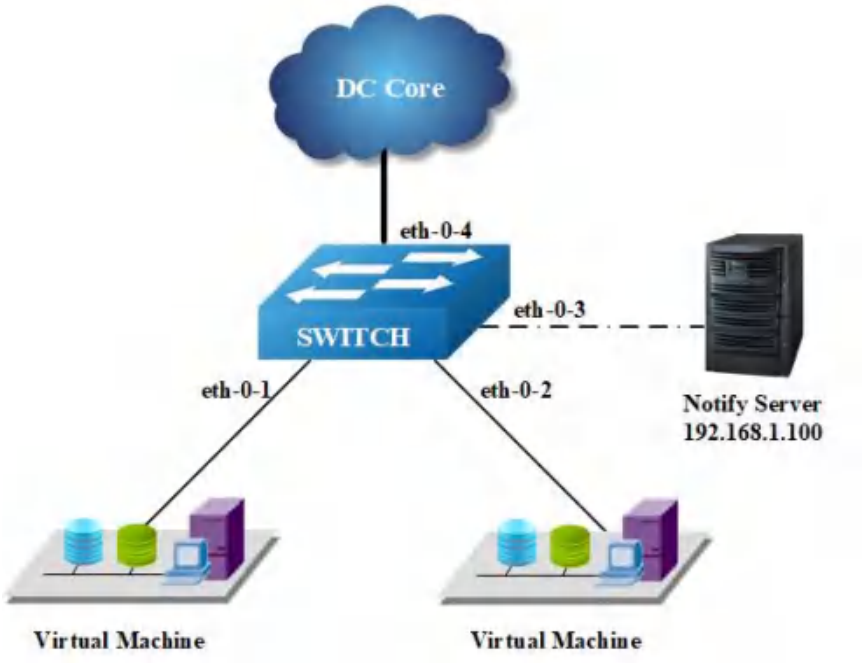


Figure 10-16 Topology of notification function

The following example describes how to enable the Hardware Monitor function of port eth-0-4 and set the IP address of the notification server to 192.168.1.100 to enable the data and event reporting function of the Hardware Monitor.

2. Configuration Steps

3. step 1 Enter configure mode

```
Switch# configure terminal
```

step 2 Configure the IP address of notification server

```
Switch(config)# hardware monitor notify server 192.168.1.100
```

step 3 Configure the interval of sending notification packets

```
Switch(config)# hardware monitor notify interval 20
```

step 4 Configure port eth-0-3 as a layer 3 port

```
Switch(config)# interface eth-0-3
Switch(config-if)# no switchport
```

step 5 Configure the IP address of port eth-0-3

```
Switch(config-if)# ip address 192.168.1.1/24
```

step 6 Exit interface configuration mode

```
Switch(config-if)# exit
```

step 7 Enable the congestion event reporting function and set the high and low threshold for triggering congestion events

```
Switch(config)# hardware buffer-monitor micro-burst enable
Switch(config)# hardware buffer-monitor micro-burst threshold low 32 high 256
```

step 8 Configure eight delay time ranges and high and low thresholds for triggering delayed events

```
Switch(config)# hardware latency-monitor range 640 1280 2560 5120 10240 20480 40960
Switch(config)# hardware latency-monitor threshold event low 64 high 6400
```

step 9 Enter interface configuration mode

```
Switch(config)# interface eth-0-4
```

step 10 Enable the Buffer Monitor function and Buffer Monitor related packet notification function of the port

```
Switch(config-if)# hardware buffer-monitor enable
Switch(config-if)# hardware buffer-monitor notify
```

step 11 Enable the Buffer Monitor function and Latency Monitor notification function of this port

```
Switch(config-if)# hardware latency-monitor enable
Switch(config-if)# hardware latency-monitor notify
```

step 12 Exit configuration mode

```
Switch(config-if)# end
```

step 13 Validation

```
Switch# show hardware buffer-monitor  status
Hardware buffer micro-burst event: enable
Hardware buffer micro-burst threshold low: 32    high: 256
Port      buffer monitor
--------------------------------------------------------------------------------
eth-0-1   disable
eth-0-2   disable
eth-0-3   disable
eth-0-4   enable
eth-0-5   disable
eth-0-6   disable
eth-0-7   disable
eth-0-8   disable
eth-0-9   disable
eth-0-10  disable
eth-0-11  disable
eth-0-12  disable
eth-0-13  disable
eth-0-14  disable
eth-0-15  disable
eth-0-16  disable
eth-0-17  disable
eth-0-18  disable
eth-0-19  disable
eth-0-20  disable
Switch# show hardware latency-monitor status
Hardware latency monitor range(ns)
Range1:    1-640     Range2:  640-1280     Range3:  1280-2560     Range4:  2560-5120
Range5:  5120-10240    Range6: 10240-20480    Range7: 20480-40960    Range8: 40960-more
Port    monitor     event low   event high   drop-threshold
--------------------------------------------------------------------------------
eth-0-1 disable     disable     disable     disable
eth-0-2 disable     disable     disable     disable
eth-0-3 disable     disable     disable     disable
eth-0-4 enable      64          6400        disable
```

```
eth-0-5  disable    disable    disable    disable
eth-0-6  disable    disable    disable    disable
eth-0-7  disable    disable    disable    disable
eth-0-8  disable    disable    disable    disable
eth-0-9  disable    disable    disable    disable
eth-0-10 disable    disable    disable    disable
eth-0-11 disable    disable    disable    disable
eth-0-12 disable    disable    disable    disable
eth-0-13 disable    disable    disable    disable
eth-0-14 disable    disable    disable    disable
eth-0-15 disable    disable    disable    disable
eth-0-16 disable    disable    disable    disable
eth-0-17 disable    disable    disable    disable
eth-0-18 disable    disable    disable    disable
eth-0-19 disable    disable    disable    disable
eth-0-20 disable    disable    disable    disable
```

11 Traffic Managemant Configuration Guide

11.1 Configuring QoS

11.1.1 Overview

Function Introduction

Quality of Service (QoS) can be used to give certain traffic priority over other traffic. Without QoS, all traffic in a network has the same priority and chance of being delivered on time. If congestion occurs, all traffic has the same chance of being dropped. With QoS, specific network traffic can be prioritized to receive preferential treatment. In turn, a network performs more predictably, and utilizes bandwidth more effectively.

Classification information can be carried in the Layer-3 IP packet header or the Layer-2 frame. IP packet headers carry the information using 6 bits or 3 bits from the deprecated IP type of service (TOS) field. Layer-2 802.1Q frames carry the information using a 2-byte Tag Control Information field.

All switches and routers accessing the Internet depend on class information to give the same forwarding treatment to packets with the same class information, and give different treatment to packets with different class information. A packet can be assigned class information, as follows:

- End hosts or switches along a path, based on a configured policy
- Detailed packet examination, expected to occur nearer to the network edge, to prevent overloading core switches and routers
- A combination of the above two techniques

Class information can be used by switches and routers along a path to limit the amount of allotted resources per traffic class.

Per-hop behavior is an individual device's behavior when handling traffic in the DiffServ architecture. An end-to-end QoS solution can be created if all devices along a path have consistent per-hop behavior.

QoS (Quality of Service) provides different service qualities to different requirements for all kinds of applications. For example, providing specified bandwidth, reducing the rate of lost for packets, and reducing the delay time and delay variation. To reach the goals, QoS often using different techniques to realize, for instance, traffic classification and tinting, traffic policing, traffic shaping, administrate network congestion, and avoid network congestion and so on. They will do as the following:

- Traffic classification: According to certain match regulations to recognized objects.
- Traffic policing: it will supervise the specification of certain traffic which entering the network equipments. When the traffic exceeds the specification, it will take actions of limitation or punishments, to protect the service provider's business profits and network resources from damages.
- Traffic shaping: it is a kind of traffic control that adjusts the output of speed rate actively. Usually it is used for adapting downstream router that provides network resources, and avoids unnecessary packets discard and congestion.
- Administrate network congestion: it is the actions that when network jam happens and needs to solve the resource competition. It usually put the packets into queue buffer, and uses some kinds of scheduling algorithm to arrange the order of forwarding packets.
- Avoid network congestion: excessive congestion will cost damages for network resource; network congestion will avoid the usage of network resource. When it detects that the congestion has a trend of rapidly growth, it will take actions to discard packets, which terminates the network overload by adjusting traffic.

Among these QoS techniques, traffic classification is the fundamental one, it will recognize packets by certain regulations and it is the precondition of operating QoS; the rest of these techniques are detailed expression of QoS by controlling network traffic and resources from different aspects.

Principle Description

1.Brief description of terms and concepts used to describe QoS

ACL

Access control lists (ACLs) classify traffic with the same characteristics. IP traffic is classified using IP ACLs, and non-IP traffic is classified using MAC ACLs. The ACL can have multiple access control entries (ACEs), which are commands that match fields against the contents of the packet.

CoS Value

Class of Service (CoS) is a 3-bit value used to classify the priority of Layer-2 frames upon entry into a network.

QoS classifies frames by assigning priority-indexed CoS values to them, and gives preference to higher-priority traffic.

Layer-2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS values in the 3 most significant bits, called the User Priority bits. On interfaces configured as Layer-2 802.1Q trunks, all traffic is in 802.1Q frames, except for traffic in the native VLAN.

Other frame types cannot carry Layer-2 CoS values. CoS values range from 0 to 7.

DSCP Value

Differentiated Services Code Point (DSCP) is a 6-bit value used to classify the priority of Layer-3 packets upon entry into a network.

DSCP values range from 0 to 63.

IP-Precedence Value

IP-Precedence is a 3-bit value used to classify the priority of Layer-3 packets upon entry into a network.

IP-Precedence values range from 0 to 7.

EXP Value

EXP value is a 3-bit value used to classify the priority of MPLS packets upon entry into a network.

MPLS EXP values range from 0 to 7.

Classification

Classification distinguishes one kind of traffic from another by examining the fields in the packet. The process generates an internal priority for a packet, which identifies all future QoS actions to be taken on the packet.

Each packet is classified upon entry into the network. At the ingress, the packet is inspected, and the priority is determined based on ACLs or the configuration. The Layer-2 CoS value is then mapped to a priority value.

The classification is carried in the IP packet header using 6 bits or 3 bits from the deprecated IP TOS field to carry the classification information. Classification can also occur in the Layer-2 frame.

Classification occurs on an ingress physical port, but not at the switch virtual interface level.

Classification can be based on CoS/inner-CoS/DSCP/IP-Precedence, default port cos, or class maps and policy maps.

Shaping

Shaping is to change the rate of incoming traffic flow to regulate the rate in such a way that the outgoing traffic flow behaves more smoothly. If the incoming traffic is highly bursty, it needs to be buffered so that the output of the buffer is less bursty and smoother.

Shaping has the following attributes:

- Shaping can be deployed base on physical port.
- Shaping can be deployed on queues of egress interface.

Policing

Policing determines whether a packet is in or out of profile by comparing the internal priority to the configured policer.

The policer limits the bandwidth consumed by a traffic flow. The result is given to the marker.

There are two types of policers:

- Individual: QoS applies the bandwidth limits specified in the policer, separately, to each matched traffic class. An individual policer is configured within a policy map.
- Aggregate: QoS applies the bandwidth limits specified in an aggregate policer, cumulatively, to all matched traffic flows. An aggregate policer is configured by specifying the policer name within a policy map. The bandwidth limits of the policer are specified. In this way, the aggregate policer is shared by multiple classes of traffic within one or multiple policy map.

Marking

Marking determines how to handle a packet when it is out of profile. It assesses the policer and the configuration information to determine the action required for the packet, and then handles the packet using one of the following methods:

- Let the packet through and mark color down
- Drop the packet

Marking can occur on ingress and egress interfaces.

Queuing

Queuing maps packets to a queue. Each egress port can accommodate up to 8 unicast queues, 1 multicast queue and 1 SPAN queue.

The packet internal priority can be mapped to one of the egress queues. The unit of queue depth is buffer cell. Buffer cell is the granularity, which is 288 bytes, for packet storing.

After the packets are mapped to a queue, they are scheduled.

Tail Drop

Tail drop is the default congestion-avoidance technique on the interface. With tail drop, packets are queued until the thresholds are exceeded. The packets with different priority and color are assigned to different drop precedence. The mapping between priority and color to queue and drop precedence is configurable. You can modify the three tail-drop threshold to every egress queue by using the queue threshold interface configuration command. Each threshold value is packet buffer cell.

WRED

Weighted Random Early Detection (WRED) differs from other congestion-avoidance techniques because it attempts to anticipate and avoid congestion, rather than controlling congestion when it occurs.

WRED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once. Thus, WRED allows the transmission line to be fully used at all times. WRED also drops more packets from large users than small. Therefore, sources that generate the most traffic are more likely to be slowed down versus sources that generate little traffic.

You can enable WRED and configure the two thresholds for a drop-precedence assigned to every egress queues. The WRED's color drop precedence map is the same as tail-drop's. Each min-threshold represents where WRED starts to randomly drop packets. After min-threshold is exceeded, WRED randomly begins to drop packets assigned to this threshold. As the queue max-threshold is approached, WRED continues to drop packets randomly with the rate of drop-probability. When the max-threshold is reached, WRED drops all packets assigned to the threshold. By default, WRED is disabled.

Scheduling

Scheduling forwards conditions packets using combination of WDRR and SP. Every queue belongs to a class. The class range from 0 to 7, and 7 is the highest priority. Several queues can be in a same class, or non queue in some class. Packets are scheduled by SP between classes and WDRR between queues in a class.

- Strict Priority-Based (SP), in which any high-priority packets are first transmitted. Lower-priority packets are transmitted only when the higher-priority queues are empty. A problem may occur when too many lower-priority packets are not transmitted.
- Weighted Deficit Round Robin (WDRR), in which each queue is assigned a weight to control the number of packets relatively sent from each queue.

Class Map

A class map names and isolates specific traffic from other traffic. The class map defines the criteria used to match against a specific traffic flow to further classify it. The criteria can match several access groups defined by the ACL.

If there is more than one type of traffic to be classified, another class map can be created under a different name. After a packet is matched against the class-map criteria, it is further classified using a policy map.

Policy Map

A policy map specifies on which traffic class to act. This can be implemented as follows:

- Set a specific priority and color in the traffic class.
- Set a specific trust policy to map priority and color.

- Specify the traffic bandwidth limitations for each matched traffic class (policer) and the action to take (marking) when the traffic is out of profile.
- Redirect the matched traffic class to a specific physical interface.
- Mirror the matched traffic class to a specific monitor session, which's destination is defined in mirror module(please refer to the "monitor session destination" command).
- Enable statistics of matching each ace or each class-map(if the class-map operator is match-any).
- Policy maps have the following attributes:
- A policy map can contain multiple class statements, each with different match criteria and action.
- A separate policy-map class can exist for each type of traffic received through an interface.
- There can be only one policy map per interface per direction. The same policy map can be applied to multiple interfaces and directions.
- Before a policy map can be effective, it must be attached to an interface.
- A policy map can be applied on physical interface(not link agg member), link agg interface, or vlan interface.

Mapping Tables

During QoS processing, the switch represents the priority of all traffic (including non-IP traffic) with an internal priority value:

- During classification, QoS uses configurable mapping tables to derive the internal priority (a 6-bit value) from received CoS, EXP(3-bit), DSCP or IP precedence (3-bit) values. These maps include the CoS-to-priority-color/COS-to-PHB map, EXP-to-priority-color/EXP-to-PHB map, DSCP-to-priority-color/DSCP-to-PHB map and the IP-precedence-to- priority-color/IP-PREC-to-PHB map.
- During policing, QoS can assign another priority and color to an IP or non-IP packet (if the packet matches the class-map). This configurable map is called the policed-priority-color map.
- Before the traffic reaches the scheduling stage, and replace CoS or DSCP is set, QoS uses the configurable priority-color-to-CoS or priority-color-to-DSCP map to derive a CoS or DSCP value from the internal priority color.
- Each QoS domain has an independent set of map tables mentioned above.

Time-range

By using time-range, the aces in the class-map can be applied based on the time of day or week. First, define a time-range name and set the times and the dates or the days of the week in the time range. Then enter the time-range name when adding an ace. You can use the time-range to define when the aces in the class-map are in effect, for example, during a specified time period or on specified days of the week.

These are some of the many possible benefits of using time-range:

- You can control over permitting or denying a user access to resources, such as an application, which is identified by an IP address and a port number.
- You can obtain the traffic statistics during appointed time.
- You can define when the action of a traffic class is in effect.

SRTCM

Single Rate Three Color Marker

TRTCM

Two Rate Three Color Marker

CIR

Committed Information Rate

CBS

Committed Burst Size

EIR

Excess Information Rate

EBS

Excess Burst Size

PIR

Peak Information Rate

PBS

Peak Burst Size

Modular QoS CLI

Input traffic is classified to a specified traffic class. All qos policies are attached to this traffic class.

class-map type qos

Type qos of class-map is used to identify traffic. The identification rules can be CoS/DSCP/IP Precendence/EXP/ACL.

policy-map type qos

Type qos of policy-map is used to assign traffic class. Type qos of class-map is refered by same type of policy-map.

class-map type traffic-class

Type traffic-class of class-map is used to identify traffic class. The identification rules is traffic class value.

policy-map type traffic-class

Type traffic-class of policy-map is used to specify qos policies. Type traffic-class of class-map is refered by same type of policy-map.

2. Brief description of terms and concepts used to describe MQC

Switch provides much simpler MQC (Modular QoS CLI) configuration patterns than the original one. The network administrators will not need to care about how to solve with the message import or export separately. They only need to configure aim at the policy map, and then apply policy map to certain interface. It reduces the requirements for network administrators and maintainers.

class map and policy map:

- Class-map: it is used for matching message and queue, and then using the results into certain policy-map to continue for next step.
- Policy-map: it is used to define a group of class-map class set, and it will set up corresponding policy for each class.

Class-map and policy-map has two types each:

- qos: it is used for techniques of traffic classification, traffic policing, associated queue and re-mark priority level and will used for the inbound of packets.
- Traffic-class: it is the related operations of setting queue, for example, traffic shaping, discard policy, schedule priority and weight.

The process of system MQC configuration is:

- Configure class-map type qos matches the message, it can be COS, DSCP, IP-Precedence or ACL.
- Apply class-map type qos into policy-map type qos. For different class-map, set up different relevant queue, traffic policing, re-mark and other operations. Those traffic classifications which do not match will classify by class-default, default associated queue 0 and cannot be setup with WDRR scheduling weight. This policy map will apply into the inbound of message.
- Configure class-map type traffic-class matching queue, support configure queue 1 to 7, 0 is reserved for system and cannot be configured by user.
- Apply class-map type traffic-class into policy-map type traffic-class. For different class-map, they set up different traffic shaping, discards type, schedule priority, weighted factor and so on. Class-default has the default value which is marked with queue 0; it does not provide operations that setting WDRR schedule weight. This policy map will apply into outbound of packets

NOTE

- The same policy-map can apply to multiple interfaces.
- The same interface only can apply to one ingress policy-map (policy-map type qos) and one egress policy-map (policy-map type traffic-class) at the same time.

11.1.2 Configuration

The following provides information to consider before configuring QoS:

- QoS policing cannot be configured on Linkagg interface.
- Traffic can be only classified per ingress port.
- There can be multiple ACLs per class map. An ACL can have multiple access control entries that match fields against the packet contents.
- Policing cannot be done at the switch virtual interface level.

To configure a QoS policy, the following is usually required:

- Categorize traffic into classes.
- Configure policies to apply to the traffic classes.
- Attach policies to interfaces.

Classify Traffic Using ACLs

IP traffic can be classified using IP ACLs. The following shows creating an IP ACL for IP traffic. Follow these steps from Privileged Exec mode.

- configure terminal.
- ip access-list ACCESS-LIST-NAME. ACCESS-LIST-NAME = name of IP ACL
- create ACEs, Repeat this step as needed. For detail, please refer to ACL configuration Guide

The no ip access-list command deletes an access list.

The following example shows allowing access only for hosts on three specified networks. Wildcard bits correspond to the network address host portions. If a host has a source address that does not match the access list statements, it is rejected.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create ACL and ACEs

```
Switch(config)# ip access-list ip-acl
Switch(config-ip-acl)# permit any 128.88.12.0 0.0.0.255 any
Switch(config-ip-acl)# permit any 28.88.0.0 0.0.255.255 any
Switch(config-ip-acl)# permit any 11.0.0.0 0.255.255.255 any
Switch(config-ip-acl)# exit
```


Use the “no ip access-list” in global configure mode to remove the ACL. Use the “no sequence-num” in ACL configure mode to remove the ACE.

Terminology:

- ACL：Access Control List
- ACE：Access Control Entry

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show access-list ip ip-acl
ip access-list ip-acl
 10 permit any 128.88.12.0 0.0.0.255 any
 20 permit any 28.88.0.0 0.0.255.255 any
 30 permit any 11.0.0.0 0.255.255.255 any
```

Create class-map

The following shows classifying IP traffic on a physical-port basis using class maps. This involves creating a class map, and defining the match criterion. In this case it is configuring a class map named cmap1 with 1 match criterion: IP access list ip-acl, which allows traffic from any source to any destination.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create ACL and ACEs

```
Switch(config)# ip access-list ip-acl
Switch(config-ip-acl)# permit any any any
Switch(config-ip-acl)# quit
```

step 3 Create class-map and match the ACL

```
Switch(config)# class-map cmap1
Switch (config-cmap)# match access-group ip-acl
Switch (config-cmap)# quit
```



- match-any keyword to perform a logical-OR of all matching statements under this class map. One or more match criteria must be matched. match-any any is the default mode.
- match-all = Use the match-all keyword to perform a logical-AND of all matching statements under this class map. All match criteria in the class map must be matched.

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show class-map cmap1
  CLASS-MAP-NAME: cmap1 (match-any)
    match access-group: ip-acl
```

Create Policy Map

The following shows creating a policy map to classify, policer, and mark traffic. In this example it is creating a policy map, and attaching it to an ingress interface. In this example, the IP ACL allows traffic from network 10.1.0.0. If the matched traffic exceeds a 48000-kbps average traffic rate, it is dropped.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create ACL and ACEs

```
Switch(config)# ip access-list ip-acl
Switch(config-ip-acl)# permit any 10.1.0.0 0.0.255.255 any
Switch(config-ip-acl)# quit
```

step 3 Create class-map and match the ACL

```
Switch(config)# class-map type qos cmap1
Switch(config-cmap)# match access-group ip-acl
Switch(config-cmap)# quit
```

step 4 Create policy-map and match the class-map; set the action in policy-class configure mode

```
switch(config)# policy-map type qos pmap1
switch(config-pmap)# class type qos cmap1
Switch(config-pmap-c)# policer color-blind cir 48000 cbs 10000 ebs 16000 violate drop
Switch(config-pmap-qos-c)# set traffic-class 5
Switch(config-pmap-qos-c)# set color yellow
Switch(config-pmap-c)# quit
Switch(config-pmap)# quit
```


Use the “no policy-map” in global configure mode to remove the policy-map. Use the “no policer” in policy-class configure mode to remove the policer, Use the “no set” in policy-class configure mode to reset the default value for priority or color.(By default the priority is 0 and color is green.)

step 5 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type qos input pmap1
Switch(config-if)# exit
```


Currently only one policy-map is supported per-direction for each interface. The “no service-policy input|output” command is used to unapply the policy map.

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show policy-map pmap1

 POLICY-MAP-NAME: pmap1 ( type qos)
  State: detached

  CLASS-MAP-NAME: cmap1
    match access-group: ip-acl
      set traffic-class      : 5
      set color              : yellow
      policer color-blind cir 48000 cbs 10000 ebs 16000 violate drop
```

Create Aggregate Policer

The following shows creating an aggregate policer to classify, police, and mark traffic. In this example it is creating an aggregate policer, and attaching it to multiple classes within a policy map. In this example, the IP ACLs allow traffic from network 10.1.0.0 and host 11.3.1.1. The traffic rate from network 10.1.0.0 and host 11.3.1.1 is policed. If the traffic exceeds a 48000-kbps average traffic rate and an 8000-byte normal burst size, it is considered out of profile, and is dropped. The policy map is attached to an ingress interface.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create ACL and ACEs

```
Switch(config)# ip access-list ip-acl1
Switch(config-ip-acl)# permit any 10.1.0.0 0.0.255.255 any
Switch(config-ip-acl)# exit
Switch(config)# ip access-list ip-acl2
Switch(config-ip-acl)# permit any host 11.3.1.1 any
Switch(config-ip-acl)# exit
```

step 3 Create an aggregate-policer

```
Switch(config)# qos aggregate-policer transmit1 color-blind cir 48000 cbs 8000 ebs 10000 violate drop
```


To delete the aggregate-policer, use the "no qos aggregate-policer" command.

step 4 Create class-map and match the ACL

```
Switch(config)# class-map type qos cmap1
Switch(config-cmap)# match access-group ip-acl1
Switch(config-cmap)# exit
Switch(config)# class-map type qos cmap2
Switch(config-cmap)# match access-group ip-acl2
Switch(config-cmap)# exit
```

step 5 Create policy-map and match the class-map; Apply the aggregate-policer in policy-class configure mode

```
Switch(config)# policy-map type qos aggflow1
Switch(config-pmap)# class type qos cmap1
Switch(config-pmap-c)# aggregate-policer transmit1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class type qos cmap2
Switch(config-pmap-c)# aggregate-policer transmit1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```


To remove the aggregate-policer, use the "no policer-aggregate" command in in policy-class configure mode.

step 6 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type qos input aggflow1
Switch(config-if)# exit
Switch(config)# exit
```

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

```
Switch# show qos aggregate-policer
Aggreate policer: transmit1
 color blind
 CIR 48000 kbps, CBS 8000 bytes, EBS 10000 bytes
 drop violate packets
```

11.1.3 Configuration for Queue

Configuring Schedule

Packets are scheduled by SP between different classes and WDRR between queues in the same class.

The following example shows configuring schedule parameters for egress queues. In this example, traffic 5 and 6 belongs to class 6, which is highest priority. Traffic 2 belongs class 0, the bandwidth is 20%.

step 1 Enter the configure mod

```
Switch# configure terminal
```

step 2 Create class-map and match the traffic-class

```
Switch(config)# class-map type traffic-class tc5
Switch(config-cmap-tc)# match traffic-class 5
Switch(config-cmap-tc)# exit

Switch(config)# class-map type traffic-class tc6
Switch(config-cmap-tc)# match traffic-class 6
Switch(config-cmap-tc)# exit

Switch(config)# class-map type traffic-class tc2
Switch(config-cmap-tc)# match traffic-class 2
Switch(config-cmap-tc)# exit
```

step 3 Create policy-map and match the class-map； Set the priority in policy-class configure mode

```
Switch(config)# policy-map type traffic-class tc
Switch(config-pmap-tc)# class type traffic-class tc5
Switch(config-pmap-tc-c)# priority level 6
Switch(config-pmap-tc-c)# exit

Switch(config-pmap-tc)# class type traffic-class tc6
Switch(config-pmap-tc-c)# priority level 6
Switch(config-pmap-tc-c)# exit

Switch(config-pmap-tc)# class type traffic-class tc2
Switch(config-pmap-tc-c)# bandwidth percentage 20
Switch(config-pmap-tc-c)# exit
Switch(config-pmap-tc)# exit
```

step 4 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class tc
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show qos interface eth-0-1 egress
TC Priority Bandwidth Shaping(kbps) Drop-Mode   Max-Queue-Limit(Cell)   ECN
0 0     -     -         dynamic    level 10            -
1 0     -     -         random-drop 596                Disable
2 0     20    -          dynamic    level 10            -
3 0     -     -         tail-drop   2000               2000
4 0     -     -         dynamic    level 10            -
5 6     -     -         dynamic    level 10            -
```

```
6 6    -    -    dynamic   level 10        -
7 7    -    -    tail-drop 64              -
```

Configuring Tail Drop

Tail drop is the default congestion-avoidance technique on every egress queue. With tail drop, packets are queued until the thresholds are exceeded. The following shows configuring tail drop threshold for different drop-precedence. Follow these steps from Privileged Exec mode.

In this example it is configuring tail drop threshold for traffic class 3. In this example, packet drop threshold is 2000.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create class-map and match the traffic-class

```
Switch(config)# class-map type traffic-class tc3
Switch(config-cmap-tc)# match traffic-class 3
Switch(config-cmap-tc)# exit
```

step 3 Create policy-map and match the class-map

```
Switch(config)# policy-map type traffic-class tc
Switch(config-pmap-tc)# class type traffic-class tc3
```

step 4 Set the threshold for tail drop in policy-class configure mode

```
Switch(config-pmap-tc-c)# queue-limit 2000
Switch(config-pmap-tc-c)# exit
Switch(config-pmap-tc)# exit
```

step 5 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class tc
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show qos interface eth-0-1 egress
TC Priority Bandwidth Shaping(kbps) Drop-Mode   Max-Queue-Limit(Cell)   ECN
0 0    -    -    dynamic   level 10        -
1 0    -    -    dynamic   level 10        -
2 0    -    -    dynamic   level 10        -
3 0    -    -    tail-drop 2000            2000
4 0    -    -    dynamic   level 10        -
5 0    -    -    dynamic   level 10        -
6 0    -    -    dynamic   level 10        -
7 7    -    -    tail-drop 64              -
```

Configuring WRED

WRED reduces the chances of tail drop by selectively dropping packets when the output interface detects congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids TCP synchronization dropping and thereafter improves the overall network throughput.

The following example shows configuring WRED threshold for traffic class 1. In this example, the max-threshold is 596, min-threshold is 596/8=71. If buffered packets exceed min-threshold, the subsequent packet will be dropped randomly.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create class-map and match the traffic-class

```
Switch(config)# class-map type traffic-class tc1
Switch(config-cmap-tc)# match traffic-class 1
Switch(config-cmap-tc)# exit
```

step 3 Create policy-map and match the class-map

```
Switch(config)# policy-map type traffic-class tc
Switch(config-pmap-tc)# class type traffic-class tc1
```

step 4 Set the threshold for WRED in policy-class configure mode

```
Switch(config-pmap-tc-c)# random-detect maximum-threshold 596
Switch(config-pmap-tc-c)# exit
Switch(config-pmap-tc)# exit
```

step 5 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class tc
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show qos interface eth-0-1 egress
TC Priority Bandwidth Shaping(kbps) Drop-Mode   Max-Queue-Limit(Cell)   ECN
0 0     -     -       dynamic    level 10          -
1 0     -     -       random-drop 596              Disable
2 0     -     -       dynamic    level 10          -
3 0     -     -       tail-drop  2000              2000
4 0     -     -       dynamic    level 10          -
5 0     -     -       dynamic    level 10          -
6 0     -     -       dynamic    level 10          -
7 7     -     -       tail-drop  64               -
```

Queue shaping

All the traffic in the egress queue can be shaped, and all the exceeding traffic will be buffered. If no buffer, it is dropped.

The following example shows creating a queue shaping for queue 3. In this example, if the traffic in queue 3 exceeds 1000Mbps, it is buffered.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create class-map and match the traffic-class

```
Switch(config)# class-map type traffic-class tc3
Switch(config-cmap-tc)# match traffic-class 3
Switch(config-cmap-tc)# exit
```

step 3 Create policy-map and match the class-map

```
Switch(config)# policy-map type traffic-class tc
Switch(config-pmap-tc)# class type traffic-class tc3
```

step 4 Set the shape rate in policy-class configure mode

```
Switch(config-pmap-tc-c)# shape rate pir 1000000
Switch(config-pmap-tc-c)# exit
Switch(config-pmap-tc)# exit
```



Use the "no shape rate" command to unset the shape rate.

step 5 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class tc
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show qos interface eth-0-1 egress
TC Priority Bandwidth Shaping(kbps) Drop-Mode   Max-Queue-Limit(Cell)   ECN
0 0     -     -         dynamic    level 10            -
1 0     -     -         random-drop 596               Disable
2 0     20     -          dynamic    level 10           -
3 0     -     1000000     tail-drop   2000              2000
4 0     -     -         dynamic    level 10            -
5 6     -     -         dynamic    level 10            -
6 6     -     -         dynamic    level 10            -
7 7     -     -         tail-drop   64                -
```

11.1.4 Configuration for Port shaping & port policing

Configuring Port policing

All traffic received or transmitted in the physical interface can be limited rate, and all the exceeding traffic will be dropped.

The following example shows creating an ingress port policer. In this example, if the received traffic exceeds a 48000-kbps average traffic rate, it is dropped.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the policer rate

```
Switch(config)# interface eth-0-1
Switch(config-if)# qos policer input color-blind cir 48000 cbs 10000 ebs 20000 violate drop
Switch(config-if)# exit
```


To remove the configuration of policer, use the "no port-policier input|output" command.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show qos interface eth-0-1 statistics policer port input
Interface: eth-0-1
 input port policer:
  color blind
   CIR 48000 kbps, CBS 10000 bytes, EBS 20000 bytes
   drop violate packets
```

Configuring Port shaping

All traffic transmitted in the physical interface can be shaped, and all the exceeding traffic will be buffered. If no buffer, it is dropped.

The following example shows creating a port shaping. In this example, if the received traffic exceeds a 1000Mbps, it is buffered.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the shape rate

```
Switch(config)# interface eth-0-1
Switch(config-if)# qos shape rate pir 1000000
Switch(config-if)# exit
```


To remove the configuration of shape, use the "no shape" command.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show running-config interface eth-0-1
Building configuration...
!
interface eth-0-1
 service-policy type traffic-class tc
 qos policer input color-blind cir 48000 cbs 10000 ebs 20000 violate drop
 qos shape rate pir 1000000
!
```

Configuring Mark ECN

All traffic transmitted in the physical interface can be shaped, and all the traffic exceeding ECN threshold will be marked.

The following example shows how to configure ecn marking on the queue. In this example, if the queue buffer exceeds 500 buffer cells, it will be marked.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create class-map and match the traffic-class

```
Switch(config)# class-map type traffic-class tc5
Switch(config-cmap-tc)# match traffic-class 5
Switch(config-cmap-tc)# exit
```

step 3 Create policy-map and match the class-map； Set the priority in policy-class configure mode

```
Switch(config)# policy-map type traffic-class tc
Switch(config-pmap-tc)# class type traffic-class tc5
Switch(config-pmap-tc-c)# queue-limit 6000 ecn-threshold 500
Switch(config-pmap-tc-c)# exit
```

step 4 Enter the interface configure mode and apply the policy-map

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class tc
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show running-config interface eth-0-1
Building configuration...
!
class-map type traffic-class tc5
 match traffic-class 5
!
policy-map type traffic-class tc
 class type traffic-class tc5
  queue-limit 6000 ecn-threshold 500
!
interface eth-0-1
 service-policy type traffic-class tc
!
```

11.1.5 Application cases

Typical application of MQC

Service Requirements

Business	CoS	Schedule	Business Features
Network control	6,7	The highest priority, guarantee 50Mbps	Few data, very important
Instant voice	5	Second priority, guarantee 200Mbps	Latency sensitively, important
Video on demand	3,4	guarantee 100Mbps	Large data, normal
Surfing the internet	2	50% of the rest bandwidth	Few data, try the best
Data transport	0,1	50% of the rest bandwidth	Large data, try the best

Configuration Steps

step 1 configure classification mode class-map, match business.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# class-map type qos network-control
switch(config-cmap-qos)# match cos 6
switch(config-cmap-qos)# match cos 7
switch(config-cmap-qos)# exit
switch(config)# class-map type qos voice
switch(config-cmap-qos)# match cos 5
switch(config-cmap-qos)# exit
switch(config)# class-map type qos video
switch(config-cmap-qos)# match cos 3
switch(config-cmap-qos)# match cos 4
switch(config-cmap-qos)# exit
switch(config)# class-map type qos internet
switch(config-cmap-qos)# match cos 2
switch(config-cmap-qos)# exit
switch(config)# class-map type qos data
switch(config-cmap-qos)# match cos 0
switch(config-cmap-qos)# match cos 1
switch(config-cmap-qos)# end
```

step 2 configure policy-map using classification mode, classify business and relevant queue

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# policy-map type qos classify
switch(config-pmap-qos)# class type qos network-control
switch(config-pmap-qos-c)# set traffic-class 6
switch(config-pmap-qos-c)# exit
switch(config-pmap-qos)# class type qos voice
switch(config-pmap-qos-c)# set traffic-class 5
switch(config-pmap-qos-c)# exit
switch(config-pmap-qos)# class type qos video
switch(config-pmap-qos-c)# set traffic-class 4
switch(config-pmap-qos-c)# exit
switch(config-pmap-qos)# class type qos internet
switch(config-pmap-qos-c)# set traffic-class 3
switch(config-pmap-qos-c)# exit
switch(config-pmap-qos)# class type qos data
switch(config-pmap-qos-c)# set traffic-class 2
switch(config-pmap-qos-c)# end
```

step 3 configure class-map using queue mode, match queues

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# class-map type traffic-class queue6
switch(config-cmap-tc)# match traffic-class 6
switch(config-cmap-tc)# exit
switch(config)# class-map type traffic-class queue5
switch(config-cmap-tc)# match traffic-class 5
switch(config-cmap-tc)# exit
switch(config)# class-map type traffic-class queue4
switch(config-cmap-tc)# match traffic-class 4
switch(config-cmap-tc)# exit
switch(config)# class-map type traffic-class queue3
switch(config-cmap-tc)# match traffic-class 3
```

```
switch(config-cmap-tc)# exit
switch(config)# class-map type traffic-class queue2
switch(config-cmap-tc)# match traffic-class 2
switch(config-cmap-tc)# end
```

step 4 configure class-map using queue mode, match queues

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# policy-map type traffic-class queuing
switch(config-pmap-tc)# class type traffic-class queue6
switch(config-pmap-tc-c)# priority level 6
switch(config-pmap-tc-c)# shape rate pir 50000
switch(config-pmap-tc-c)# exit
switch(config-pmap-tc)# class type traffic-class queue5
switch(config-pmap-tc-c)# priority level 5
switch(config-pmap-tc-c)# shape rate pir 200000
switch(config-pmap-tc-c)# exit
switch(config-pmap-tc)# class type traffic-class queue4
switch(config-pmap-tc-c)# priority level 2
switch(config-pmap-tc-c)# shape rate cir 100000 pir 100000000
switch(config-pmap-tc-c)# exit
switch(config-pmap-tc)# class type traffic-class queue3
switch(config-pmap-tc-c)# priority level 1
switch(config-pmap-tc-c)# bandwidth percentage 50
switch(config-pmap-tc-c)# exit
switch(config-pmap-tc)# class type traffic-class queue2
switch(config-pmap-tc-c)# priority level 1
switch(config-pmap-tc-c)# bandwidth percentage 50
switch(config-pmap-tc-c)# end
```

step 5 Apply classification mode policy-map to inbound interface and queue mode policy-map to outbound interface

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface eth-0-1
switch(config-if)# service-policy type qos input classify
switch(config-if)# exit
switch(config)# interface eth-0-2
switch(config-if)# service-policy type traffic-class queuing
switch(config-if)# end
```

step 6 Validation

check the classification status of business at input port

```
switch# show qos interface eth-0-1 ingress cos-policy
CoS          TC      Color    Mark-Type  Mark-Value
0            2       green    -       -
1            2       green    -       -
2            3       green    -       -
3            4       green    -       -
4            4       green    -       -
5            5       green    -       -
6            6       green    -       -
7            6       green    -       -
```

check the allocated status of schedule queues at output port

```
switch# show qos interface eth-0-2 egress
TC Priority Bandwidth Shaping(kbps) Drop-Mode   Max-Queue-Limit(Cell)   ECN
0 0    -      -        dynamic    level 10             -
1 0    -      -        dynamic    level 10             -
2 1    50     -        dynamic    level 10             -
3 1    50     -        dynamic    level 10             -
4 2    -  10000000         dynamic    level 10             -
5 5    -      200000     dynamic    level 10             -
6 6    -      50000      dynamic    level 10             -
7 7    -      -        tail-drop    64                -
```

 NOTE Discard mode and queue depth does not need to be configured to satisfy most of the requirements of business by default.

12 IPv6 Service Configuration

12.1 Configuring IPv6 over IPv4 Tunnel

12.1.1 Overview

Function Introduction

Tunneling is an encapsulation technology, which uses one network protocol to encapsulate packets of another network protocol and transfer them over a virtual point-to-point connection. The virtual connection is called a tunnel. Tunneling refers to the whole process from data encapsulation to data transfer and data decapsulation.

Principle Description

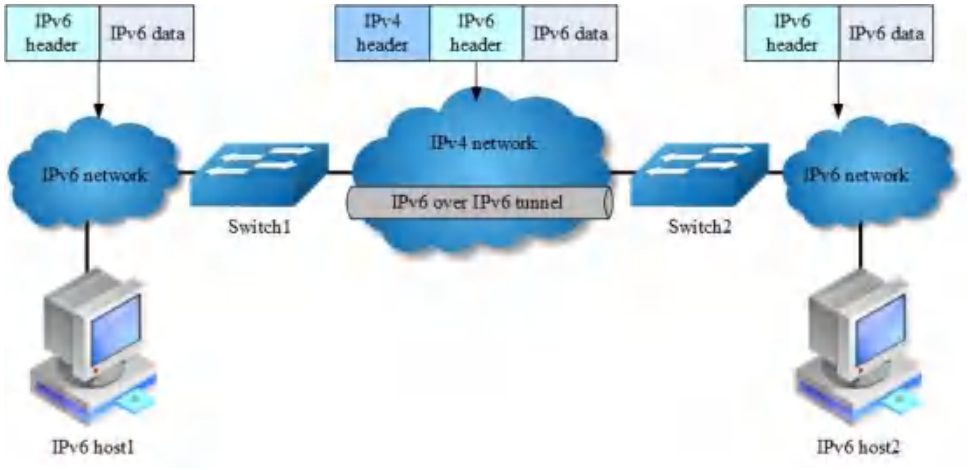


Figure 12-1 IPv6 over IPv4 Tunnel

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet. By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between a border router and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. The IPv6 over IPv4 tunnel processes packets in the following ways:

- A host in the IPv6 network sends an IPv6 packet to Switch1 at the tunnel source.
- After determining according to the routing table that the packet needs to be forwarded through the tunnel, Switch1 encapsulates the IPv6 packet with an IPv4 header and forwards it through the physical interface of the tunnel.
- Upon receiving the packet, Switch2 decapsulates the packet.
- Switch2 forwards the packet according to the destination address in the de-encapsulated IPv6 packet. If the destination address is the device itself, Switch2 forwards the IPv6 packet to the upper-layer protocol for processing.

The benefit of the technique is that current ipv4 networks do not need to update on all nodes. Only the edge nodes are required to support dual stack and tunnel.

IPv6 over IPv4 tunnels are divided into manually configured tunnels and automatic tunnels, depending on how the IPv4 address of the tunnel destination is acquired:

- Manually configured tunnel: The destination address of the tunnel cannot be automatically acquired through the destination IPv6 address of an IPv6 packet at the tunnel source, and must be manually configured.
- Automatic tunnel: The destination address of the tunnel is an IPv6 address with an IPv4 address embedded, and the IPv4 address can be automatically acquired through the destination IPv6 address of an IPv6 packet at the tunnel source.

Normally, system supports the following types of overlay tunneling mechanisms:

- Manual
- 6to4
- Intra-site Automatic Tunnel Addressing Protocol (ISATAP)

The details of the 3 types of overlay tunneling mechanisms are described below：

Manual Tunnel

A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication between two edge routers or between an end system and an edge router, or for connection to remote IPv6 networks.

An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host.

6to4 Tunnel

Ordinary 6to4 tunnel

- An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The key difference between automatic 6to4 tunnels and manually configured tunnels is that the tunnel is not point-to-point; it is point-to-multipoint. In automatic 6to4 tunnels, routers are not configured in pairs because they treat the IPv4 infrastructure as a virtual non-broadcast multi-access (NBMA) link. The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.
- An automatic 6to4 tunnel may be configured on a border router in an isolated IPv6 network, which creates a tunnel on a per-packet basis to a border router in another IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16, where the format is 2002:border-router-IPv4-address::/48.
- Following the embedded IPv4 address are 16 bits that can be used to number networks within the site. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers or between a border router and a host.

6to4 relay

A 6to4 tunnel is only used to connect 6to4 networks, whose IP prefix must be 2002::/16. However, IPv6 network addresses with the prefix such as 2001::/16 may also be used in IPv6 networks. To connect a 6to4 network to an IPv6 network, a 6to4 router must be used as a gateway to forward packets to the IPv6 network. Such a router is called 6to4 relay router.

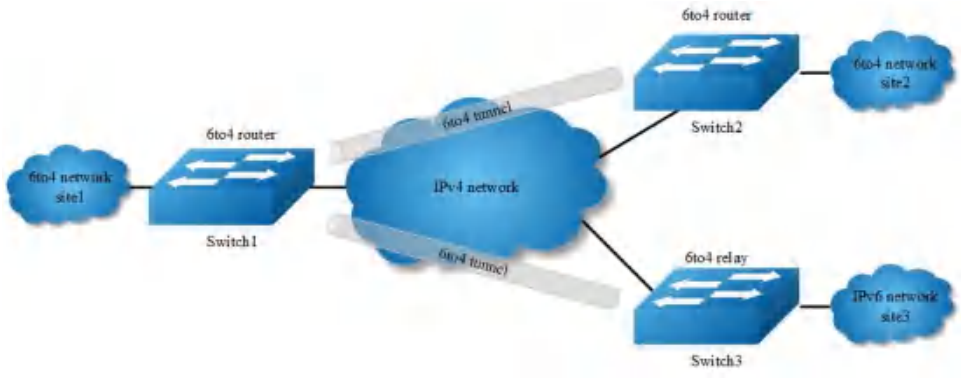


Figure 12-2 IPv6 over IPv4 Tunnel

As shown in the above figure, a static route must be configured on the border router (Switch1) in the 6to4 network and the next-hop address must be the 6to4 address of the 6to4 relay router (Switch3). In this way, all packets destined for the IPv6 network will be forwarded to the 6to4 relay router, and then to the IPv6 network. Thus, interworking between the 6to4 network (with the address prefix starting with 2002) and the IPv6 network is realized.

ISATAP Tunnel

ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6. ISATAP is designed for transporting IPv6 packets within a site where a native IPv6 infrastructure is not yet available; for example, when sparse IPv6 hosts are deployed for testing. ISATAP tunnels allow individual IPv4 or IPv6 dual-stack hosts within a site to communicate with other such hosts on the same virtual link, basically creating an IPv6 network using the IPv4 infrastructure.

When an ISATAP tunnel is used, the destination address of an IPv6 packet and the IPv6 address of a tunnel interface both adopt special ISATAP addresses. ISATAP uses a well-defined IPv6 address format composed of any unicast IPv6 prefix (/64), which can be link local, or

global (including 6to4 prefixes), enabling IPv6 routing locally or on the Internet. The IPv4 address is encoded in the last 32 bits of the IPv6 address, enabling automatic IPv6-in-IPv4 tunneling. The ISATAP address format is prefix(64bit):0:5EFE: IPv4-address.

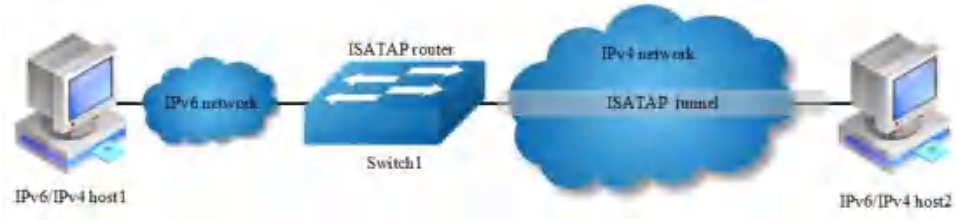


Figure 12-3 ISATAP Tunnel

The ISATAP router provides standard router advertisement network configuration support for the ISATAP site. This feature allows clients to automatically configure themselves as they would do if they were connected to an Ethernet. It can also be configured to provide connectivity out of the site.

Although the ISATAP tunneling mechanism is similar to other automatic tunneling mechanisms, such as IPv6 6to4 tunneling, ISATAP is designed for transporting IPv6 packets within a site, not between sites.

12.1.2 Configuration

Configure Manual Tunnel

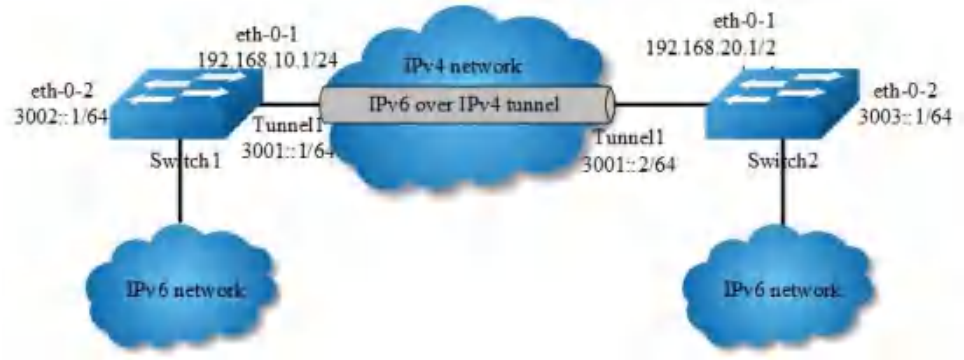


Figure 12-4 Manual Tunnel

As shown in the above Figure, two IPv6 networks are connected over an IPv4 network. Configure an IPv6 manual tunnel between Switch1 and Switch2 to make the two IPv6 networks reachable to each other.



- Must enable IPv6/IPv4 dual stack before tunnel configuration.
- Make sure tunnel destination is reachable in the IPv4 network.
- There must exist an IPv6 address in the tunnel interface, otherwise routes with tunnel interface as nexthop will be invalid.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.168.10.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 3002::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel destination 192.168.20.1
Switch(config-if)# tunnel mode ipv6ip
Switch(config-if)# ipv6 address 3001::1/64
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.168.20.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 3003::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel destination 192.168.10.1
Switch(config-if)# tunnel mode ipv6ip
Switch(config-if)# ipv6 address 3001::2/64
Switch(config-if)# exit
```

step 4 Create static routes

Configuring Switch1:

```
Switch(config)# ip route 192.168.20.0/24 192.168.10.2
Switch(config)# ipv6 route 3003::/16 tunnel1
```

Configuring Switch2:

```
Switch(config)# ip route 192.168.10.0/24 192.168.20.2
Switch(config)# ipv6 route 3002::/16 tunnel1
```

step 5 Configuring static arp

Configuring Switch1:

```
Switch(config)# arp 192.168.10.2 0.0.2222
```

Configuring Switch2:

```
Switch(config)# arp 192.168.20.2 0.0.1111
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP, Status Valid
 Tunnel source 192.168.10.1(eth-0-1), destination 192.168.20.1
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes

Switch1# show ipv6 interface tunnel1
 Interface current state: UP
 The maximum transmit unit is 1480 bytes
 IPv6 is enabled, link-local address is fe80::c0a8:a01
 Global unicast address(es):
  3001::1, subnet is 3001::/64
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ND DAD is enabled, number of DAD attempts: 1
 ND router advertisement is disabled
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 milliseconds
 ND retransmit interval is 1000 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND router advertisements max interval: 600 secs
 ND router advertisements min interval: 198 secs
 ND router advertisements live for 1800 seconds
 ND router advertisements hop-limit is 0
 Hosts use stateless autoconfig for addresses.
```

Display the result on Switch2:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP, Status Valid
 Tunnel source 192.168.20.1(eth-0-1), destination 192.168.10.1
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes
Switch1# show ipv6 interface tunnel1
 Interface current state: UP
 The maximum transmit unit is 1480 bytes
```

```
IPv6 is enabled, link-local address is fe80::c0a8:1401
Global unicast address(es):
 3001::2, subnet is 3001::/64
ICMP error messages limited to one every 1000 milliseconds
ICMP redirects are always sent
ND DAD is enabled, number of DAD attempts: 1
ND router advertisement is disabled
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements max interval: 600 secs
ND router advertisements min interval: 198 secs
ND router advertisements live for 1800 seconds
ND router advertisements hop-limit is 0
Hosts use stateless autoconfig for addresses.
```

Configure 6to4 Tunnel

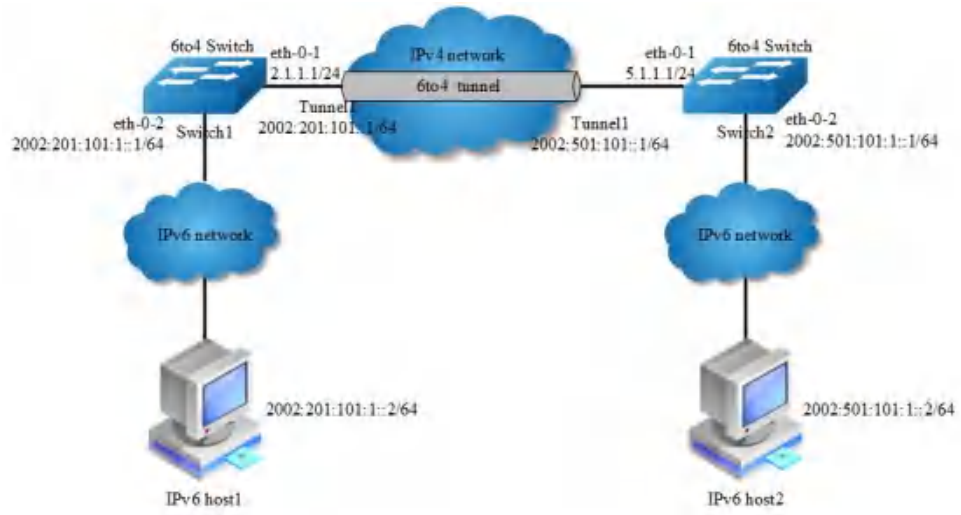


Figure 12-5 6to4 tunnel

As shown in the above Figure, two 6to4 networks are connected to an IPv4 network through two 6to4 routers (Switch1 and Switch2) respectively. Configure a 6to4 tunnel to make Host1 and Host2 reachable to each other.

To enable communication between 6to4 networks, you need to configure 6to4 addresses for 6to4 routers and hosts in the 6to4 networks.

The IPv4 address of eth-0-1 on Switch1 is 2.1.1.1/24, and the corresponding 6to4 prefix is 2002:0201:0101::/48 after it is translated to an IPv6 address. Assign interface tunnel 1 to subnet 2002:0201:0101::/64 and eth-0-2 to subnet 2002:0201:0101:1::/64.

The IPv4 address of eth-0-1 on Switch2 is 5.1.1.1/24, and the corresponding 6to4 prefix is 2002:0501:0101::/48 after it is translated to an IPv6 address. Assign interface tunnel 1 to subnet 2002:0501:0101::/64 and eth-0-2 to subnet 2002:0501:0101:1::/64.



- No destination address needs to be configured for a 6to4 tunnel
- The automatic tunnel interfaces using the same encapsulation protocol cannot share the same source IP address
- To encapsulate and forward IPv6 packets whose destination address does not belong to the network segment where the receiving tunnel interface resides, you need to configure a static route to reach the destination IPv6 address through this tunnel interface on the router. Because automatic tunnels do not support dynamic routing, you can configure a static route to that destination IPv6 address with this tunnel interface as the outbound interface or the peer tunnel interface address as the next hop
- Only on4 6to4 tunnel can exist in the same node.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 2.1.1.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2002:201:101:1::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel mode ipv6ip 6to4
Switch(config-if)# ipv6 address 2002:201:101::1/64
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 5.1.1.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2002:501:101:1::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel mode ipv6ip 6to4
Switch(config-if)# ipv6 address 2002:501:101::1/64
Switch(config-if)# exit
```

step 4 Create static routes

Configuring Switch1:

```
Switch(config)# ip route 5.1.1.0/24 2.1.1.2
Switch(config)# ipv6 route 2002::/16 tunnel1
```

Configuring Switch2:

```
Switch(config)# ip route 2.1.1.0/24 5.1.1.2
Switch(config)# ipv6 route 2002::/16 tunnel1
```

step 5 Configuring static arp

Configuring Switch1:

```
Switch(config)# arp 2.1.1.2 0.0.2222
```

Configuring Switch2:

```
Switch(config)# arp 5.1.1.2 0.0.1111
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch1# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP 6to4, Status Valid
 Tunnel source 2.1.1.1(eth-0-1), destination UNKNOWN
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes
```

Display the result on Switch2:

```
Switch2# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP 6to4, Status Valid
 Tunnel source 5.1.1.1(eth-0-1), destination UNKNOWN
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes
```

Configure 6to4 relay

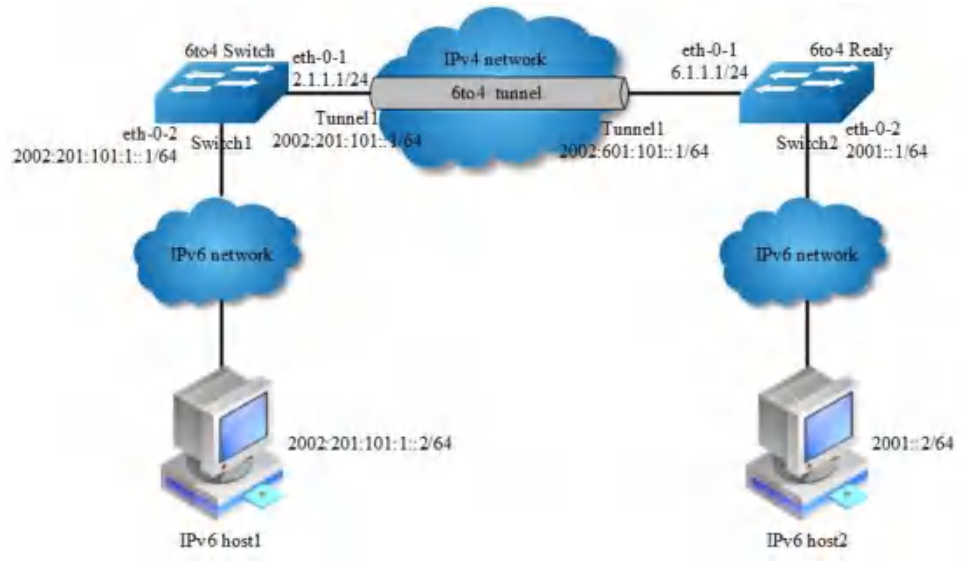


Figure 12-6 6to4 relay

As shown in the above Figure, Switch1 is a 6to4 router, and 6to4 addresses are used on the connected IPv6 network. Switch2 serves as a 6to4 relay router and is connected to the IPv6 network (2001::/16). Configure a 6to4 tunnel between Router A and Router B to make Host A and Host B reachable to each other.

- The configuration on a 6to4 relay router is similar to that on a 6to4 router. However, to enable communication between the 6to4 network and the IPv6 network, you need to configure a route to the IPv6 network on the 6to4 router.
- It is not allowed to change the tunnel mode from 6to4 to ISATAP when there are any 6to4 relay routes existing. You must delete this route first.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 2.1.1.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2002:201:101:1::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel mode ipv6ip 6to4
Switch(config-if)# ipv6 address 2002:201:101::1/64
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 5.1.1.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2002:501:101:1::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel mode ipv6ip 6to4
Switch(config-if)# ipv6 address 2002:501:101::1/64
Switch(config-if)# exit
```

step 4 Create static routes

Configuring Switch1:

```
Switch(config)# ip route 6.1.1.0/24 2.1.1.2
Switch(config)# ipv6 route 2001::/16 2002:601:101::1
Switch(config)# ipv6 route 2002:601:101::/48 tunnel1
```

Configuring Switch2:

```
Switch(config)# ip route 2.1.1.0/24 6.1.1.2
Switch(config)# ipv6 route 2002::/16 tunnel1
```

step 5 Configuring static arp

Configuring Switch1:

```
Switch(config)# arp 2.1.1.2 0.0.2222
```

Configuring Switch2:

```
Switch(config)# arp 6.1.1.2 0.0.1111
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
```

```
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP 6to4, Status Valid
 Tunnel source 2.1.1.1(eth-0-1), destination UNKNOWN
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes

Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP
    [*] - [AD/Metric]
Timers: Uptime

S     2001::/16 [1/0]
    via 2002:601:101::1 (recursive via ::, tunnel1), 00:00:32
C     2002:201:101::/64
    via ::, tunnel1, 00:00:04
C     2002:201:101::1/128
    via ::1, tunnel1, 00:00:04
S     2002:601:101::/48 [1/0]
    via ::, tunnel1, 00:00:22

Switch# show ipv6 interface tunnel1
Interface tunnel1
 Interface current state: UP
 The maximum transmit unit is 1480 bytes
 IPv6 is enabled, link-local address is fe80::201:101
 Global unicast address(es):
  2002:201:101::1, subnet is 2002:201:101::/64
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ND DAD is enabled, number of DAD attempts: 1
 ND router advertisement is disabled
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 milliseconds
 ND retransmit interval is 1000 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND router advertisements max interval: 600 secs
 ND router advertisements min interval: 198 secs
 ND router advertisements live for 1800 seconds
 ND router advertisements hop-limit is 0
 Hosts use stateless autoconfig for addresses.
```

Display the result on Switch2:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP 6to4, Status Valid
 Tunnel source 6.1.1.1(eth-0-1), destination UNKNOWN
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes
```

Configure ISATAP Tunnel

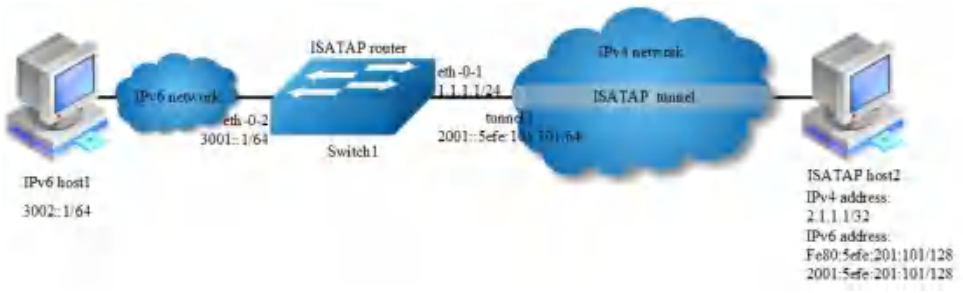


Figure 12-7 ISATAP tunnel

As shown in the above Figure, an IPv6 network is connected to an IPv4 network through an ISATAP router. It is required that the IPv6 host in the IPv4 network can access the IPv6 network through the ISATAP tunnel.

- No destination address needs to be configured for a ISATAP tunnel
- The automatic tunnel interfaces using the same encapsulation protocol cannot share the same source IP address
- To encapsulate and forward IPv6 packets whose destination address does not belong to the network segment where the receiving tunnel interface resides, you need to configure a static route to reach the destination IPv6 address through this tunnel interface on the router. Because automatic tunnels do not support dynamic routing, you can configure a static route to that destination IPv6 address with this tunnel interface as the outbound interface or the peer tunnel interface address as the next hop

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 3001::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface tunnel1
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel mode ipv6ip isatap
Switch(config-if)# ipv6 address 2001::/64 eui-64
Switch(config-if)# no ipv6 nd ra suppress

Switch(config-if)# exit
```

step 4 Create static routes

```
Switch(config)# ip route 2.1.1.0/24 1.1.1.2
Switch(config)# ipv6 route 2001::/16 tunnel1
```

step 5 Configuring static arp

```
Switch(config)# arp 1.1.1.2 0.0.2222
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport IPv6/IP ISATAP, Status Valid
 Tunnel source 1.1.1.1(eth-0-1), destination UNKNOWN
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel transport MTU 1480 bytes

Switch# show ipv6 interface tunnel1
Interface tunnel1
 Interface current state: UP
 The maximum transmit unit is 1480 bytes
 IPv6 is enabled, link-local address is fe80::101:101
 Global unicast address(es):
  2001::101:101, subnet is 2001::/64 [EUI]
 ICMP error messages limited to one every 1000 milliseconds
 ICMP redirects are always sent
 ND DAD is enabled, number of DAD attempts: 1
 ND router advertisement is enabled
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 milliseconds
 ND retransmit interval is 1000 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND router advertisements max interval: 600 secs
 ND router advertisements min interval: 198 secs
 ND next router advertisement due in 359 secs.
 ND router advertisements live for 1800 seconds
 ND router advertisements hop-limit is 0
 Hosts use stateless autoconfig for addresses.
```

Configure ISATAP host

The specific configuration on the ISATAP host is related to its operating system. The following example shows the configuration of the host running the Windows XP.

Install IPv6.

```
C:\>ipv6 install
```

On a Windows XP-based host, the ISATAP interface is usually interface 2. Configure the IPv4 address of the ISATAP router on interface 2 to complete the configuration on the host. Before that, display information on the ISATAP interface:

```
Interface 2: Automatic Tunneling Pseudo-Interface
 Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}
```

```
does not use Neighbor Discovery
does not use Router Discovery
routing preference 1
EUI-64 embedded IPv4 address: 0.0.0.0
router link-layer address: 0.0.0.0
  preferred link-local fe80::5efe:2.1.1.1, life infinite
link MTU 1280 (true link MTU 65515)
current hop limit 128
reachable time 25000ms (base 30000ms)
retransmission interval 1000ms
DAD transmits 0
default site prefix length 48
```

A link-local address (fe80::5efe:2.1.1.2) in the ISATAP format was automatically generated for the ISATAP interface. Configure the IPv4 address of the ISATAP router on the ISATAP interface.

```
C:\>ipv6 rlu 2 1.1.1.1
```

After carrying out the above command, look at the information on the ISATAP interface.

```
Interface 2: Automatic Tunneling Pseudo-Interface
 Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}
 does not use Neighbor Discovery
 does not use Router Discovery
 routing preference 1
 EUI-64 embedded IPv4 address: 2.1.1.1
 router link-layer address: 1.1.1.1
preferred global 2001::5efe:2.1.1.1, life 29d23h59m46s/6d23h59m46s (public)
   preferred link-local fe80::5efe:2.1.1.1, life infinite
 link MTU 1280 (true link MTU 65515)
 current hop limit 128
 reachable time 25000ms (base 30000ms)
 retransmission interval 1000ms
 DAD transmits 0
 default site prefix length 48
```

12.1.3 Application cases

N/A

12.2 Configuring ND

12.2.1 Overview

Function Introduction

Nodes (hosts and routers) use Neighbor Discovery to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid.

Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf.

Finally, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses. When a router or the path to a router fails, a host actively searches for functioning alternates.

Principle Description

N/A

12.2.2 Configuration

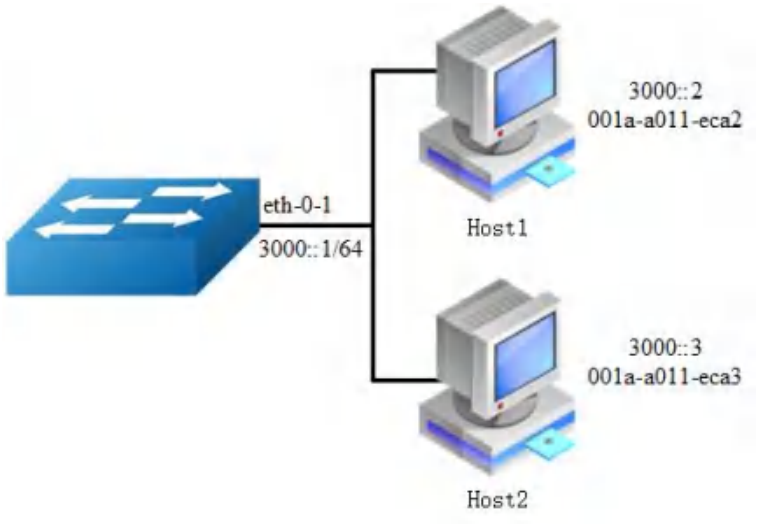


Figure 12-8 NDP

In this example, interface eth-0-1 assigned with ipv6 address 3000::1/64, on subnet 3000::/64, there are two hosts, and their IP addresses are 3000::2, 3000::3, MAC address are 001a-a011-eca2, 001a-a011-eca3. Neighbor entry of host 3000::2 is added manually, the entry of host 3000::3 is added dynamically. The reachable time of neighbor entries for interface eth-0-1 configure to 10 minutes, NS interval on interface eth-0-1 configure to 2 seconds.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

```
Switch (config)# interface eth-0-1
Switch (config-if)# no switchport
Switch (config-if)# no shutdown
Switch (config-if)# ipv6 address 3000::1/64
Switch (config-if)# ipv6 nd reachable-time 600
Switch (config-if)# ipv6 nd ns-interval 2000
Switch (config-if)# exit
```

step 3 Add a static neighbor entry

```
Switch (config)# ipv6 neighbor 3000::2 001a.a011.eca2
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch # show ipv6 neighbors
IPv6 address                  Age     Link-Layer Addr State Interface
3000::2                       -       001a-a011-eca2  REACH eth-0-1
3000::3                       6       001a-a011-eca3  REACH eth-0-1
fe80::6d8:e8ff:fe4c:e700              6       001a-a011-eca3  STALE eth-0-1
```

12.3 Configuring DHCPv6 Relay

12.3.1 Overview

Function Introduction

DHCPv6 relay is any host that forwards DHCPv6 packets between clients and servers. Relay is used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay forwarding is distinct from the normal forwarding of an IPv6 router, where IPv6 datagram are switched between networks somewhat transparently.

By contrast, relay receive DHCPv6 messages and then generate a new DHCPv6 message to send out on another interface. The relay sets the link address (used by server to identify the subnet that client is belong to), and, if configured, adds the remote-id option in the packet and forwards it to the DHCPv6 server..

Principle Description

N/A

12.3.2 Configuration

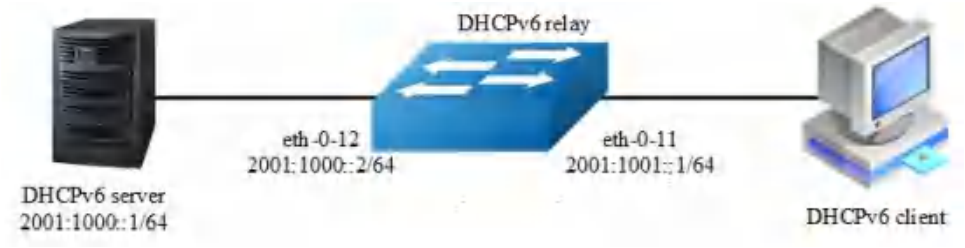


Figure 12-9 DHCP Relay

This figure is the networking topology for testing DHCPv6 relay functions. We need two Linux boxes and one Switch to construct the test bed.

- Computer A is used as DHCPv6 server.
- Computer B is used as DHCPv6 client.
- Switch is used as DHCPv6 relay.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable DHCPv6 relay globally

```
Switch(config)# service dhcpv6 enable
Switch(config)# dhcpv6 relay
Switch(config)# dhcpv6 relay remote-id option
Switch(config)# dhcpv6 relay pd route
```

step 3 Configure the DHCPv6 server

```
Switch(config)# dhcpv6-server 1 2001:1000::1
```

step 4 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-12
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1000::2/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-11
```

```
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1001::1/64
Switch(config-if)# no shutdown
Switch(config-if)# dhcpv6-server 1
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Check the interface configuration

```
Switch# show running-config interface eth-0-12
!
interface eth-0-12
 no switchport
 ipv6 address 2001:1000::1/64
!

Switch # show running-config interface eth-0-11
!
interface eth-0-11
 no switchport
 ipv6 address 2001:1001::1/64
 dhcpv6-server 1
!
```

Check the dhcpv6 service status

```
Switch# show services
Networking services configuration:
Service Name       Status
============================================================
dhcp               disable
dhcpv6             enable
```

Check the dhcpv6 server group configuration

```
Switch# show dhcpv6-server
DHCPv6 server group information:
============================================================
group 1 ipv6 address list:
[1] 2001:1000::1
```

Check the dhcpv6 relay statistics。

```
Switch# show dhcpv6 relay statistics
DHCPv6 relay packet statistics:
============================================================
Client relayed packets : 8
Server relayed packets : 8

Client error packets  : 0
Server error packets  : 0
```

Check the prefix-delegation client information learning by DHCPv6 relay

```
Switch# show dhcpv6 relay pd client
DHCPv6 prefix-delegation client information:
============================================================
Interface  : eth-0-11
Client DUID  : 000100011804ff38c2428f04970
Client IPv6 address  : fe80::beac:d8ff:fedf:c600
  IA ID  : d8dfc60
    IA Prefix :  2002:2:9:eebe::/64
      prefered/max lifetime : 280/300
      expired time :  2001-1-1 09:10:58
============================================================
```

12.3.3 Application cases

N/A

12.4 Configuration DHCPv6 Server

12.4.1 Overview

Function Introduction

The IPv6 protocol has the characteristic of a huge address space, but at the same time, the 128 bit IPv6 address requires efficient and reasonable address automatic allocation and management strategies. There are currently several methods for assigning IPv6 addresses:

- Manual configuration. Manually configure IPv6 address/prefix and other network parameters (DNS, NIS, SNTP server address, etc.).
- Automatic address allocation without status. Generate the link-local address from the interface ID, and automatically configure the local address based on the prefix information contained in the router advertisement message RA.
- State based automatic address allocation, also known as DHCPv6 mode.

DHCPv6 （Dynamic Host Configuration Protocol for IPv6） It is a protocol that runs between the client(s) and server, and all protocol packets are based on UDP. Due to the absence of broadcast packets in IPv6, DHCPv6 uses multicast packets, and clients do not need to configure the IPv6 address of the server. According to different configuration methods, DHCPv6 can be divided into two modes: stateful and stateless.

Stateful DHCPv6

The DHCPv6 server automatically assigns IPv6 addresses/PD prefixes and other network configuration parameters (such as DNS, NIS, SNTP server addresses, etc.). In this mode, the server allocates a unique IPv6 address from the address pool to the client and can also provide other network parameters such as DNS servers, domain names, etc.

Working principle:

- The client searches for a DHCPv6 server by sending a Solicit message.
- After receiving the request, the server responds with an Advertisement message to announce the service.
- The client sends a Request message requesting specific address and parameters.
- The server allocates resources and replies with reply messages to complete the allocation of addresses and parameters.

State management:

Server maintenance and management of address pools, recording the address allocation status of each client.

Applicable scenarios:

Network environments that require precise control over address allocation, such as enterprise networks, virtual machine hosting, etc.

Stateless DHCPv6

In this mode, the client's IPv6 address is still automatically generated through route advertising, and the DHCPv6 server only assigns configuration parameters other than the IPv6 address, including DNS, NIS, SNTP server, and other parameters. The client generates an

IPv6 address based on the prefix information in the Router Advertisement (RA) message using SLAAC (stateless address auto configuration).

Working principle:

The client generates a globally unique IPv6 address based on the IPv6 prefix and its own MAC address in the RA message sent by the router. DHCPv6 servers do not participate in the address allocation process.

State management:

The server does not maintain address pools or address allocation status for clients.

Applicable scenarios:

Used in conjunction with SLAAC, it is suitable for mobile devices or dynamic networks, where address flexibility is more important than fixed allocation.

12.4.2 Configuration

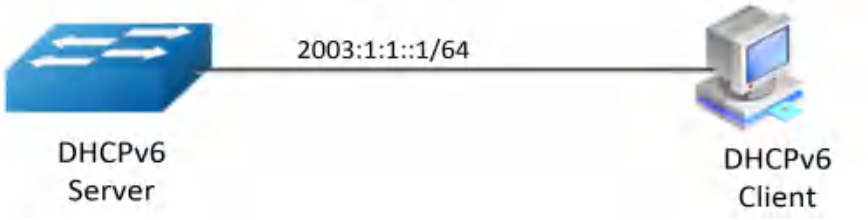


Figure 12-10 DHCPv6 Server

The above figure shows the switch acting as a DHCPv6 Server to allocate IPv6 addresses to the PC.

step 1 Enter the configure mode

```
DHCP-Server# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
DHCP-Server(config)#
```

step 2 Enable IPv6 globally and configure the IPv6 interface.

Enable global IPv6 and configure the IPv6 address of the DHCPv6 server interface. Cancel the suppression of switch publishing RA messages. The configuration flag of the managed address is set to 1, which means that the host obtains the IPv6 address through the DHCPv6 server. Configure other information to set the flag to 1, which means that the host obtains information other than the IPv6 address through the DHCPv6 server

```
DHCP-Server(config)# ipv6 enable
DHCP-Server(config)# vlan database
DHCP-Server(config-vlan)# vlan  11
DHCP-Server(config)# interface eth-0-32
DHCP-Server(config-if)# switchport access vlan 11
DHCP-Server(config-if)# exit
DHCP-Server(config)# interface vlan11
DHCP-Server(config-if)# ipv6 address 2003:1:1::1/64
DHCP-Server(config-if)# ipv6 nd managed-config-flag
DHCP-Server(config-if)# ipv6 nd other-config-flag
DHCP-Server(config-if)# no ipv6 nd ra suppress
```

step 3 Configure IPv6 address pool

```
DHCP-Server(config)# dhcpv6 pool test1
DHCP-Server(config-dhcpv6)# network 2003:1:1::/64
DHCP-Server(config-dhcpv6)# excluded-address 2003:1:1::1
```

```
DHCP-Server(config-dhcpv6)# dns-domain-name hohunet.com
DHCP-Server(config-dhcpv6)# dns-server 2003:1:1::1
DHCP-Server(config-dhcpv6)# exit
DHCP-Server(config)#
```

step 4 enable DHCPv6service

```
DHCP-Server(config)# service dhcpv6 enable
DHCP-Server(config)# dhcpv6 server
DHCP-Server(config)# interface vlan11
DHCP-Server(config-if)# dhcpv6 server enable
DHCP-Server(config-if)# dhcpv6 server test1
```

step 5 Validation

Display DHCPv6configuration

```
Switch# show dhcpv6 server interfaces
List of DHCPv6 server enabled interface(s):
DHCPv6 server service status: enabled
Interface Name
============================================================
vlan11
Switch#
Switch# show dhcpv6 server pool
 Pool : test1
  NA-Network            : 2003:1:1::/64
    Number of Active Leases  : 2
    Number of Conflict Leases: 0
  Lifetime valid          : 172800 seconds
  Lifetime preferred      : 86400 seconds
  Information refresh time  : 86400 seconds

Switch#
```

Display the address allocation status of DHCPv6 Server

```
Switch# show dhcpv6 server pool test1 allocated address

Address                          State    Valid    Expires          Left
-------------------------------------------------------------------------------------------------------
2003:1:1:0:5df6:b600:2e4a:d481              active   172800   2025/10/14 22:27:20  160700
2003:1:1:0:31a4:f422:9078:d626              active   172800   2025/10/15 01:46:46  172666
-------------------------------------------------------------------------------------------------------
Total: 2
Switch#
```

13 IPv6 Security Configuration Guide

13.1 DHCPv6 Snooping Configuration

13.1.1 Overview

Function Introduction

DHCPv6 snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCPv6 servers. The DHCPv6 snooping feature performs the following activities:

- Validate DHCPv6 messages received from untrusted sources and filters out invalid messages.
- Build and maintain the DHCPv6 snooping binding database, which contains information about untrusted hosts with leased IPv6 addresses.
- The DHCPv6 snooping feature is implemented in software basis. All DHCPv6 messages are intercepted in the chip and directed to the CPU for processing.

Principle Description

N/A

13.1.2 Configuration

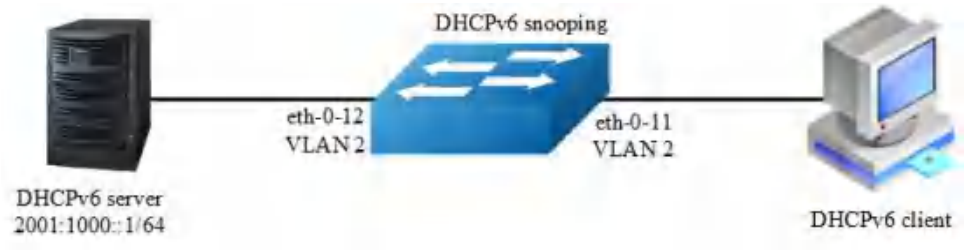


Figure 13-1 DHCPv6 Snooping

This figure is the networking topology for testing DHCPv6 snooping functions. We need two PCs and one switch to construct the test bed.

- PC A is used as a DHCPv6 server.
- PC B is used as a DHCPv6 client.
- Switch A is used as a DHCPv6 Snooping device.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-11
Switch(config-if)# switchport
Switch(config-if)# switchport access vlan 2
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface eth-0-12
Switch(config-if)# switchport
Switch(config-if)# switchport access vlan 2
Switch(config-if)# dhcpv6 snooping trust
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 4 Enable DHCPv6 snooping globally and set the attributes

```
Switch(config)# service dhcpv6 enable
Switch(config)# dhcpv6 snooping
Switch(config)# dhcpv6 snooping vlan 2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Check the interface configuration.

```
Switch# show running-config interface eth-0-12
!
interface eth-0-12
switchport access vlan 2
dhcpv6 snooping trust

!
Switch# show running-config interface eth-0-11
!
interface eth-0-11
 switchport access vlan 2
!
```

Check the dhcpv6 service status.

```
Switch# show services
Networking services configuration:
Service Name       Status
============================================================
dhcp               disable
dhcpv6             enable
```

Show dhcpv6 snooping statistics.

```
Switch# show dhcpv6 snooping config
dhcpv6 snooping service: enabled
dhcpv6 snooping switch: enabled
dhcpv6 snooping vlan 2
```

Enable DHCPv6 snooping global feature

```
Switch# show dhcpv6 snooping statistics
DHCPv6 snooping statistics:
============================================================
DHCPv6 packets                 21

Packets forwarded              21
Packets invalid                0
Packets dropped                0
```

Step 5 Show dhcpv6 snooping binding information

```
Switch# show dhcpv6 snooping binding all
DHCPv6 snooping binding table:
VLAN MAC Address    Lease(s)    Interface   IPv6 Address
============================================================
2    0016.76a1.7ed9 978         eth-0-11    2001:1000::2
```

14 IPv6 Routing Configuration

14.1 Configuring IPv6 Unicast-Routing

14.1.1 Overview

Function Introduction

Static routing is a concept describing one way of configuring path selection of routers in computer networks. It is the type of routing characterized by the absence of communication between routers regarding the current topology of the network. This is achieved by manually adding routes to the routing table. The opposite of static routing is dynamic routing, sometimes also referred to as adaptive routing.

In these systems, routes through a data network are described by fixed paths (statically). These routes are usually entered into the router by the system administrator. An entire network can be configured using static routes, but this type of configuration is not fault tolerant. When there is a change in the network or a failure occurs between two statically defined nodes, traffic will not be rerouted. This means that anything that wishes to take an affected path will either have to wait for the failure to be repaired or the static route to be updated by the administrator before restarting its journey. Most requests will time out (ultimately failing) before these repairs can be made. There are, however, times when static routes can improve the performance of a network. Some of these include stub networks and default routes.

Principle Description

N/A

14.1.2 Configuration



Figure 14-1 ipv6 unicast routing

The following example shows how to deploy static routes in a simple environment.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address auto link-local
Switch(config-if)# ipv6 address 2001:1::1/64
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address auto link-local
Switch(config-if)# ipv6 address 2001:1::2/64
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address auto link-local
Switch(config-if)# ipv6 address 2001:2::2/64
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address auto link-local
Switch(config-if)# ipv6 address 2001:2::3/64
Switch(config-if)# exit
```

step 4 Create static routes

Configuring Switch1:

```
Switch(config)# ipv6 route 2001:2::/64 2001:1::2
```

Configuring Switch3:

```
Switch(config)# ipv6 route 2001:1::/64 2001:2::2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP
    [*] - [AD/Metric]
Timers: Uptime
C     2001:1::/64
    via ::, eth-0-9, 02:08:50
C     2001:1::1/128
    via ::1, eth-0-9, 02:08:50
S     2001:2::/64 [1/0]
    via 2001:1::2, eth-0-9, 02:05:36
C     fe80::/10
    via ::, Null0, 02:09:11
```

Display the result on Switch2:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP
    [*] - [AD/Metric]
```

```
Timers: Uptime
C     2001:1::/64
      via ::, eth-0-9, 00:03:37
C     2001:1::2/128
      via ::1, eth-0-9, 00:03:37
C     2001:2::/64
      via ::, eth-0-17, 00:03:21
C     2001:2::2/128
      via ::1, eth-0-17, 00:03:21
C     fe80::/10
      via ::, Null0, 00:03:44
```

Display the result on Switch3:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP
    [*] - [AD/Metric]
Timers: Uptime
S     2001:1::/64 [1/0]
      via 2001:2::2, eth-0-17, 00:02:14
C     2001:2::/64
      via ::, eth-0-17, 00:03:28
C     2001:2::3/128
      via ::1, eth-0-17, 00:03:28
C     fe80::/10
      via ::, Null0, 00:03:53
```

Use the "ping" command on switch1 to contact the switch3:

```
Switch1# ping ipv6 2001:2::3
PING 2001:2::3(2001:2::3) 56 data bytes
64 bytes from 2001:2::3: icmp_seq=0 ttl=63 time=127 ms
64 bytes from 2001:2::3: icmp_seq=1 ttl=63 time=132 ms
64 bytes from 2001:2::3: icmp_seq=2 ttl=63 time=124 ms
64 bytes from 2001:2::3: icmp_seq=3 ttl=63 time=137 ms
64 bytes from 2001:2::3: icmp_seq=4 ttl=63 time=141 ms
--- 2001:2::3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 124.950/132.719/141.251/5.923 ms, pipe 2
```

14.1.3 Application cases

N/A

14.2 Configuring OSPFv3

14.2.1 Overview

Function Introduction

OSPF is an Interior Gateway Protocol (IGP) designed expressly for IP networks, supporting IP subnet ting and tagging of externally derived routing information.

The implementation conforms to the OSPF Version 3, which is described in RFC 5340, expands on OSPF version 2 to support IPv6 routing prefixes. Much of the OSPF for IPv6 feature is the same as in OSPF version 2. Changes between OSPF for IPv4, OSPF Version 2, and OSPF for IPv6 as described herein include the following:

- Addressing semantics have been removed from OSPFv3 packets and the basic Link State Advertisements (LSAs).

- OSPFv3 now runs on a per-link basis rather than on a per-IP-subnet basis.
- Authentication has been removed from the OSPFv3 protocol.

Principle Description

The OSPFv3 module is based on the following RFC: RFC 5340 – OSPF for IPv6

14.2.2 Configuration

Basic OSPFv3 Parameters Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create OSPFv3 instance

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Use the command "no router ipv6 ospf process-id" in global configure mode to delete the OSPFv3 instance.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ipv6 protocols
 Routing Protocol is "OSPFv3 (100)" with ID 1.1.1.1
  Redistributing:
  Routing for Networks:
  Distance: (default is 110)
```

Enabling OSPFv3 on an Interface

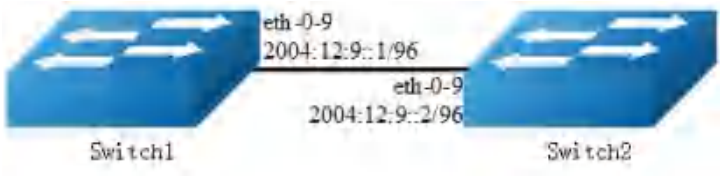


Figure 14-2 OSPFv3

This example shows the minimum configuration required for enabling OSPFv3 on an interface Switch1 and 2 are two routers in Area 0 connecting to prefix 2004:12:9::/96.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create OSPFv3 instance

Configuring Switch1:

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ipv6 ospf 200
Switch(config-router)# router-id 2.2.2.2
Switch(config-router)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::1/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::2/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Display the result on Switch1:

```
Switch# show ipv6 ospf database
        OSPFv3 Router with ID (1.1.1.1) (Process 100)
          Link-LSA (Interface eth-0-9)
Link State ID  ADV Router     Age  Seq#      CkSum  Prefix
0.0.0.9        1.1.1.1        614 0x80000001 0x6a40     1
0.0.0.9        2.2.2.2         68 0x80000001 0x4316     1
          Router-LSA (Area 0.0.0.0)
Link State ID  ADV Router     Age  Seq#      CkSum   Link
0.0.0.0        1.1.1.1         54 0x80000003 0xb74b     1
0.0.0.0        2.2.2.2         55 0x80000003 0x9965     1
          Network-LSA (Area 0.0.0.0)
Link State ID  ADV Router     Age  Seq#      CkSum
0.0.0.9        1.1.1.1         54 0x80000001 0x3ed1
          Intra-Area-Prefix-LSA (Area 0.0.0.0)
Link State ID  ADV Router     Age  Seq#      CkSum  Prefix  Reference
0.0.0.2        1.1.1.1         53 0x80000001 0x450a     1  Network-LSA

Switch# show ipv6 ospf neighbor
OSPFv3 Process (100)
Neighbor ID     Pri   State         Dead Time   Interface  Instance ID
```

```
2.2.2.2         1  Full/Backup     00:00:33    eth-0-9   0

Switch# show ipv6 ospf route
OSPFv3 Process (100)
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
     E1 - OSPF external type 1, E2 - OSPF external type 2
  Destination                          Metric
    Next-hop
C  2004:12:9::/96                          1
    directly connected, eth-0-9, Area 0.0.0.0
```

Display the result on Switch2:

```
Switch# show ipv6 ospf database
         OSPFv3 Router with ID (2.2.2.2) (Process 200)
            Link-LSA (Interface eth-0-9)
Link State ID   ADV Router     Age  Seq#      CkSum  Prefix
0.0.0.9         1.1.1.1         774 0x80000001 0x6a40     1
0.0.0.9         2.2.2.2         228 0x80000001 0x4316     1
            Router-LSA (Area 0.0.0.0)
Link State ID   ADV Router     Age  Seq#      CkSum    Link
0.0.0.0         1.1.1.1         217 0x80000003 0xb74b     1
0.0.0.0         2.2.2.2         214 0x80000003 0x9965     1
            Network-LSA (Area 0.0.0.0)
Link State ID   ADV Router     Age  Seq#      CkSum
0.0.0.9         1.1.1.1         215 0x80000001 0x3ed1
            Intra-Area-Prefix-LSA (Area 0.0.0.0)
Link State ID   ADV Router     Age  Seq#      CkSum  Prefix  Reference
0.0.0.2         1.1.1.1         214 0x80000001 0x450a     1  Network-LSA

Switch# show ipv6 ospf neighbor
OSPFv3 Process (200)
Neighbor ID     Pri   State           Dead Time   Interface  Instance ID
1.1.1.1           1   Full/DR         00:00:35    eth-0-9   0

Switch# show ipv6 ospf route
OSPFv3 Process (200)
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
     E1 - OSPF external type 1, E2 - OSPF external type 2
  Destination                          Metric
    Next-hop
C  2004:12:9::/96                          1
    directly connected, eth-0-9, Area 0.0.0.0
```

Configuring Priority

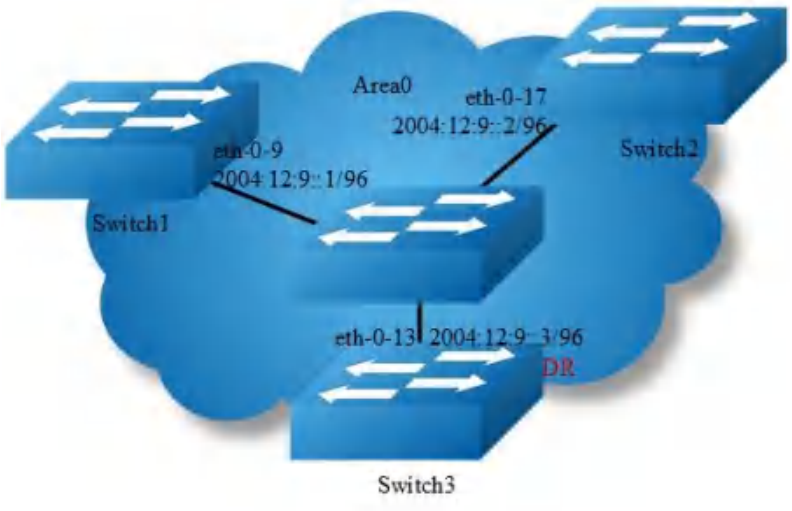


Figure 14-3 OSPFv3 priority

This example shows the configuration for setting the priority for an interface. You can set a high priority for a router to make it the Designated Router (DR). Router Switch3 is configured to have a priority of 10, which is higher than the default priority (default priority is 1) of Switch1 and 2; making it the DR.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create OSPFv3 instance

Configuring Switch1:

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ipv6 ospf 200
Switch(config-router)# router-id 2.2.2.2
Switch(config-router)# exit
```

Configuring Switch3:

```
Switch(config)# router ipv6 ospf 300
Switch(config-router)# router-id 3.3.3.3
Switch(config-router)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::1/96
```

```
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::2/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::3/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# ipv6 ospf priority 10
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show ipv6 ospf neighbor
OSPFv3 Process (100)
Neighbor ID     Pri   State           Dead Time   Interface  Instance ID
2.2.2.2           1   Full/Backup     00:00:31    eth-0-9    0
3.3.3.3          10   Full/DR         00:00:36    eth-0-9    0
Switch#
Switch# show ipv6
interface   isis      mif       mld       mroute     mroute-rpf
multicast   neighbors   ospf      pim       prefix-list  protocols
rip       route
Switch# show ipv6 ospf interface
eth-0-9 is up, line protocol is up
 Interface ID 9
 IPv6 Prefixes
  fe80::20e6:7eff:fee2:d400/10 (Link-Local Address)
  2004:12:9::1/96
 OSPFv3 Process (100), Area 0.0.0.0, Instance ID 0
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DROther, Priority 1
  Designated Router (ID) 3.3.3.3
   Interface Address fe80::ba5d:79ff:fe55:ed00
  Backup Designated Router (ID) 2.2.2.2
   Interface Address fe80::fcc8:7bff:fe3e:ec00
  Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:03
  Neighbor Count is 2, Adjacent neighbor count is 2
```

Display the result on Switch2:

```
Switch# show ipv6 ospf neighbor
OSPFv3 Process (200)
Neighbor ID    Pri   State           Dead Time   Interface  Instance ID
1.1.1.1          1   Full/DROther      00:00:31    eth-0-17   0
3.3.3.3         10   Full/DR           00:00:37    eth-0-17   0

Switch# show ipv6 ospf interface
eth-0-17 is up, line protocol is up
 Interface ID 17
 IPv6 Prefixes
  fe80::fcc8:7bff:fe3e:ec00/10 (Link-Local Address)
  2004:12:9::2/96
 OSPFv3 Process (200), Area 0.0.0.0, Instance ID 0
  Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 3.3.3.3
   Interface Address fe80::ba5d:79ff:fe55:ed00
  Backup Designated Router (ID) 2.2.2.2
   Interface Address fe80::fcc8:7bff:fe3e:ec00
  Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:07
  Neighbor Count is 2, Adjacent neighbor count is 2
```

Display the result on Switch3:

```
Switch# show ipv6 ospf neighbor
OSPFv3 Process (300)
Neighbor ID    Pri   State           Dead Time   Interface  Instance ID
1.1.1.1          1   Full/DROther      00:00:40    eth-0-13   0
2.2.2.2          1   Full/Backup       00:00:29    eth-0-13   0

Switch# show ipv6 ospf interface
eth-0-13 is up, line protocol is up
 Interface ID 13
 IPv6 Prefixes
  fe80::ba5d:79ff:fe55:ed00/10 (Link-Local Address)
  2004:12:9::3/96
 OSPFv3 Process (300), Area 0.0.0.0, Instance ID 0
  Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 10
  Designated Router (ID) 3.3.3.3
   Interface Address fe80::ba5d:79ff:fe55:ed00
  Backup Designated Router (ID) 2.2.2.2
   Interface Address fe80::fcc8:7bff:fe3e:ec00
  Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:06
  Neighbor Count is 2, Adjacent neighbor count is 2
```

Configuring OSPFv3 Area Parameters

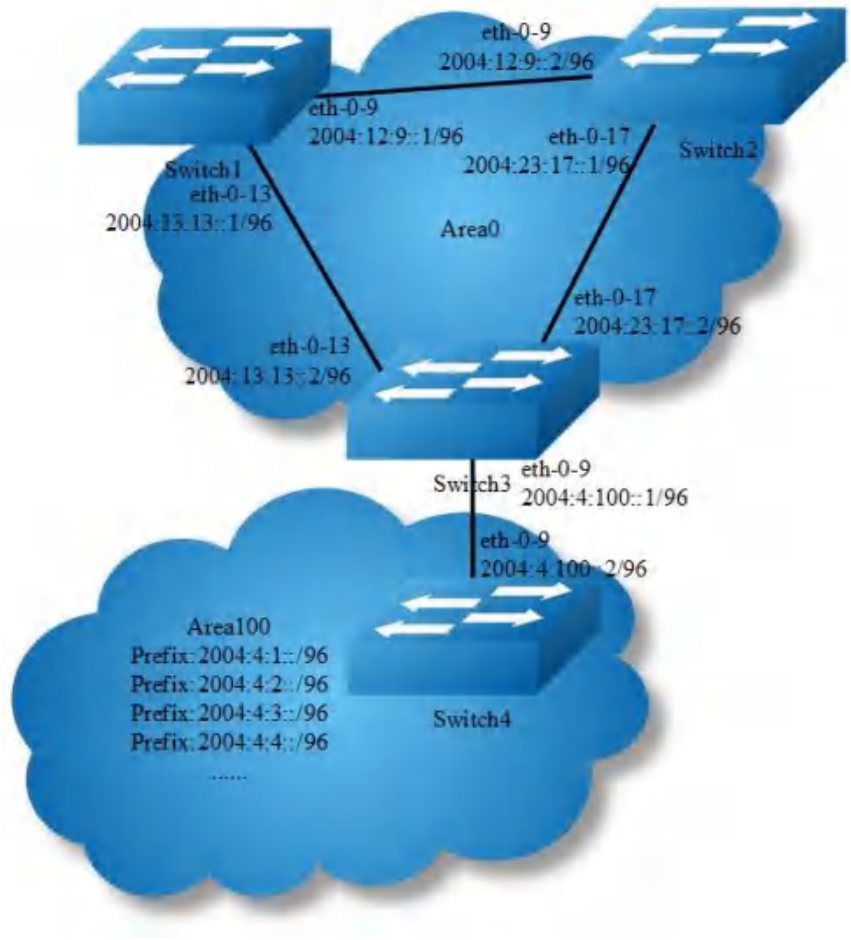


Figure 14-4 OSPFv3 area

You can optionally configure several OSPFv3 area parameters. These parameters include authentication for password-based protection against unauthorized access to an area and stub areas. Stub areas are areas into which information on external routes is not sent. Instead, the area border router (ABR) generates a default external route into the stub area for destinations outside the autonomous system (AS).

Route summarization is the consolidation of advertised addresses into a single summary route to be advertised by other areas. If network numbers are contiguous, you can use the area range router configuration command to configure the ABR to advertise a summary route that covers all networks in the range.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create OSPFv3 instance

Configuring Switch1:

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ipv6 ospf 200
Switch(config-router)# router-id 2.2.2.2
Switch(config-router)# exit
```

Configuring Switch3:

```
Switch(config)# router ipv6 ospf 300
Switch(config-router)# router-id 3.3.3.3
Switch(config-router)# exit

Switch(config)# router ipv6 ospf 300
Switch(config-router)# area 100 range 2004:4::/32
Switch(config-router)# area 100 stub no-summary
Switch(config-router)# exit
```

Configuring Switch4:

```
Switch(config)# router ipv6 ospf 400
Switch(config-router)# router-id 4.4.4.4
Switch(config-router)# area 100 stub no-summary
Switch(config-router)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::1/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-13
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)# ipv6 address 2004:13:13::2/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::2/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)# ipv6 address 2004:23:17::1/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
```

```
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:100::1/96
Switch(config-if)# ipv6 router ospf 300 area 100 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:13:13::2/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:23:17::2/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch4:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:1::1/96
Switch(config-if)# ipv6 router ospf 400 area 100 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:2::1/96
Switch(config-if)# ipv6 router ospf 400 area 100 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-3
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:3::1/96
Switch(config-if)# ipv6 router ospf 400 area 100 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-4
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:4::1/96
Switch(config-if)# ipv6 router ospf 400 area 100 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:100::2/96
Switch(config-if)# ipv6 router ospf 400 area 100 instance 0
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O IA   2004:4::/32 [110/3]
    via fe80::c629:f2ff:fe02:3600, eth-0-13, 00:01:00
C     2004:12:9::/96
    via ::, eth-0-9, 00:15:56
C     2004:12:9::1/128
    via ::1, eth-0-9, 00:15:56
C     2004:13:13::/96
    via ::, eth-0-13, 00:15:55
C     2004:13:13::2/128
    via ::1, eth-0-13, 00:15:55
O     2004:23:17::/96 [110/2]
    via fe80::bc22:aeff:fe64:aa00, eth-0-9, 00:08:10
    via fe80::c629:f2ff:fe02:3600, eth-0-13, 00:08:10
C     fe80::/10
    via ::, Null0, 00:15:57
```

Display the result on Switch2:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O IA   2004:4::/32 [110/3]
    via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:00:57
C     2004:12:9::/96
    via ::, eth-0-9, 00:12:24
C     2004:12:9::2/128
    via ::1, eth-0-9, 00:12:24
O     2004:13:13::/96 [110/2]
    via fe80::b242:55ff:fe05:ff00, eth-0-9, 00:07:52
    via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:07:52
C     2004:23:17::/96
    via ::, eth-0-17, 00:12:24
C     2004:23:17::1/128
    via ::1, eth-0-17, 00:12:24
C     fe80::/10
    via ::, Null0, 00:12:26
```

Display the result on Switch3:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O     2004:4::/32 [110/0]
    via ::, Null0, 00:08:31
O     2004:4:1::/96 [110/2]
    via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:01:08
O     2004:4:2::/96 [110/2]
    via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:01:08
O     2004:4:3::/96 [110/2]
    via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:01:08
O     2004:4:4::/96 [110/2]
    via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:01:08
C     2004:4:100::/96
    via ::, eth-0-9, 00:08:32
C     2004:4:100::1/128
    via ::1, eth-0-9, 00:08:32
O     2004:12:9::/96 [110/2]
    via fe80::b242:55ff:fe05:ff00, eth-0-13, 00:08:03
    via fe80::bc22:aeff:fe64:aa00, eth-0-17, 00:08:03
O     2004:13:13::/96 [110/1]
    via fe80::b242:55ff:fe05:ff00, eth-0-13, 00:08:18
C     2004:23:17::/96
    via ::, eth-0-17, 00:08:32
C     2004:23:17::2/128
    via ::1, eth-0-17, 00:08:32
C     fe80::/10
    via ::, Null0, 00:08:34
```

Display the result on Switch4:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O IA    ::/0 [110/2]
    via fe80::c629:f2ff:fe02:3600, eth-0-9, 00:00:53
C     2004:4:1::/96
    via ::, eth-0-1, 00:03:09
C     2004:4:1::1/128
    via ::1, eth-0-1, 00:03:09
C     2004:4:2::/96
    via ::, eth-0-2, 00:03:08
C     2004:4:2::1/128
```

```
    via ::1, eth-0-2, 00:03:08
C     2004:4:3::/96
    via ::, eth-0-3, 00:03:08
C     2004:4:3::1/128
    via ::1, eth-0-3, 00:03:08
C     2004:4:4::/96
    via ::, eth-0-4, 00:03:09
C     2004:4:4::1/128
    via ::1, eth-0-4, 00:03:09
C     2004:4:100::/96
    via ::, eth-0-9, 00:03:09
C     2004:4:100::2/128
    via ::1, eth-0-9, 00:03:09
C     fe80::/10
  via ::, Null0, 00:03:10
```

Redistributing Routes into OSPFv3

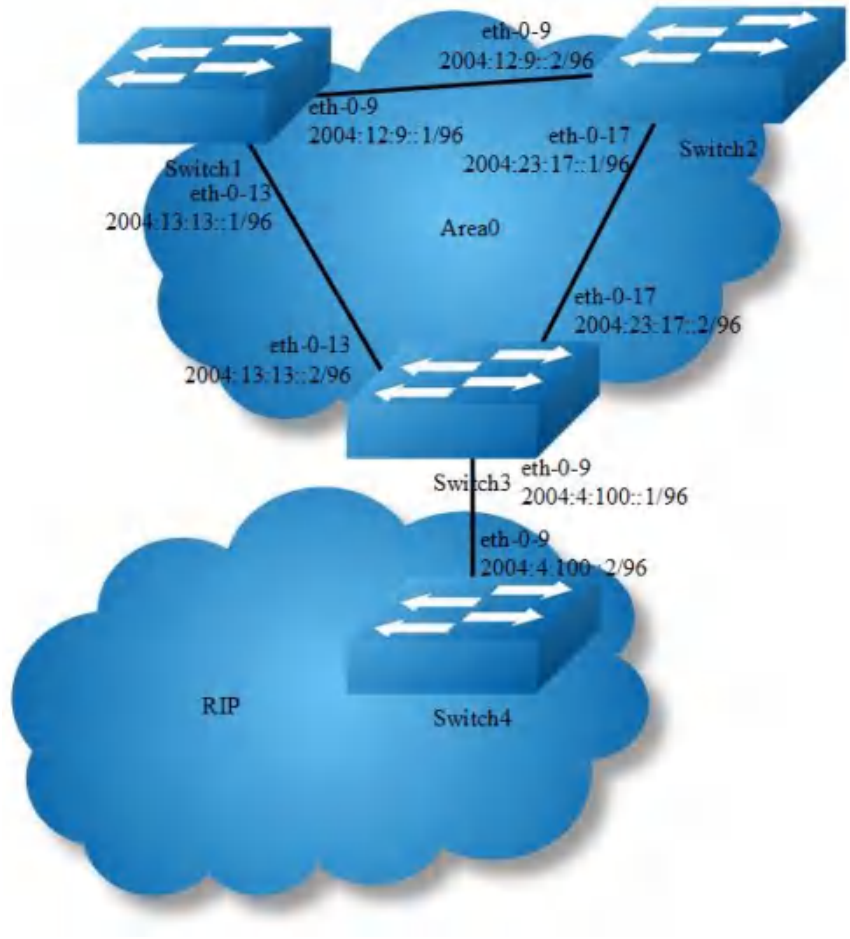


Figure 14-5 OSPFv3 Redistribute

In this example the configuration causes RIPng routes to be imported into the OSPFv3 routing table and advertised as Type 5 External LSAs into Area 0.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create OSPFv3 instance

Configuring Switch1:

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ipv6 ospf 200
Switch(config-router)# router-id 2.2.2.2
Switch(config-router)# exit
```

Configuring Switch3:

```
Switch(config)# router ipv6 ospf 300
Switch(config-router)# router-id 3.3.3.3
Switch(config-router)# redistribute ripng
Switch(config-router)# exit
```

step 4 Create RIPng instance

Configuring Switch3:

```
Switch(config)# router ipv6 rip
Switch(config-router)# exit
```

Configuring Switch4:

```
Switch(config)# router ipv6 rip
Switch(config-router)# exit
```

step 5 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::1/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:13:13::2/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
```

```
Switch(config-if)# ipv6 address 2004:12:9::2/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)# ipv6 address 2004:23:17::1/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:100::1/96
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit

Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:13:13::2/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:23:17::2/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch4:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:1::1/96
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:4:100::2/96
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O E2   2004:4:1::/96 [110/20]
    via fe80::c629:f2ff:fe02:3600, eth-0-13, 00:00:03
C     2004:12:9::/96
    via ::, eth-0-9, 00:34:20
C     2004:12:9::1/128
    via ::1, eth-0-9, 00:34:20
C     2004:13:13::/96
    via ::, eth-0-13, 00:34:19
C     2004:13:13::2/128
    via ::1, eth-0-13, 00:34:19
O     2004:23:17::/96 [110/2]
    via fe80::bc22:aeff:fe64:aa00, eth-0-9, 00:26:34
    via fe80::c629:f2ff:fe02:3600, eth-0-13, 00:26:34
C     fe80::/10
    via ::, Null0, 00:34:21

Switch# show ipv6 ospf database external
        OSPFv3 Router with ID (1.1.1.1) (Process 100)
          AS-external-LSA
 LS age: 140
 LS Type: AS-External-LSA
 Link State ID: 0.0.0.1
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x80000001
 Checksum: 0x66F7
 Length: 44
  Metric Type: 2 (Larger than any link state path)
  Metric: 20
  Prefix: 2004:4:1::/96
  Prefix Options: 0 (-|-|-|-)
  External Route Tag: 0
```

Display the result on Switch2:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O E2   2004:4:1::/96 [110/20]
    via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:02:43
C     2004:12:9::/96
    via ::, eth-0-9, 00:33:31
```

```
C     2004:12:9::2/128
      via ::1, eth-0-9, 00:33:31
O     2004:13:13::/96 [110/2]
      via fe80::b242:55ff:fe05:ff00, eth-0-9, 00:28:59
      via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:28:59
C     2004:23:17::/96
      via ::, eth-0-17, 00:33:31
C     2004:23:17::1/128
      via ::1, eth-0-17, 00:33:31
C     fe80::/10
      via ::, Null0, 00:33:33

Switch# show ipv6 ospf database external
show ipv6 ospf database external
          OSPFv3 Router with ID (2.2.2.2) (Process 200)
            AS-external-LSA
 LS age: 195
 LS Type: AS-External-LSA
 Link State ID: 0.0.0.1
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x80000001
 Checksum: 0x66F7
 Length: 44
   Metric Type: 2 (Larger than any link state path)
   Metric: 20
   Prefix: 2004:4:1::/96
   Prefix Options: 0 (-|-|-|-)
   External Route Tag: 0
```

Display the result on Switch3:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       Dr - DHCPV6 Relay
       [*] - [AD/Metric]
Timers: Uptime
R     2004:4:1::/96 [120/2]
      via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:03:43
C     2004:4:100::/96
      via ::, eth-0-9, 00:07:01
C     2004:4:100::1/128
      via ::1, eth-0-9, 00:07:01
O     2004:12:9::/96 [110/2]
      via fe80::b242:55ff:fe05:ff00, eth-0-13, 00:29:57
      via fe80::bc22:aeff:fe64:aa00, eth-0-17, 00:29:57
O     2004:13:13::/96 [110/1]
      via fe80::b242:55ff:fe05:ff00, eth-0-13, 00:30:12
C     2004:23:17::/96
      via ::, eth-0-17, 00:30:26
C     2004:23:17::2/128
      via ::1, eth-0-17, 00:30:26
C     fe80::/10
```

```
    via ::, Null0, 00:30:28

Switch# show ipv6 ospf database external
show ipv6 ospf database external
        OSPFv3 Router with ID (3.3.3.3) (Process 300)
          AS-external-LSA
 LS age: 250
 LS Type: AS-External-LSA
 Link State ID: 0.0.0.1
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x80000001
 Checksum: 0x66F7
 Length: 44
  Metric Type: 2 (Larger than any link state path)
  Metric: 20
  Prefix: 2004:4:1::/96
  Prefix Options: 0 (-|-|-|-)
  External Route Tag: 0
```

Display the result on Switch4:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
C     2004:4:1::/96
    via ::, eth-0-1, 00:04:48
C     2004:4:1::1/128
    via ::1, eth-0-1, 00:04:48
C     2004:4:100::/96
    via ::, eth-0-9, 00:06:59
C     2004:4:100::2/128
    via ::1, eth-0-9, 00:06:59
C     fe80::/10
    via ::, Null0, 00:07:00
```

Configure OSPFv3 Cost

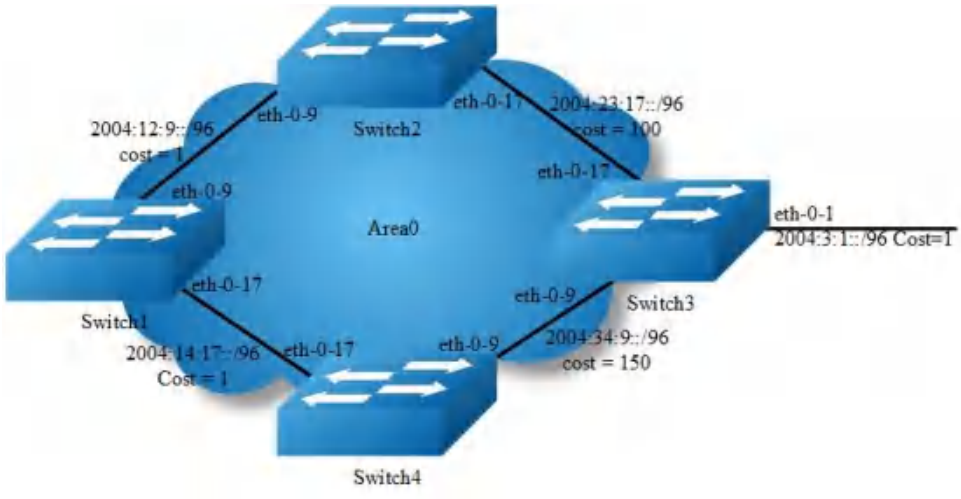


Figure 14-6 OSPFv3 Cost

You can make a route the preferred route by changing its cost. In this example, cost has been configured to make Switch2 the next hop for Switch1.

The default cost on each interface is 1(1000M speed). Interface eth2 on Switch2 has a cost of 100 and interface eth2 on Switch3 has a cost of 150. The total cost to reach(Switch4 network 10.10.14.0) through Switch2 and Switch3:

Switch2: 1+1+100 = 102 Switch3: 1+1+150 = 152

Therefore, Switch1 chooses Switch2 as its next hop for destination Switch4

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Create OSPFv3 instance

Configuring Switch1:

```
Switch(config)# router ipv6 ospf 100
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ipv6 ospf 200
Switch(config-router)# router-id 2.2.2.2
Switch(config-router)# exit
```

Configuring Switch3:

```
Switch(config)# router ipv6 ospf 300
Switch(config-router)# router-id 3.3.3.3
Switch(config-router)# exit
```

Configuring Switch4:

```
Switch(config)# router ipv6 ospf 400
Switch(config-router)# router-id 4.4.4.4
Switch(config-router)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::1/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)# ipv6 address 2004:14:17::1/96
Switch(config-if)# ipv6 router ospf 100 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:12:9::2/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)# ipv6 address 2004:23:17::1/96
Switch(config-if)# ipv6 router ospf 200 area 0 instance 0
Switch(config-if)# ipv6 ospf cost 100
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:3:1::1/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:34:9::1/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:23:17::2/96
Switch(config-if)# ipv6 router ospf 300 area 0 instance 0
Switch(config-if)# exit
```

Interface configuration for Switch4:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:34:9::2/96
Switch(config-if)# ipv6 router ospf 400 area 0 instance 0
Switch(config-if)# ipv6 ospf cost 150
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2004:14:17::2/96
Switch(config-if)# ipv6 router ospf 400 area 0 instance 0
Switch(config-if)# end
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show ipv6 ospf route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O     2004:3:1::/96 [110/102]
    via fe80::bc22:aeff:fe64:aa00, eth-0-9, 00:08:06
C     2004:12:9::/96
    via ::, eth-0-9, 01:15:43
C     2004:12:9::1/128
    via ::1, eth-0-9, 01:15:43
C     2004:14:17::/96
    via ::, eth-0-17, 00:18:38
C     2004:14:17::1/128
    via ::1, eth-0-17, 00:18:38
O     2004:23:17::/96 [110/101]
    via fe80::bc22:aeff:fe64:aa00, eth-0-9, 00:08:06
O     2004:34:9::/96 [110/102]
    via fe80::bc22:aeff:fe64:aa00, eth-0-9, 00:03:56
C     fe80::/10
    via ::, Null0, 01:15:44
```

Display the result on Switch2:

```
Switch# show ipv6 ospf route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O    2004:3:1::/96 [110/101]
    via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:08:33
C    2004:12:9::/96
    via ::, eth-0-9, 01:12:40
C    2004:12:9::2/128
    via ::1, eth-0-9, 01:12:40
O    2004:14:17::/96 [110/2]
    via fe80::b242:55ff:fe05:ff00, eth-0-9, 00:18:43
C    2004:23:17::/96
    via ::, eth-0-17, 01:12:40
C    2004:23:17::1/128
    via ::1, eth-0-17, 01:12:40
O    2004:34:9::/96 [110/101]
    via fe80::c629:f2ff:fe02:3600, eth-0-17, 00:04:23
C    fe80::/10
    via ::, Null0, 01:12:42
```

Display the result on Switch3:

```
Switch# show ipv6 ospf route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
C    2004:3:1::/96
    via ::, eth-0-1, 00:13:54
C    2004:3:1::1/128
    via ::1, eth-0-1, 00:13:54
O    2004:12:9::/96 [110/2]
    via fe80::bc22:aeff:fe64:aa00, eth-0-17, 00:19:47
O    2004:14:17::/96 [110/2]
    via fe80::ee66:91ff:fe45:db00, eth-0-9, 00:02:27
C    2004:23:17::/96
    via ::, eth-0-17, 01:09:02
C    2004:23:17::2/128
    via ::1, eth-0-17, 01:09:02
C    2004:34:9::/96
    via ::, eth-0-9, 00:04:52
C    2004:34:9::1/128
    via ::1, eth-0-9, 00:04:52
C    fe80::/10
    via ::, Null0, 01:09:04
```

Display the result on Switch4:

```
Switch# show ipv6 route
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
    O - OSPF, IA - OSPF inter area
```

```
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    Dr - DHCPV6 Relay
    [*] - [AD/Metric]
Timers: Uptime
O     2004:3:1::/96 [110/103]
    via fe80::b242:55ff:fe05:ff00, eth-0-17, 00:02:35
O     2004:12:9::/96 [110/2]
    via fe80::b242:55ff:fe05:ff00, eth-0-17, 00:02:35
C     2004:14:17::/96
    via ::, eth-0-17, 00:04:09
C     2004:14:17::2/128
    via ::1, eth-0-17, 00:04:09
O     2004:23:17::/96 [110/102]
    via fe80::b242:55ff:fe05:ff00, eth-0-17, 00:02:35
C     2004:34:9::/96
    via ::, eth-0-9, 00:06:06
C     2004:34:9::2/128
    via ::1, eth-0-9, 00:06:06
C     fe80::/10
    via ::, Null0, 00:44:59
```

Monitoring OSPFv3

You can display specific statistics such as the contents of IPv6 routing tables, caches, and databases.

Display general information about OSPFv3 routing processes

```
Switch# show ipv6 ospf
 Routing Process "OSPFv3 (300)" with ID 3.3.3.3
 Process uptime is 3 hours 23 minutes
 SPF schedule delay min 0.500 secs, SPF schedule delay max 50.0 secs
 Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum Sum 0x0000
 Number of AS-Scoped Unknown LSA 0
 Number of LSA originated 6
 Number of LSA received 43
 Number of areas in this router is 1
   Area BACKBONE(0)
      Number of interfaces in this area is 1(1)
      SPF algorithm executed 14 times
      Number of LSA 5.  Checksum Sum 0x30DCD
      Number of Unknown LSA 0
```

Display lists of information related to the OSPFv3 database

```
Switch# show ipv6 ospf database database-summary

        OSPFv3 Router with ID (3.3.3.3) (Process ID 300)
Area (0.0.0.0) database summary
 LSA Type          Count       MaxAge
 Router            3           0
 Network           1           0
 Inter-Prefix      0           0
 Inter-Router      0           0
```

```
 Intra-Prefix        1          0
 Subtotal            5          0

Process 300 database summary
 LSA Type          Count       MaxAge
 Router            3           0
 Network           1           0
 Inter-Prefix      0           0
 Inter-Router      0           0
 Type-5 Ext        0           0
 Link              3           0
 Intra-Prefix      1           0
 Total             8           0

Switch# show ipv6 ospf database router

        OSPFv3 Router with ID (3.3.3.3) (Process 300)

          Router-LSA (Area 0.0.0.0)

 LS age: 600
 LS Type: Router-LSA
 Link State ID: 0.0.0.0
 Advertising Router: 1.1.1.1
 LS Seq Number: 0x80000008
 Checksum: 0x9A57
 Length: 40
 Flags: 0x00 (-|-|-|-|-)
 Options: 0x000013 (-|R|-|-|E|V6)

  Link connected to: a Transit Network
    Metric: 1
    Interface ID: 9
    Neighbor Interface ID: 13
    Neighbor Router ID: 3.3.3.3



 LS age: 597
 LS Type: Router-LSA
 Link State ID: 0.0.0.0
 Advertising Router: 2.2.2.2
 LS Seq Number: 0x8000000D
 Checksum: 0xE2FD
 Length: 40
 Flags: 0x00 (-|-|-|-|-)
 Options: 0x000013 (-|R|-|-|E|V6)

  Link connected to: a Transit Network
    Metric: 1
    Interface ID: 17
    Neighbor Interface ID: 13
    Neighbor Router ID: 3.3.3.3
```

```
 LS age: 599
 LS Type: Router-LSA
 Link State ID: 0.0.0.0
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x8000000C
 Length: 40
 Flags: 0x00 (-|-|-|-|-)
 Options: 0x000013 (-|R|-|-|E|V6)

   Link connected to: a Transit Network
     Metric: 1
     Interface ID: 13
     Neighbor Interface ID: 13
     Neighbor Router ID: 3.3.3.3

Switch# show ipv6 ospf database network self-originate

          OSPFv3 Router with ID (3.3.3.3) (Process 300)

            Network-LSA (Area 0.0.0.0)

 LS age: 1261
 LS Type: Network-LSA
 Link State ID: 0.0.0.13
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x80000004
 Checksum: 0x727E
 Length: 36
 Options: 0x000013 (-|R|-|-|E|V6)
   Attached Router: 3.3.3.3
   Attached Router: 1.1.1.1
   Attached Router: 2.2.2.2

Switch# show ipv6 ospf database inter-router

          OSPFv3 Router with ID (3.3.3.3) (Process 300)

Switch# show ipv6 ospf database intra-prefix

          OSPFv3 Router with ID (3.3.3.3) (Process 300)

            Intra-Area-Prefix-LSA (Area 0.0.0.0)

 LS age: 1623
 LS Type: Intra-Area-Prefix-LSA
 Link State ID: 0.0.0.2
 Advertising Router: 3.3.3.3
 LS Seq Number: 0x80000004
 Checksum: 0x8FA8
 Length: 48
 Number of Prefixes: 1
 Referenced LS Type: 0x2002
 Referenced Link State ID: 0.0.0.13
 Referenced Advertising Router: 3.3.3.3
```

```
  Prefix: 2004:12:9::/96
  Prefix Options: 0 (-|-|-|-)
  Metric: 0



Switch# show ipv6 ospf database inter-prefix

        OSPFv3 Router with ID (3.3.3.3) (Process 300)

Switch# show ipv6 ospf database link

        OSPFv3 Router with ID (3.3.3.3) (Process 300)

          Link-LSA (Interface eth-0-13)

 LS age: 641
 LS Type: Link-LSA
 Link State ID: 0.0.0.9
 Advertising Router: 1.1.1.1
 LS Seq Number: 0x80000005
 Checksum: 0x9C1C
 Length: 60
 Priority: 1
 Options: 0x000013 (-|R|-|-|E|V6)
 Link-Local Address: fe80::20e6:7eff:fee2:d400
 Number of Prefixes: 1

   Prefix: 2004:12:9::/96
   Prefix Options: 0 (-|-|-|-)



 LS age: 698
 LS Type: Link-LSA
 Link State ID: 0.0.0.17
 Advertising Router: 2.2.2.2
 LS Seq Number: 0x80000008
 Checksum: 0x2159
 Length: 60
 Priority: 1
 Options: 0x000013 (-|R|-|-|E|V6)
 Link-Local Address: fe80::fcc8:7bff:fe3e:ec00
 Number of Prefixes: 1

   Prefix: 2004:12:9::/96
   Prefix Options: 0 (-|-|-|-)



 LS age: 1535
 LS Type: Link-LSA
 Link State ID: 0.0.0.13
```

```
Advertising Router: 3.3.3.3
LS Seq Number: 0x80000008
Checksum: 0x6E9A
Length: 60
Priority: 10
Options: 0x000013 (-|R|-|-|E|V6)
Link-Local Address: fe80::ba5d:79ff:fe55:ed00
Number of Prefixes: 1

  Prefix: 2004:12:9::/96
  Prefix Options: 0 (-|-|-|-)

Switch# show ipv6 ospf database external

        OSPFv3 Router with ID (3.3.3.3) (Process 300)
```

Display OSPFv3-related interface information

```
Switch# show ipv6 ospf interface
eth-0-13 is up, line protocol is up
 Interface ID 13
 IPv6 Prefixes
  fe80::ba5d:79ff:fe55:ed00/10 (Link-Local Address)
  2004:12:9::3/96
 OSPFv3 Process (300), Area 0.0.0.0, Instance ID 0
  Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 10
  Designated Router (ID) 3.3.3.3
    Interface Address fe80::ba5d:79ff:fe55:ed00
  Backup Designated Router (ID) 2.2.2.2
    Interface Address fe80::fcc8:7bff:fe3e:ec00
  Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
  Neighbor Count is 2, Adjacent neighbor count is 2
```

Display OSPFv3 interface neighbor information

```
Switch# show ipv6 ospf neighbor
OSPFv3 Process (300)
Neighbor ID    Pri   State           Dead Time   Interface  Instance ID
1.1.1.1          1   Full/DROther      00:00:39    eth-0-13   0
2.2.2.2          1   Full/Backup       00:00:33    eth-0-13   0
```

14.2.3 Application cases

N/A

14.3 Configuring RIPng

14.3.1 Overview

Function Introduction

Routing Information Protocol Next Generation (RIPng) is an IPv6 route exchange protocol that uses a distance vector (a number representing distance) to measure the cost of a given route. The cost is a distance vector because the cost is often equivalent to the number of router hops between the source and the destination networks. RIPng can receive multiple paths to a destination. The system evaluates the paths, selects the best path, and saves the path in the IPv6 route table as the route to the destination.

Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If RIPng receives a RIPng update from another router that contains a path with fewer hops than the path stored in the route table, the system replaces the older route with the newer one. The system then includes the new path in the updates it sends to other RIPng routers. RIPng routers also can modify a route's cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes. A RIPng route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

This chapter contains basic RIPng configuration examples. To see details on the commands used in these examples, or to see the outputs of the Validation commands, refer to the RIPng Command Reference. To avoid repetition, some Common commands, like configure terminal, have not been listed under the Commands Used section.

There are some differences between RIPng and RIP:

- UDP port number: RIPng uses UDP port number 521 to send or receive package.
- Multicast address: RIPng uses FF02::9 to multicast package to other routers of link local.
- Nexthop address: RIPng uses 128 bit ipv6 address.
- Source address: RIPng uses IPv6 link-local address FE80::/10 to be the source address when updating package to neighbor.

Principle Description

The RIPng module is based on the following RFC: RFC 2080 – RIPng for IPv6

14.3.2 Configuration

Enabling RIPng

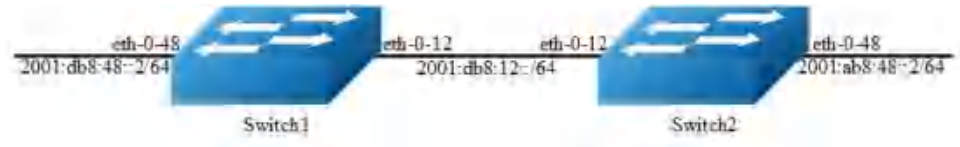


Figure 14-7 RIPng

This example shows how to enable RIPng protocols on two switches:

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 globally

```
Switch(config)# ipv6 enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-12
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2001:db8:12::1/64
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit

Switch(config)# interface eth-0-48
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2001:db8:48::2/64
```

```
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-12
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address  2001:db8:12::2/64
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit

Switch(config)# interface eth-0-48
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ipv6 address 2001:ab8:49::2/64
Switch(config-if)# ipv6 router rip
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Display the result on Switch1:

```
Switch# show ipv6 rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, Ra - RIP aggregated,
     Rcx - RIP connect suppressed, Rsx - RIP static suppressed,
     K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
  Network                Next Hop              If      Met Tag Time
R  2001:ab8:49::/64       fe80::1271:d1ff:fec8:3300 eth-0-12 5   0   00:02:34
Rc 2001:db8:12::/64        ::                  eth-0-12 1   0
Rc 2001:db8:48::/64        ::                  eth-0-48 1   0

Switch# show ipv6 rip interface
eth-0-12 is up, line protocol is up
 Routing Protocol: RIPng
   Passive interface: Disabled
   Split horizon: Enabled with Poisoned Reversed
   IPv6 interface address:
     2001:db8:12::1/64
     fe80::7e14:63ff:fe76:8900/10
eth-0-48 is up, line protocol is up
 Routing Protocol: RIPng
   Passive interface: Disabled
   Split horizon: Enabled with Poisoned Reversed
   IPv6 interface address:
     2001:db8:48::2/64
     fe80::7e14:63ff:fe76:8900/10

Switch# show ipv6 protocols rip
Routing Protocol is "ripng"
 Sending updates every 30 seconds with +/-5 seconds, next due in 7 seconds
 Timeout after 180 seconds, garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
```

```
 Default redistribute metric is 1
 Redistributing:
 Interface
  eth-0-12
  eth-0-48
 Routing for Networks:
 Number of routes (including connected): 3
 Distance: (default is 120)

Switch# show ipv6 route rip
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     Dr - DHCPV6 Relay
     [*] - [AD/Metric]
Timers: Uptime
R     2001:ab8:49::/64 [120/5]
     via fe80::1271:d1ff:fec8:3300, eth-0-12, 00:26:05
```

Display the result on Switch2:

```
Switch# show ipv6 rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, Ra - RIP aggregated,
     Rcx - RIP connect suppressed, Rsx - RIP static suppressed,
     K - Kernel, C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
  Network              Next Hop             If      Met Tag Time
Rc 2001:ab8:49::/64        ::                 eth-0-48 1   0
Rc 2001:db8:12::/64        ::                 eth-0-12 1   0
R  2001:db8:48::/64        fe80::7e14:63ff:fe76:8900 eth-0-12 2   0   00:02:33

Switch# show ipv6 rip interface
eth-0-12 is up, line protocol is up
 Routing Protocol: RIPng
  Passive interface: Disabled
  Split horizon: Enabled with Poisoned Reversed
  IPv6 interface address:
    2001:db8:12::2/64
    fe80::1271:d1ff:fec8:3300/10
eth-0-48 is up, line protocol is up
 Routing Protocol: RIPng
  Passive interface: Disabled
  Split horizon: Enabled with Poisoned Reversed
  IPv6 interface address:
    2001:ab8:49::2/64
    fe80::1271:d1ff:fec8:3300/10

Switch# show ipv6 protocols rip
Routing Protocol is "ripng"
 Sending updates every 30 seconds with +/-5 seconds, next due in 13 seconds
 Timeout after 180 seconds, garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Outgoing routes will have 3 added to metric if on list ripng_acl
 Default redistribute metric is 1
```

```
 Redistributing:
 Interface
  eth-0-12
  eth-0-48
 Routing for Networks:
 Number of routes (including connected): 3
 Distance: (default is 120)

Switch# show ipv6 route rip
IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, I - IS-IS, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     Dr - DHCPV6 Relay
     [*] - [AD/Metric]
Timers: Uptime
R     2001:db8:48::/64 [120/2]
     via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:23:31
```

Configuring Metric Parameters

A RIPng offset list allows you to add to the metric of specific inbound or outbound routes learned or advertised by RIPng. RIPng offset lists provide a simple method for adding to the cost of specific routes and therefore biasing the router's route selection away from those routes. An offset list consists of the following parameters:

- An ACL that specifies the routes to which to add the metric.
- In: applies to routes the router learns from RIPng neighbors.
- Out: applies to routes the router is advertising to its RIPng neighbors.
- The offset value that will be added to the routing metric of the routes that match the ACL.
- The interface that the offset list applies (optional).

If a route matches both a global offset list (without specified interface) and an interface-based offset list, the interface-based offset list takes precedence. The interface-based offset list's metric is added to the route in this case.

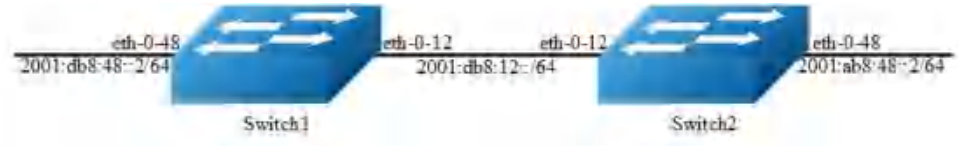


Figure 14-8 RIPng Metric

This example Switch 1 will advertise route 2001:db8:48::2/64 out of interface eth-0-12 with metric 3.

step 1 Check the current configuration

Current configuration of Switch1:

```
Switch# show running-config
!
ipv6 enable
!
Switch# show run
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::1/64
 ipv6 router rip
```

```
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:db8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Current configuration of Switch2:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::2/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:ab8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Check the RIPng states on Switch2:

```
Switch# show ipv6 route rip
R      2001:db8:48::/64 [120/2]
       via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:44:47
```

The following configurations are operated on Switch1:

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Create access list

```
Switch(config)#ipv6 access-list ripngoffset
Switch(config-ipv6-acl)# permit any 2001:db8:48::/64 any
Switch(config-ipv6-acl)# exit
```

step 4 Apply the access list

```
Switch(config)# router ipv6 rip
Switch(config-router)# offset-list ripngoffset out 3 eth-0-12
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch2:

```
Switch# show ipv6 route rip
R      2001:db8:48::/64 [120/5]
       via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:00:07
```

Configuring the Administrative Distance

By default, RIPng assigns the default RIPng administrative distance (120) to RIPng routes. When comparing routes based on administrative distance, the router selects the route with the lower distance. You can change the administrative distance for RIPng routes.

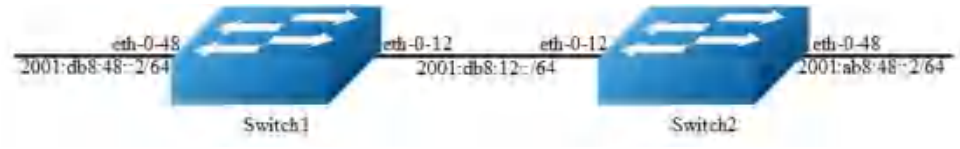


Figure 14-9 RIPng Distance

This example shows how to change the RIPng administrative distance.

step 1 Check the current configuration

Current configuration of Switch1:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::1/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:db8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Current configuration of Switch2:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::2/64
 ipv6 router rip
!
interface eth-0-48
```

```
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:ab8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Check the RIPng states on Switch2:

```
Switch# show ipv6 route rip
R     2001:db8:48::/64 [120/2]
      via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:44:47
```

The following configurations are operated on Switch2:

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Change the administrative distance

```
Switch(config)# router ipv6 rip
Switch(config-router)# distance 100
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Display the result on Switch2:

```
Switch# show ipv6 route rip
R     2001:db8:48::/64 [100/5]
      via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:00:09
```

Configuring Redistribution

You can configure the router to redistribute static routes, direct connected routes or routes learned through Open Shortest Path First (OSPF) into RIPng. When you redistribute a route from one of these other protocols into RIPng, the router can use RIPng to advertise the route to its RIPng neighbors.

Change the default redistribution metric (optional). The router assigns a RIPng metric of 1 to each redistributed route by default. You can change the default metric to a value up to 16.

Enable specified routes to redistribute with default or specified metric.

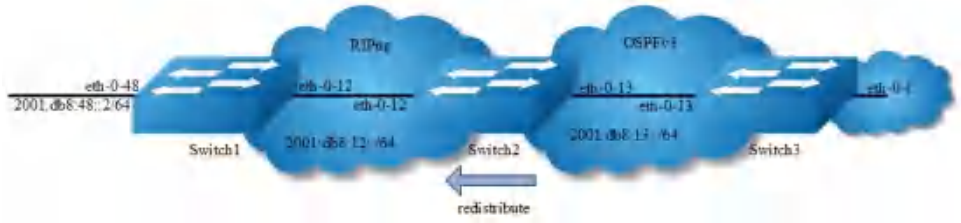


Figure 14-10 RIPng redistribute

This example shows how to redistribute other protocols into RIPng.

step 1 Check the current configuration

Current configuration of Switch1:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::1/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:db8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Current configuration of Switch2:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::2/64
 ipv6 router rip
!
interface eth-0-13
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:13::1/64
 ipv6 router ospf area 0
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:ab8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
router ipv6 ospf
 router-id 1.1.1.1
```

Current configuration of Switch3:

```
Switch# show running-config
!
```

```
ipv6 enable
!
interface eth-0-1
 no switchport
ipv6 address auto link-local
 ipv6 address 2001:db8:1::1/64
 ipv6 router ospf area 0
!
interface eth-0-13
 no switchport
 ipv6 address 2001:db8:13::2/64
 ipv6 router ospf area 0
!
router ipv6 ospf
 router-id 2.2.2.2
!
```

Check the RIPng states on Switch1:

```
Switch# show ipv6 route rip
R      2001:ab8:48::/64 [120/5]
       via fe80::1271:d1ff:fec8:3300, eth-0-12, 01:43:37
```

Check the RIPng states on Switch2:

```
Switch# show ipv6 route
O      2001:db8:1::/64 [110/2]
       via fe80::5c37:1dff:febe:2d00, eth-0-13, 00:31:17
R      2001:db8:48::/64 [100/5]
       via fe80::7e14:63ff:fe76:8900, eth-0-12, 00:49:57
```

The following configurations are operated on Switch2:

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Enable redistribute, and et the default metric and redistribute metric

```
Switch(config)# router ipv6 rip
Switch(config-router)# default-metric 2
Switch(config-router)# redistribute ospfv3 metric 5
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Display the result on Switch1:

```
Switch# show ipv6 route rip
R      2001:ab8:48::/64 [120/5]
       via fe80::1271:d1ff:fec8:3300, eth-0-12, 01:48:23
R      2001:db8:1::/64 [120/6]
       via fe80::1271:d1ff:fec8:3300, eth-0-12, 00:00:19
```

Configuring Split-horizon Parameters

Normally, routers that are connected to multicast-type IPv6 networks and that use distance-vector routing protocols employ the split horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routers, particularly when links are broken. However, with non-multicast networks (such as Frame Relay), situations can arise for which this behavior is less than ideal. For these situations, you might want to disable split horizon for RIPng.

You can avoid including routes in updates sent to the same gateway from which they were learned. Using the split horizon command omits routes learned from one neighbor, in updates sent to that neighbor. Using the poisoned parameter with this command includes such routes in updates, but sets their metrics to infinity. Thus, advertising these routes means that they are not reachable.

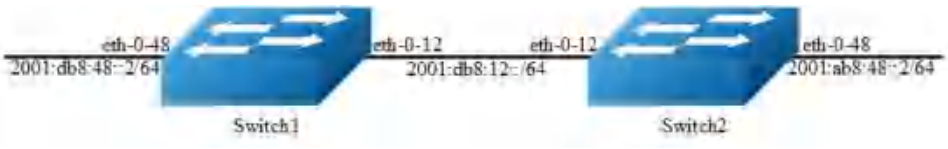


Figure 14-11 RIPng Split-horizon

step 1 Check the current configuration

Current configuration of Switch1:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::1/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:db8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Current configuration of Switch2:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::2/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:ab8:48::2/64
```

```
 ipv6 router rip
!
router ipv6 rip
!
```

Enable debug on switch2

```
Switch# debug ipv6 rip packet send detail
Switch# terminal monitor
```

The following configurations are operated on Switch2:

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Set the split-horizon on interface configure mode

Disable split-horizon:

```
Switch(config)#interface eth-0-12
Switch(config-if)# no ipv6 rip split-horizon
Switch(config-if)# exit
```

System debug information:

```
Oct 24 10:00:06 Switch RIPNG6-7: SEND[eth-0-12]: Send to [ff02::9]:521
Oct 24 10:00:06 Switch RIPNG6-7: SEND[eth-0-12]: RESPONSE version 1 packet size 64
Oct 24 10:00:06 Switch RIPNG6-7: 2001:ab8:49::/64 metric 4 tag 0
Oct 24 10:00:06 Switch RIPNG6-7: 2001:db8:12::/64 metric 1 tag 0
Oct 24 10:00:06 Switch RIPNG6-7: 2001:db8:48::/64 metric 5 tag 0
```

Enable split-horizon:

```
Switch(config)#interface eth-0-12
Switch(config-if)# ipv6 rip split-horizon
Switch(config-if)# exit
```

System debug information:

```
Oct 24 10:05:16 Switch RIPNG6-7: SEND[eth-0-12]: Send to [ff02::9]:521
Oct 24 10:05:16 Switch RIPNG6-7: SEND[eth-0-12]: RESPONSE version 1 packet size 44
Oct 24 10:05:16 Switch RIPNG6-7: 2001:ab8:49::/64 metric 4 tag 0
Oct 24 10:05:16 Switch RIPNG6-7: 2001:db8:12::/64 metric 1 tag 0
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ipv6 rip interface
eth-0-12 is up, line protocol is up
 Routing Protocol: RIPng
  Passive interface: Disabled
  Split horizon: Disabled
  IPv6 interface address:
   2001:ab8:48::2/64
   2001:db8:12::2/64
   fe80::7eff:80ff:fef4:ff00/10
```

Configuring Timers

RIPng use several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune RIPng performance to better suit your internet-work needs. You can make the following timer adjustments:

- The rate (time in seconds between updates) at which routing updates are sent.
- The interval of time (in seconds) after which a route is declared invalid.
- The amount of time (in seconds) that must pass before a route is removed from the routing table.

To configure the timers, use the following command:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the timers

Set the routing table update timer to 10 seconds. Set the routing information timeout timer to 180 seconds. Set the routing garbage collection timer to 120 seconds.

```
Switch(config)# router ipv6 rip
Switch(config-router)# timers basic 10 180 120
Switch(config-router)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

Use the commands as follows to validate the configuration:

```
Switch# show ipv6 protocols rip
Routing Protocol is "ripng"
 Sending updates every 10 seconds with +/-5 seconds, next due in 5 seconds
 Timeout after 180 seconds, garbage collect after 120 seconds
 Outgoing update filter list for all interface is not set
 Incoming update filter list for all interface is not set
 Outgoing routes will have 3 added to metric if on list ripng_acl
 Default redistribute metric is 2
 Redistributing:
 Interface
  eth-0-12
  eth-0-48
 Routing for Networks:
 Number of routes (including connected): 3
 Distance: (default is 100)
```

Configuring RIPng Route Distribute Filters

A RIP distribute list allows you to permit or deny learning or advertising of specific routes. A distribute list consists of the following parameters:

- An ACL or a prefix list that filter the routes.
- In: filter applies to learned routes.
- Out: filter applies to advertised routes
- The interface that the filer applies (optional).

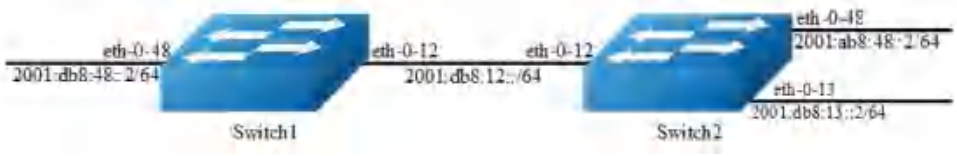


Figure 14-12 RIPng Route Distribute Filters

step 1 Check the current configuration

Current configuration of Switch1:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::1/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:db8:48::2/64
 ipv6 router rip
!
router ipv6 rip
!
```

Current configuration of Switch2:

```
Switch# show running-config
!
ipv6 enable
!
interface eth-0-12
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:12::2/64
 ipv6 router rip
!
interface eth-0-13
 no switchport
 ipv6 address auto link-local
 ipv6 address 2001:db8:13::1/64
 ipv6 router rip
!
interface eth-0-48
 no switchport
 ipv6 nd ra mtu suppress
 ipv6 address auto link-local
 ipv6 address 2001:ab8:48::2/64
 ipv6 router rip
!
```

```
router ipv6 rip
!
```

Check the RIPng states on Switch1:

```
Switch# show ipv6 route rip
R     2001:ab8:48::/64 [120/5]
      via fe80::1271:d1ff:fec8:3300, eth-0-12, 00:18:29
R     2001:db8:13::/64 [120/2]
      via fe80::1271:d1ff:fec8:3300, eth-0-12, 00:03:37
```

The following configurations are operated on Switch2:

step 2 Enter the configure mode

```
Switch# configure terminal
```

step 3 Create IPv6 Prefix list

```
Switch(config)# ipv6 prefix-list ripngfilter seq 5 deny 2001:db8:48::/64
Switch(config)# ipv6 prefix-list ripngfilter seq 10 permit any
```

step 4 Apply the IPv6 Prefix list

```
Switch(config)# router ipv6 rip
Switch(config-router)# distribute-list prefix ripngfilter out eth-0-12
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show ipv6 route rip
R     2001:db8:13::/64 [120/2]
      via fe80::1271:d1ff:fec8:3300, eth-0-12, 00:03:37
```

14.3.3 Application cases

N/A

14.4 Configuring Ipv6 Prefix-list

14.4.1 Overview

Function Introduction

Routing Policy is the technology for modifying route information to change traffic route. IPv6 Prefix list is a kind of route policies that used to control and modify routing information. A IPv6 prefix list is identified by list name and contains one or more ordered entries which are processed sequentially. Each entry provides a matched range for network prefix and has a unique sequence number in the list. In the matching process，switch will check entries orderly. If an entry matches conditions, this process would finish.

Principle Description

N/A

14.4.2 Configuration

Basic Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create IPv6 Prefix list

```
Switch(config)# ipv6 prefix-list test seq 1 deny 2001:db8::1/32 le 48
Switch(config)# ipv6 prefix-list test permit any
Switch(config)# ipv6 prefix-list test description this ipv6 prefix list is fot test
Switch(config)# ipv6 prefix-list test permit 2001:abc::1/32 le 48
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ipv6 prefix-list detail
Prefix-list list number: 1
Prefix-list entry number: 3
Prefix-list with the last deletion/insertion: test
ipv6 prefix-list test:
  Description: this ipv6 prefix list is fot test
  count: 3, range entries: 0, sequences: 1 - 10
    seq 1 deny 2001:db8::1/32 le 48 (hit count: 0, refcount: 0)
    seq 5 permit any (hit count: 0, refcount: 0)
    seq 10 permit 2001:abc::1/32 le 48 (hit count: 0, refcount: 0)
```

Used by RIPng

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create IPv6 Prefix list

```
Switch(config)# ipv6 prefix-list aa seq 11 deny 2001:db8::1/32 le 48
Switch(config)# ipv6 prefix-list aa permit any
```

Step 3 Apply the IPv6 Prefix list

```
Switch(config)# router ipv6 rip
Switch(config-router)# distribute-list prefix aa out
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show ipv6 prefix-list
ipv6 prefix-list aa: 2 entries
   seq 11 deny 1:db8::1/32 le 48
   seq 15 permit any

Switch# show running-config
Building configuration...
…
ipv6 prefix-list aa seq 11 deny 1:db8::1/32 le 48
```

```
ipv6 prefix-list aa seq 15 permit any
…
router ipv6 rip
distribute-list prefix aa out
```

Used by Route-map

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create IPv6 Prefix list

```
Switch(config)# ipv6 prefix-list ripng_pre_1 seq 11 permit fe80::a8f0:d8ff:fe7d:c501/128
Switch(config)# ipv6 prefix-list ripng_pre_1 permit any
```

step 3 Apply the IPv6 Prefix list to the route map

```
Switch(config)# route-map ripng_rmap permit
Switch(config-route-map)# match ipv6 address prefix-list ripng_pre_1
Switch(config-route-map)# set local-preference 200
Switch(config-route-map)# exit
```

step 4 Apply the route map to the RIPng instance

```
Switch(config)# router ipv6 rip
Switch(config-router)# redistribute static route-map ripng_rmap
Switch(config-router)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch # show route-map
route-map ripng_rmap, permit, sequence 10
 Match clauses:
   ipv6 next-hop prefix-list ripng_pre_1
 Set clauses:
   ipv6 next-hop local fe80::1

Switch # show running-config
Building configuration...
…
ipv6 prefix-list ripng_pre_1 seq 11 permit fe80::a8f0:d8ff:fe7d:c501/128
ipv6 prefix-list ripng_pre_1 seq 15 permit any
!
!
route-map ripng_rmap permit 10
 match ipv6 next-hop prefix-list ripng_pre_1
 set ipv6 next-hop local fe80::1
!
router ipv6 rip
 redistribute static route-map ripng_rmap
 !
ipv6 route 2001:dbc::/64 fe80::a8f0:d8ff:fe7d:c501 eth-0-9
!
```

```
Switch# show ipv6 rip database
S  2001:dbc::/64          fe80::1              eth-0-9 1   0
```

14.4.3 Application cases

N/A

14.5 ConfiguringIPv6 PBR

14.5.1 Overview

Brief Introduction

Different from forwarding ipv6 packets based on destination address, PBR is a mechanism that forwards routes based on user-defined policies. Packets that meet certain conditions will be forwared according to the specified behavior. Users can set different next-hop based on different packet characteristics to implement flexible routing and forwarding.

1. Process of Forwarding IPv6 Packets

After packets arrive, system will perform the forwarding operation in the following sequence:

- Match the route-map. If any route-map configured next-hop is matched, packets will be forwarded according to this matched next-hop.
- If no next-hop of route-map is matched, packets will be forwarded according to specific route.
- If there is no specfic route, but any route-map configured default next-hop is matched, packets will be forwared according to this matched default next-hop.
- If the preceding conditions are not met, packets will be forwared according to the default-route.

The prioritties of packet forwarding can be considered as follows:
PBR next-hop > Specific Route > PBR default next-hop > Default Route

2. ACL and route-map in IPv6 PBR

- Matching rules for IPv6 PBR:
- In IPv6 PBR, access control list (ACL) are used to match packets, and ACL clauses include two types: permit and deny.
- The matching features supported by IPv6 PBR include IPv6 source address, IPv6 destination address, protocol type, and port number.
- Currently, IPv6 PBR only supports permit type ACL clauses and ignores deny type clauses.
- Routing policies for IPv6 PBR:
- In IPv6 PBR, route-map are used to determine the combination of ACL used to match packets and the forwarding next hop after matching.
- By setting match rules and set rules to specify the forwarding behavior of PBR, IPv6 PBR match rule is “match ipv6 address,” set rules are “set ipv6 nexthop” or “set ipv6 default nexthop.”
- Route-map consist of permit and deny sequences, with IPv6 PBR only supporting permit type route-map sequences and ignoring deny type sequences.
- Relationship between packet matching results and types of ACL and route-map sequences:
- If there are multiple permit clauses in an ACL, the ACL matching result is a hit as long as it matches one of the permit clauses; if none of the permit clauses can match the packet, the matching result is a miss.
- If the ACL matching result is a hit, the corresponding route-map sequence is also a hit, and the packet will be forwarded according to the set rules of that sequence.

- Route-map sequences have priorities, with higher priority sequences matched first; if a match is found, the packet is directly forwarded. If no match is found, the sequences are progressively matched by priority until a sequence is hit, or there are no more sequences to match.

14.5.2 Configuration Examples

IPv6 PBR Configuration

1. Topology

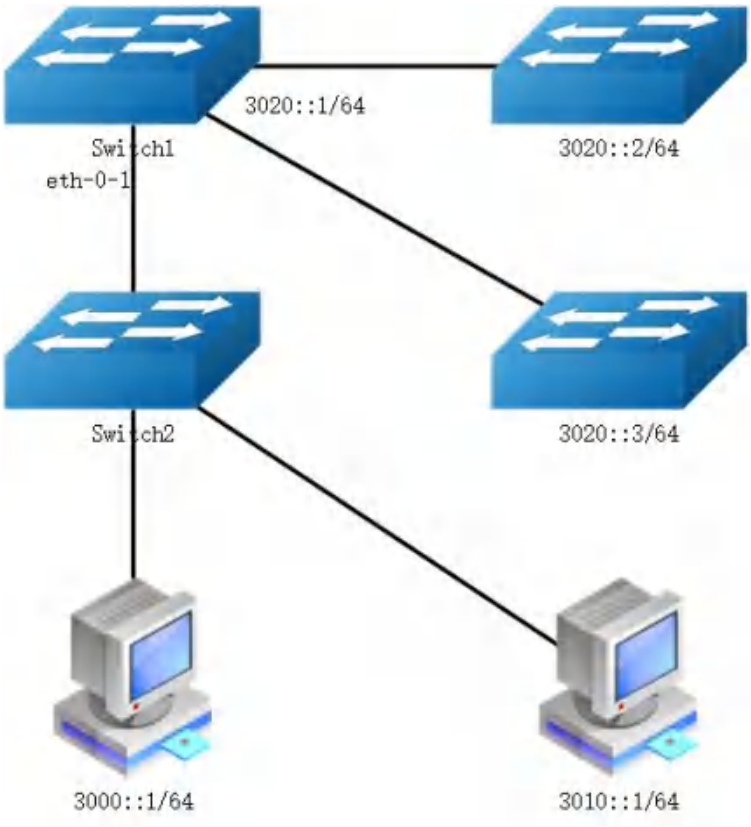


Figure 14-13 Typical networking diagram of IPv6 PBR

The figure above is a typical topology: After Enabling PBR on interface eth-0-1 of Switch1, packets from 3000::1 should be forwarded to 3020::2, and other packets should be forwarded according to the original routes.

Configure on Switch1:

2. Configuration Steps

3. Step1 Enter the configure mode

```
Switch# configure terminal
```

Step2 Create an IPv6 access list to match source ip address

```
Switch(config)# ipv6 access-list acl1
Switch(config-ip-acl)# 10 permit any 3000::1/128 any
Switch(config-ip-acl)# exit
```

Step3 Create a route map, to match the IPv6 access list and set the nexthop ip

```
Switch(config)# route-map rmap permit 10
Switch(config-route-map)# match ipv6 address acl1
```

```
Switch(config-route-map)# set ipv6 next-hop 3020::2
Switch(config-route-map)# exit
```

Step4 Enter the interface configure mode, set the attributes and ip address, and apply the route map

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 3020::1/64
Switch(config-if)# no shutdown
Switch(config-if)# ip policy route-map rmap
Switch(config-if)# exit
```

Step5 Create a static route with the nexthop ip 3020::3 (optional)

To forwarding the packets which not hit the PBR, we can use a static route. Dynamic protocols such as RIP/OSPF are can also meet this requirement.

```
Switch(config)# ipv6 route 0::0/0 3020::3
```

Step6 Exit the configure mode

```
Switch(config)# end
```

Validation

```
Switch# show ip policy route-map
Route-map              interface
rmap                   eth-0-1
```

15 IPv6 Multicast Configuration Guide

15.1 Configuring IPv6 Multicast-Routing

15.1.1 Overview

Function Introduction

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmit of multicast data.

Distribution of stock quotes, video transmissions such as news services and remote classrooms, and video conferencing are all examples of applications that use multicast routing.

- Mulitcast Listener Discovery (MLD) is used among hosts on a LAN and the routers (and multilayer switches) on that LAN to track the multicast groups of which hosts are members.
- Protocol-Independent Multicast (PIM) protocol is used among routers and multilayer switches to track which multicast packets to forward to each other and to their directly connected LANs. PIM has two modes: Sparse-mode and Dense-mode. Currently, we only support Sparse-mode

Principle Description

N/A

15.1.2 Configuration

Configuring IPv6 multicast route limit

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the limit of the IPv6 multicast route

```
Switch(config)# ipv6 multicast route-limit 1000
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ipv6 mroute route-limit
IPv6 Max Multicast Route Limit Number: 1000
IPv6 Multicast Route Limit Warning Threshold: 1000
IPv6 Multicast Hardware Route Limit: 255
IPv6 Current Multicast Route Entry Number: 0
```

15.1.3 Application cases

N/A

15.2 Configuring MLD

15.2.1 Overview

Function Introduction

To participate in IPv6 multicasting, multicast hosts, routers, and multilayer switches must have the MLD operating. This protocol defines the query and host roles:

- A query is a network device that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver that sends report messages (in response to query messages) to inform a querier of a host membership.
- A set of queries and hosts that receive IPv6 multicast data streams from the same source is called an IPv6 multicast group. Queries and hosts use MLD messages to join and leave IPv6 multicast groups. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message. Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group.
- A host can be a member of more than one multicast group at a time. How active a multicast group is and what members it has can vary from group to group and from time to time. A multicast group can be active for a long time, or it can be very short-lived. Membership in a group can constantly change. A group that has members can have no activity.

MLD packets are sent using these IPv6 multicast group addresses:

- MLD general queries are destined to the address ff02::1 (all systems on a subnet).
- MLD group-specific queries are destined to the group IPv6 address for which the switch is querying.
- MLD group membership reports are destined to the group IPv6 address for which the switch is reporting.
- MLD Version 1 (MLDv1) leave messages are destined to the address ff02::2 (all-multicast-routers on a subnet). In some old host IPv6 stacks, leave messages might be destined to the group IPv6 address rather than to the all-routers address.

Principle Description

The MLD module is based on the following RFC

- RFC 2710
- RFC 3810

15.2.2 Configuration

There is no explicit command to enable MLD, which is always combined with PIMv6-SM. When PIMv6-SM is enabled on an interface, MLD will be enabled automatically on this interface, vice versa. But notice, before MLD can work, IPv6 Multicast-routing must be enabled globally firstly. We support build MLD group record by learning MLD packets or configuring static MLD group by administer.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable ipv6 and ipv6 multicast-routing globally

```
Switch(config)# ipv6 enable
Switch(config)# ipv6 multicast-routing
```

step 3 Enter the interface configure mode, set the ipv6 address and enable pim sparse mode

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1::1/64
Switch(config-if)# ipv6 pim sparse-mode
```

step 4 Configuring MLD Interface Parameters

```
Switch(config-if)# ipv6 mld version 2
Switch(config-if)# ipv6 mld query-interval 120
Switch(config-if)# ipv6 mld query-max-response-time 12
Switch(config-if)# ipv6 mld robustness-variable 3
Switch(config-if)# ipv6 mld last-member-query-count 3
Switch(config-if)# ipv6 mld last-member-query-interval 2000
```

step 5 Limit Max MLD Group Number

Set the maxinum of ipv6 mld on the interface:

```
Switch(config-if)# ipv6 mld limit 1000
Switch(config-if)# exit
```

Set the maxinum of ipv6 mld globally:

```
Switch(config)# ipv6 mld limit 2000
```

step 6 Create static mld group

```
Switch(config)# interface eth-0-1
Switch(config-if)# ipv6 mld static-group ff0e::1234
Switch(config-if)# exit
```

step 7 Set IPv6 MLD proxy (optional)

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# ipv6 mld proxy-service
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# ipv6 mld mroute-proxy eth-0-1
Switch(config-if)# exit
```

step 8 Exit the configure mode

```
Switch(config)# end
```

step 9 Validation

Displaying MLD Interface:

```
Switch# show ipv6 mld interface
Interface eth-0-2 (Index 2)
 MLD Inactive, Version 1 (default)
 MLD mroute-proxy interface is eth-0-1
 MLD global limit is 2000
 MLD global limit states count is currently 0
 MLD interface limit is 4096
 MLD interface has 0 group-record states
 MLD activity: 0 joins, 0 leaves
 MLD query interval is 125 seconds
 MLD querier timeout is 255 seconds
 MLD max query response time is 10 seconds
 Last member query response interval is 1000 milliseconds
 Group Membership interval is 260 seconds
 Last memeber query count is 2
 Robustness Variable is 2
Interface eth-0-1 (Index 1)
 MLD Inactive, Configured for Version 2 proxy-service
 MLD host version 2
 MLD global limit is 2000
 MLD global limit states count is currently 0
 MLD interface limit is 1000
 MLD interface has 0 group-record states
 MLD activity: 0 joins, 0 leaves
```

```
MLD query interval is 120 seconds
MLD querier timeout is 366 seconds
MLD max query response time is 12 seconds
Last member query response interval is 2000 milliseconds
Group Membership interval is 372 seconds
Last memeber query count is 3
Robustness Variable is 3
```

Displaying MLD group:

```
Switch# show ipv6 mld groups
MLD Connected Group Membership
Group Address                Interface        Expires
ff0e::1234                   eth-0-1          stopped
```

15.3 Configuring PIMv6-SM

15.3.1 Overview

Function Introduction

The Protocol Independent Multicasting-Sparse Mode for IPv6 (PIMv6-SM) is a multicast routing protocol designed to operate efficiently across Wide Area Networks (WANs) with sparsely distributed groups. It helps network nodes that are geographically dispersed to conserve bandwidth, and reduces traffic by simultaneously delivering a single stream of information to multiple locations.

PIMv6-SM uses the IPv6 multicast model of receiver-initiated membership, supporting both shared and shortest-path trees, and uses soft-state mechanisms to adapt to changing network conditions. It relies on a topology-gathering protocol to populate a multicast routing table with routes.

Principle Description

The PIMv6-SM module is based on the following IETF standard: RFC 4601

Terminology：

- **Rendezvous Point (RP)**: A Rendezvous Point (RP) router is configured as the root of the non-source-specific distribution tree for a multicast group. Join messages from receivers for a group are sent towards the RP. Data from senders is sent to the RP so that receivers can discover who the senders are, and receive traffic destined for the group.
- **Multicast Routing Information Base (MRIB)**: The MRIB is a multicast topology table derived from the unicast routing table. In PIMv6-SM, the MRIB is used to decide where to send Join/Prune messages. It also provides routing metrics for destination addresses. These metrics are used when sending and processing Assert messages.
- **Reverse Path Forwarding**: Reverse Path Forwarding (RPF) is a concept of an optimized form of flooding, where the router accepts a packet from SourceA through Interface IF1 only if IF1 is the interface the router would use in order to reach SourceA. It determines whether the interface is correct by consulting its unicast routing tables. The packet that arrives through interface IF1 is forwarded because the routing table lists this interface as the shortest path to the network. The router's unicast routing table determines the shortest path for the multicast packets. Because a router accepts a packet from only one neighbor, it floods the packet only once, meaning that (assuming point-to-point links) each packet is transmitted over each link once in each direction.
- **Tree Information Base (TIB)**: The TIB is the collection of state at a PIM router storing the state of all multicast distribution trees at that router. It is created by receiving Join/Prune messages, Assert messages, and MLD information from local hosts.
- **Upstream**: Towards the root of the tree. The root of the tree might be either the Source or the RP.
- **Downstream**: Away from the root of the tree. The root of tree might be either the Source or the RP.
- **Source-Based Trees**: In the Source-Based Trees concept, the forwarding paths are based on the shortest unicast path to the source. If the unicast routing metric is hop counts, the branches of the multicast Source-Based Trees are minimum hop. If the metric is delay, the branches are minimum delay. For every multicast source, there is a corresponding multicast tree that directly connects the source to all receivers. All traffic to the members of an associated group passes along the tree made for their source. Source-Based Trees have two entries with a list of outgoing interfaces– the source address and the multicast group.
- **Shared Trees**: Shared trees or RP trees (RPT) rely on a central router called the Rendezvous Point (RP) that receives all traffic from the sources, and forwards that traffic to the receivers. All hosts might not be receivers. There is a single tree for each multicast group, regardless of the number of sources. Only the routers on the tree know about the group, and information is sent only to interested receivers. With an RP, receivers have a place to join, even if no source exists. The shared tree is unidirectional, and information flows only from the RP to the receivers. If a host other than the RP has to send data on the tree, the data must first be tunneled to the RP,

and then multicast to the members. This means that even if a receiver is also a source, it can only use the tree to receive packets from the RP, and not to send packets to the RP (unless the source is located between the RP and the receivers).

- **Bootstrap Router (BSR)**: When a new multicast sender starts sending data packets, or a new receiver starts sending the Join message towards the RP for that multicast group, it needs to know the next-hop router towards the RP. The BSR provides group-to-RP mapping information to all the PIMv6 routers in a domain, allowing them to map to the correct RP address.
- **Sending out Hello Messages**: PIMv6 routers periodically send Hello messages to discover neighboring PIMv6 routers. Hello messages are multicast using the address ff02::d (ALL-PIMv6-ROUTERS group). Routers do not send any acknowledgement that a Hello message was received. A hold time value determines the length of time for which the information is valid. In PIMv6-SM, a downstream receiver must join a group before traffic is forwarded on the interface.
- **Electing a Designated Router**: In a multi-access network with multiple routers connected, one of them is selected to act as a designated router (DR) for a given period of time. The DR is responsible for sending Join/Prune messages to the RP for local members.
- **Determining the RP**: PIMv6-SM uses a BootStrap Router (BSR) to originate Bootstrap messages, and to disseminate RP information. The messages are multicast to the group on each link. If the BSR is not apparent, the routers flood the domain with advertisements. The router with the highest priority (if priorities are same, the higher IPv6 address applies) is selected to be the RP. Routers receive and store Bootstrap messages originated by the BSR. When a DR gets a membership indication from MLD for (or a data packet from) a directly connected host, for a group for which it has no entry, the DR maps the group address to one of the candidate RPs that can service that group. The DR then sends a Join/Prune message towards that RP. In a small domain, the RP can also be configured statically.
- **Joining the Shared Tree**: To join a multicast group, a host sends an MLD message to its upstream router, after which the router can accept multicast traffic for that group. The router sends a Join message to its upstream PIMv6 neighbor in the direction of the RP. When a router receives a Join message from a downstream router, it checks to see if a state exists for the group in its multicast routing table. If a state already exists, the Join message has reached the shared tree, and the interface from which the message was received is entered in the Outgoing Interface list. If no state exists, an entry is created, the interface is entered in the Outgoing Interface list, and the Join message is again sent towards the RP.
- **Registering with the RP**: A DR can begin receiving traffic from a source without having a Source or a Group state for that source. In this case, the DR has no information on how to get multicast traffic to the RP through a tree. When the source DR receives the initial multicast packet, it encapsulates it in a Register message, and unicasts it to the RP for that group. The RP decapsulates each Register message, and forwards the extracted data packet to downstream members on the RPT. Once the path is established from the source to the RP, the DR begins sending traffic to the RP as standard IPv6 multicast packets, as well as encapsulated within Register messages. The RP temporarily receives packets twice. When the RP detects the normal multicast packets, it sends a Register-Stop message to the source DR, meaning it should stop sending register packets.
- **Sending Register-Stop Messages**: When the RP begins receiving traffic from the source, both as Register messages and as unencapsulated IPv6 packets, it sends a Register-Stop message to the DR. This notifies the DR that the traffic is now being received as standard IPv6 multicast packets on the SPT. When the DR receives this message, it stops encapsulating traffic in Register messages.
- **Pruning the Interface**: Routers attached to receivers send Prune messages to the RP to disassociate the source from the RP. When an RP receives a Prune message, it no longer forwards traffic from the source indicated in the Prune message. If all members of a multicast group are pruned, the MLD state of the DR is deleted, and the interface is removed from the Source and Group lists of the group.
- **Forwarding Multicast Packets**: PIMv6-SM routers forward multicast traffic onto all interfaces that lead to receivers that have explicitly joined a multicast group. Messages are sent to a group address in the local subnetwork, and have a Time to Live (TTL) of 1. The router performs an RPF check, and forwards the packet. Traffic that arrives on the correct interface is sent onto all outgoing interfaces that lead to downstream receivers if the downstream router has sent a join to this router, or is a member of this group.

15.3.2 Configuration

Configuring General PIMv6 Sparse-mode (With static RP)

PIMv6-SM is a soft-state protocol. The main requirement is to enable PIMv6-SM on desired interfaces, and configure the RP information correctly, through static or dynamic methods. All multicast group states are maintained dynamically as the result of MLD Report/Leave and PIMv6 Join/Prune messages. Currently, we support only one RP for all multicast groups (ff00::/8).

This section provides PIMv6-SM configuration examples for two relevant scenarios.

In this example, using the above topology, Switch1 is the Rendezvous Point (RP), and all routers are statically configured with RP information. While configuring the RP, make sure that:

- Every router includes the ipv6 pim rp-address 2001:1::1 statement, even if it does not have any source or group member attached to it.
- There is only one RP address for a group scope in the PIMv6 domain.
- All interfaces running PIMv6-SM must have sparse-mode enabled.

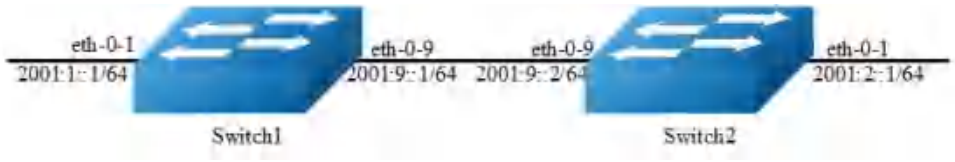


Figure 15-1 PIMv6 Sparse-mode

The graphic above displays the network topology used in these examples:

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable IPv6 & IPv6 multicast globally

```
Switch(config)# ipv6 enable
Switch(config)# ipv6 multicast-routing
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:9::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:2::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:9::2/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit
```

step 4 Create static unicast routes

Configuring Switch1:

```
Switch(config)# ipv6 route 2001:2::/64 2001:9::2
```

Configuring Switch2:

```
Switch(config)# ipv6 route 2001:1::/64 2001:9::1
```

step 5 Configure static RP address

```
Switch(config)# ipv6 pim rp-address 2001:1::1
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Configure all the routers with the same ipv6 pim rp-address 2001:1::1 command as shown above. Use the following commands to verify the RP configuration, interface details, and the multicast routing table.

RP Details

At Switch1, the show ip pim sparse-mode rp mapping command shows that 11.1.1.1 is the RP for all multicast groups ff00::/8, and is statically configured. All other routers will have a similar output.

```
Switch# show ipv6 pim sparse-mode rp mapping
PIM Group-to-RP Mappings
Group(s): ff00::/8, Static
  RP: 2001:1::1
     Uptime: 00:00:04
Embedded RP Groups:
```

Interface Details

The show ipv6 pim sparse-mode interface command displays the interface details for Switch1.

```
Switch# show ipv6 pim sparse-mode interface
Interface   VIFindex Ver/  Nbr   DR
                Mode  Count  Prior
eth-0-1     2       v2/S  0     1
  Address      : fe80::fc94:efff:fe96:2600
  Global Address: 2001:1::1
  DR           : this system
eth-0-9     0       v2/S  0     1
  Address      : fe80::fc94:efff:fe96:2600
  Global Address: 2001:9::1
  DR           : this system
```

IPv6 Multicast Routing Table

The show ipv6 pim sparse-mode mroute detail command displays the IPv6 multicast routing table.

Display the result on Switch1:

```
Switch# show ipv6 pim sparse-mode mroute detail
IPv6 Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 0
(S,G,rpt) Entries: 0
FCR Entries: 0
*, ff0e::1234:5678
Type: (*,G)
Uptime: 00:01:37
 RP: 2001:1::1, RPF nbr: None, RPF idx: None
 Upstream:
```

```
 State: JOINED, SPT Switch: Enabled, JT: off
 Macro state: Join Desired,
Downstream:
 eth-0-1:
  State: NO INFO, ET: off, PPT: off
  Assert State: NO INFO, AT: off
   Winner: ::, Metric: 4294967295, Pref: 4294967295, RPT bit: on
  Macro state: Could Assert, Assert Track
Local Olist:
 eth-0-1
```

Display the result on Switch2:

```
Switch# show ipv6 pim sparse-mode mroute detail
IPv6 Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 0
(S,G,rpt) Entries: 0
FCR Entries: 0
*, ff0e::1234:5678
Type: (*,G)
Uptime: 00:00:06
 RP: 2001:1::1, RPF nbr: None, RPF idx: None
 Upstream:
  State: JOINED, SPT Switch: Enabled, JT: off
  Macro state: Join Desired,
 Downstream:
  eth-0-1:
   State: NO INFO, ET: off, PPT: off
   Assert State: NO INFO, AT: off
    Winner: ::, Metric: 4294967295, Pref: 4294967295, RPT bit: on
   Macro state: Could Assert, Assert Track
 Local Olist:
  eth-0-1
```

Configuring General PIMv6 Sparse-mode (With dynamic RP)

A static configuration of RP works for a small, stable PIMv6 domain; however, it is not practical for a large and not-suitable internet work. In such a network, if the RP fails, the network administrator might have to change the static configurations on all PIMv6 routers. Another reason for choosing dynamic configuration is a higher routing traffic leading to a change in the RP.

We use the BSR mechanism to dynamically maintain the RP information. For configuring RP dynamically in the above scenario, Switch1 on eth-0-1 and Switch2 on eth-0-9 are configured as Candidate RP using the ipv6 pim rp candidate command. Switch2 on eth-0-9 is also configured as Candidate BSR. Since no other router has been configured as Candidate BSR, the Switch2 becomes the BSR router, and is responsible for sending group-to-RP mapping information to all other routers in this PIMv6 domain.

The following output displays the complete configuration at Switch1 and Switch2.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable IPv6 & IPv6 multicast globally

```
Switch(config)# ipv6 enable
Switch(config)# ipv6 multicast-routing
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:9::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:2::1/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:9::2/64
Switch(config-if)# ipv6 pim sparse-mode
Switch(config-if)# exit
```

step 4 Create static unicast routes

Configuring Switch1:

```
Switch(config)# ipv6 route 2001:2::/64 2001:9::2
```

Configuring Switch2:

```
Switch(config)# ipv6 route 2001:1::/64 2001:9::1
```

step 5 Configure the candidate rp

Configuring Switch1:

```
Switch(config)# ipv6 pim rp-candidate eth-0-1
```

Configuring Switch2:

```
Switch(config)# ipv6 pim rp-candidate eth-0-9
```

step 6 Configure the candidate bsr

Configuring Switch2:

```
Switch(config)# ipv6 pim bsr-candidate eth-0-9
```


The highest priority router is chosen as the RP. If two or more routers have the same priority, a hash function in the BSR mechanism is used to choose the RP, to make sure that all routers in the PIMv6-domain have the same RP for the same group.

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

PIMv6 group-to-RP mappings

Use the show ip pim sparse-mode rp mapping command to display the group-to-RP mapping details. The output displays information about RP candidates. There are two RP candidates for the group range ff00::/8. RP Candidate 2001:1::1 has a default priority of 192, whereas, RP Candidate 2001:9::2 has been configured to have a priority of 2. Since RP candidate 2001:1::1 has a higher priority, it is selected as RP for the multicast group ff00::/8. Only permit filters would be cared in group list.。

Display the result on Switch2:

```
Switch# show ipv6 pim sparse-mode rp mapping
PIM Group-to-RP Mappings
This system is the Bootstrap Router (v2)
Group(s): ff00::/8
 RP: 2001:9::2
   Info source: 2001:9::2, via bootstrap, priority 2
      Uptime: 00:00:32, expires: 00:02:02
 RP: 2001:1::1
   Info source: 2001:1::1, via bootstrap, priority 192
      Uptime: 00:00:31, expires: 00:02:03
Embedded RP Groups:
```

RP details

To display information about the RP router for a particular group, use the following command. This output displays that 2001:9::2 has been chosen as the RP for the multicast group ff02::1234.

Display the result on Switch2:

```
Switch# show ipv6 pim sparse-mode rp-hash ff02::1234

Info source: 2001:9::2, via bootstrap
```


After RP information reaches all PIMv6 routers in the domain, various state machines maintain all routing states as the result of Join/Prune from group membership. To display information on interface details and the multicast routing table, refer to the Configuring RP Statically section above.

Configuring Boostrap Router

Every PIMv6 multicast group needs to be associated with the IPv6 address of a Rendezvous Point (RP). This address is used as the root of a group-specific distribution tree whose branches extend to all nodes in the domain that want to receive traffic sent to the group. For all senders to reach all receivers, all routers in the domain use the same mappings of group addresses to RP addresses. In order to determine the RP for a multicast group, a PIMv6 router maintains a collection of group-to-RP mappings, called the RP-Set.

The Bootstrap Router (BSR) mechanism for the class of multicast routing protocols in the PIMv6 domain use the concept of a Rendezvous Point as a means for receivers to discover the sources that send to a particular multicast group. The BSR mechanism is one way that a multicast router can learn the set of group-to-RP mappings required in order to function.

Some of the PIMv6 routers within a PIMv6 domain are configured as Candidate-RPs (C-RPs). A subset of the C-RPs will eventually be used as the actual RPs for the domain. An RP configured with a lower value in the priority field has higher a priority.

Some of the PIMv6 routers in the domain are configured to be Candidate-BSRs (C-BSRs). One of these C-BSRs is elected to be the bootstrap router (BSR) for the domain, and all PIMv6 routers in the domain learn the result of this election through BSM (Bootstrap messages). The C-BSR with highest value in priority field is Elected-BSR.

The C-RPs then reports their candidacy to the elected BSR, which chooses a subset of the C-RPs and distributes corresponding group-to-RP mappings to all the routers in the domain through Bootstrap messages.

The following configuration should be operated on all switches if the switch ID is not specified.

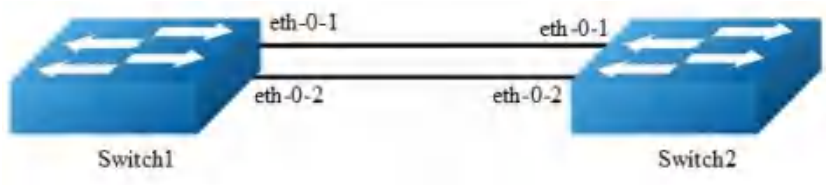


Figure 15-2 BSR

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable IPv6 & IPv6 multicast globally

```
Switch(config)# ipv6 enable
Switch(config)# ipv6 multicast-routing
```

step 3 Configure the candidate bsr

Configuring Switch1:

```
Switch(config)# ipv6 pim bsr-candidate eth-0-1
```

Configuring Switch2:

```
Switch(config)# ipv6 pim bsr-candidate eth-0-1 10 25
```

step 4 Configure the candidate rp

Configuring Switch2:

```
Switch(config)# ipv6 pim rp-candidate eth-0-1 priority 0
```

step 5 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# ipv6 pim dr-priority 10
Switch(config-if)# ipv6 pim unicast-bsm
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Verify the C-BSR state on rtr1

```
Switch# show ipv6 pim sparse-mode bsr-router
PIM6v2 Bootstrap information
This system is the Bootstrap Router (BSR)
 BSR address: 2001:9::1 (?)
 Uptime:    00:01:27, BSR Priority: 64, Hash mask length: 126
 Next bootstrap message in 00:00:16
```

```
Role: Candidate BSR
State: Elected BSR
```

Verify the C-BSR state on rtr2. The initial state of C-BSR is P-BSR before transitioning to C-BSR.

```
Switch# show ipv6 pim sparse-mode bsr-router
PIM6v2 Bootstrap information
 BSR address: 2001:9::1 (?)
 Uptime:     00:01:34, BSR Priority: 64, Hash mask length: 126
 Expires:    00:01:51
 Role: Candidate BSR
 State: Candidate BSR
 Candidate RP: 2001:9::2(eth-0-9)
   Advertisement interval 60 seconds
   Next C-RP advertisement in 00:00:35
```

Verify RP-set information on E-BSR

```
Switch# show ipv6 pim sparse-mode rp mapping
PIM Group-to-RP Mappings
This system is the Bootstrap Router (v2)
Group(s): ff00::/8
 RP: 2001:9::2
   Info source: 2001:9::2, via bootstrap, priority 0
      Uptime: 00:45:37, expires: 00:02:29
Embedded RP Groups:
```

Verify RP-set information on C-BSR

```
Switch# show ipv6 pim sparse-mode rp mapping
PIM Group-to-RP Mappings
Group(s): ff00::/8
 RP: 2001:9::2
   Info source: 2001:9::1, via bootstrap, priority 0
      Uptime: 00:03:14, expires: 00:01:51
Embedded RP Groups:
```

Configuring PIMv6-SSM feature

PIMv6-SSM can work with PIMv6-SM on the multicast router. By default, PIMv6-SSM is disabled.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable PIMv6-ssm globally

```
Switch(config)# ipv6 pim ssm default
Switch(config)# ipv6 pim ssm range ipv6acl
```

step 3 Exit the configure mode

```
Switch(config)# end
```

15.4 Configuring PIMv6-DM

15.4.1 Overview

Function Introduction

The Ipv6 Protocol Independent Multicasting-Dense Mode (PIMv6-DM) is a multicast routing protocol designed to operate efficiently across Wide Area Networks (WANs) with densely distributed groups. It helps network nodes that are geographically dispersed to conserve bandwidth, and reduces traffic by simultaneously delivering a single stream of information to multiple locations.

PIMv6-DM assumes that when a source starts sending, all down stream systems want to receive multicast datagrams. Initially, multicast datagrams are flooded to all areas of the network. PIMv6-DM uses RPF to prevent looping of multicast datagrams while flooding. If some areas of the network do not have group members, PIMv6-DM will prune off the forwarding branch by instantiating prune state.

Prune state has a finite lifetime. When that lifetime expires, data will again be forwarded down the previously pruned branch. Prune state is associated with an (S,G) pair. When a new member for a group G appears in a pruned area, a router can "graft" toward the source S for the group, thereby turning the pruned branch back into a forwarding branch.

Principle Description

The PIMv6-DM module is based on the following IETF standard: RFC 3973

15.4.2 Configuration

Configuring General PIM dense-mode

PIMv6-DM is a soft-state protocol. The main requirement is to enable PIMv6-DM on desired interfaces. All multicast group states are maintained dynamically as the result of MLD Report/Leave and PIMv6 messages.

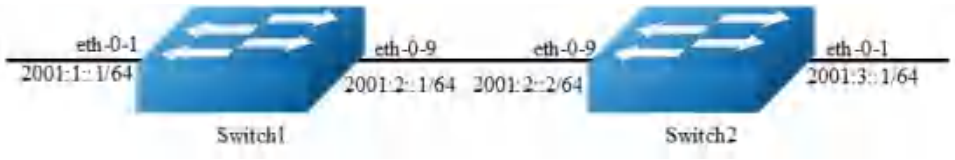


Figure 15-3 PIMv6 dense-mode

This section provides PIMv6-DM configuration examples for two relevant scenarios. The following graphic displays the network topology used in these examples: In this example, using the above topology, multicast data stream comes to eth-0-1 of Switch1, host is connected to eth-0-1 of Switch2. Here is a sample configuration:

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable IPv6 & IPv6 multicast globally

```
Switch(config)# ipv6 enable
Switch(config)# ipv6 multicast-routing
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:1::1/64
Switch(config-if)# ipv6 pim dense-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:2::1/64
Switch(config-if)# ipv6 pim dense-mode
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:3::1/64
Switch(config-if)# ipv6 pim dense-mode
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:2::2/64
Switch(config-if)# ipv6 pim dense-mode
Switch(config-if)# exit
```

step 4 Create static unicast routes

Configuring Switch1:

```
Switch(config)# ipv6 route 2001:3::/64 2001:2::2
```

Configuring Switch2：

```
Switch(config)# ipv6 route 2001:1::/64 2001:2::1
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Interface Details

The show ipv6 pim dense-mode interface command displays the interface details for Switch1.

```
Switch# show ipv6 pim dense-mode interface
Neighbor Address                Interface   VIFIndex Ver/   Nbr
                                 Mode   Count
fe80::326f:c9ff:fef2:8200        eth-0-1    0      v2/D  0
fe80::326f:c9ff:fef2:8200        eth-0-9    2      v2/D  1
```

Neighbor Details

Use *show IPv6 Pim dense mode neighbors* to display detailed information about neighbors on Switch1

```
Switch# show ipv6 pim sparse-mode neighbor
Neighbor Address                Interface   Uptime/Expires    Ver
fe80::ce47:6eff:feb7:1400          eth-0-9    00:51:51/00:01:24 v2
```

IP Multicast Routing Table

The show ip pim dense-mode mroute detail command displays the IP multicast routing table.

Display the result on Switch1:

```
Switch# show ipv6 pim dense-mode mroute
PIM-DM Multicast Routing Table
(2001:1::2, ff0e::1)
 Source directly connected on eth-0-1
 State-Refresh Originator State: Originator
 Upstream IF: eth-0-1
  Upstream State: Forwarding
```

```
  Assert State: NoInfo
Downstream IF List:
  eth-0-9, in 'olist':
    Downstream State: NoInfo
    Assert State: NoInfo
```

Display the result on Switch2:

```
Switch# show ipv6 pim dense-mode mroute
PIM-DM Multicast Routing Table
(2001:1::2, ff0e::1)
 RPF Neighbor: none
 Upstream IF: eth-0-9
  Upstream State: AckPending
  Assert State: Loser
 Downstream IF List:
  eth-0-1, in 'olist':
    Downstream State: NoInfo
    Assert State: NoInfo
```

15.5 Configuring MLD Snooping

15.5.1 Overview

Function Introduction

Layer 2 switches can use MLD snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded only to those interfaces associated with IPv6 multicast devices. As the name implies, MLD snooping requires the LAN switch to snoop on the MLD transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an MLD report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an MLD Leave Group message from a host, it removes the host port from the table entry. It also deletes entries per entry if it does not receive MLD membership reports from the multicast clients. The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send report and are added to the forwarding table entry. The switch forwards only one report per IPv6 multicast group to the multicast router. It creates one entry per VLAN in the Layer 2 forwarding table for each MAC group from which it receives an MLD report.

Layer 2 multicast groups learned through MLD snooping are dynamic. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by MLD snooping. Multicast group membership lists can consist of both user-defined and MLD snooping-learned settings.

 Limitations And Configuration Guideline

VRRP, RIPng and OSPFv3 used multicast IPv6 address, so you need to avoid use such multicast IPv6 addresses, which have same multicast MAC address with multicast IPv6 address reserved by VRRP, RIPng and OSPFv3.

- VRRP used multicast group address ff02::12, so when mld snooping and VRRP are working, you need to avoid using multicast group address that matched same mac address with group address ff02::12.
- OSPFv3 used multicast group address ff02::5, so when mld snooping and OSFPv3 are working, you need to avoid using multicast group address that matched same mac address with group address ff02::5.
- RIPng used multicast group address ff02::9, so when mld snooping and RIPng are working, you need to avoid using multicast group address that matched same mac address with group address ff02::9.

Principle Description

N/A

15.5.2 Configuration

Enable MLD Snooping

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable mld snooping globally

```
Switch(config)# ipv6 mld snooping
```

step 3 vlan 使能 mld snooping

```
Switch(config)#ipv6 mld snooping vlan 1
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch # show ipv6 mld snooping vlan 1
Global Mld Snooping Configuration
------------------------------------------------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Disabled
Mld Snooping Version            :1
Mld Snooping Max-Member-Number   :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Report-Suppression   :Enabled
Vlan 1
-----------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Disabled
Mld Snooping Report-Suppression   :Enabled
Mld Snooping Version            :1
Mld Snooping Max-Member-Number   :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Group Access-list   :N/A
Mld Snooping Mrouter Port       :
Mld Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Fast Leave

When MLD Snooping fast leave is enabled, the mld snooping group will be removed at once upon receiving a corresponding mld report. Otherwise the switch will send out specified mld specific query, if it doesn't get response in specified period, it will remove the group. By default, mld snooping fast-leave is disabled globally and per vlan.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable fast leave globally

```
Switch(config)# ipv6 mld snooping fast-leave
```

step 3 Enable fast leave for a vlan

```
Switch(config)# ipv6 mld snooping vlan 1 fast-leave
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show ipv6 mld snooping vlan 1
Global Mld Snooping Configuration
------------------------------------------------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Enabled
Mld Snooping Version            :1
Mld Snooping Max-Member-Number  :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Report-Suppression  :Enabled
Vlan 1
-----------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Enabled
Mld Snooping Report-Suppression  :Enabled
Mld Snooping Version            :1
Mld Snooping Max-Member-Number  :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Group Access-list  :N/A
Mld Snooping Mrouter Port       :
Mld Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Querier Parameters (optional)

In order for MLD, and thus MLD snooping, to function, a multicast router must exist on the network and generate MLD queries. The tables created for snooping (holding the member ports for each multicast group) are associated with the querier. Without a querier the tables are not created and snooping will not work.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configuring Querier Parameters for MLD snooping

Set mld snooping query interval and max query response time:

```
Switch(config)# ipv6 mld snooping query-interval 100
Switch(config)# ipv6 mld snooping query-max-response-time 5
```

Set mld snooping last member query interval:

```
Switch(config)# ipv6 mld snooping last-member-query-interval 2000
```

Set mld snooping query parameters for vlan 1:

```
Switch(config)# ipv6 mld snooping vlan 1 querier address fe80::1
Switch(config)# ipv6 mld snooping vlan 1 querier
Switch(config)# ipv6 mld snooping vlan 1 query-interval 200
Switch(config)# ipv6 mld snooping vlan 1 query-max-response-time 5
Switch(config)# ipv6 mld snooping vlan 1 querier-timeout 100
Switch(config)# ipv6 mld snooping vlan 1 last-member-query-interval 2000
Switch(config)# ipv6 mld snooping vlan 1 discard-unknown
```

Discard unknown multicast packets globally:

```
Switch(config)# ipv6 mld snooping discard-unknown
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch # show ipv6 mld snooping querier
Global Mld Snooping Querier Configuration
---------------------------------------------------
Version                    :1
Last-Member-Query-Interval (msec) :2000
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)           :100
Global Source-Address          :::
TCN Query Count                :2
TCN Query Interval (sec)       :10
Vlan 1:  MLD snooping querier status
---------------------------------------------
Elected querier is : fe80::1
---------------------------------------------
Admin state                :Enabled
Admin version              :1
Operational state          :Querier
Querier operational address     :fe80::1
Querier configure address       :fe80::1
Last-Member-Query-Interval (msec) :2000
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)           :200
Querier-Timeout (sec)          :100
```

Configuring Mrouter Port

An MLD Snooping mrouter port is a switch port which is assumed to connect a multicast router. The mrouter port is configured on the vlan or learnt dynamicly. When MLD general query packet or PIMv6 hello packet is received on port of specified VLAN, this port becomes mrouter port of this vlan. All the mld queries received on this port will be flooded on the belonged vlan. All the mld reports and leaves received on this vlan will be forwarded to the mrouter port, directly or aggregated, depending on the report-suppression configuration. In addition, all the multicast traffic on this vlan will be forwarded to this mrouter port.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable mld snooping report suppression globally

```
Switch(config)# ipv6 mld snooping report-suppression
```

step 3 Configure mrouter port

```
Switch(config)# ipv6 mld snooping vlan 1 mrouter interface eth-0-1
```

step 4 Configure mld snooping for parameters vlan

Enable mld snooping report suppression and Set mld snooping dynamic mrouter port aging interval:

```
Switch(config)# ipv6 mld snooping vlan 1 report-suppression
Switch(config)# ipv6 mld snooping vlan 1 mrouter-aging-interval 200
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show ipv6 mld snooping vlan 1
Global Mld Snooping Configuration
```

```
-----------------------------------------------------
Mld Snooping                        :Enabled
Mld Snooping Fast-Leave             :Enabled
Mld Snooping Version                :1
Mld Snooping Max-Member-Number      :4096
Mld Snooping Unknown Multicast Behavior  :Discard
Mld Snooping Report-Suppression     :Enabled
Vlan 1
-----------
Mld Snooping                        :Enabled
Mld Snooping Fast-Leave             :Enabled
Mld Snooping Report-Suppression     :Enabled
Mld Snooping Version                :1
Mld Snooping Max-Member-Number      :4096
Mld Snooping Unknown Multicast Behavior  :Discard
Mld Snooping Group Access-list      :N/A
Mld Snooping Mrouter Port           :eth-0-1(static)
Mld Snooping Mrouter Port Aging Interval(sec) :200
```

Configuring Querier Tcn

User can set the TCN interval and query count to adapt the multicast learning and updating after STP converging.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the parameters for MLD Snooping querier TCN

Set mld snooping querier tcn query count and interval:

```
Switch(config)# ipv6 mld snooping querier tcn query-count 5
Switch(config)# ipv6 mld snooping querier tcn query-interval 20
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch # show ipv6 mld snooping querier
Global Mld Snooping Querier Configuration
-----------------------------------------------------
Version                     :1
Last-Member-Query-Interval (msec) :2000
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)          :100
Global Source-Address          :::
TCN Query Count               :5
TCN Query Interval (sec)      :20
Vlan 1:  MLD snooping querier status
-----------------------------------------------
Elected querier is : fe80::1
-----------------------------------------------
Admin state                 :Enabled
Admin version               :1
Operational state           :Querier
Querier operational address     :fe80::1
Querier configure address       :fe80::1
```

```
Last-Member-Query-Interval (msec) :2000
Max-Query-Response-Time (sec)    :5
Query-Interval (sec)             :200
Querier-Timeout (sec)            :100
```

Configuring Report Suppression

The switch uses MLD report suppression to forward only one MLD report per multicast router query to multicast devices. When MLD router suppression is enabled (the default), the switch sends the first MLD report from all hosts for a group to all the multicast routers. The switch does not send the remaining MLD reports for the group to the multicast routers. This feature prevents duplicate reports from being sent to the multicast devices.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable mld snooping report suppression globally

```
Switch(config)# ipv6 mld snooping report-suppression
```

step 3 Enable mld snooping report suppression for a vlan

```
Switch(config)# ipv6 mld snooping vlan 1 report-suppression
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch # show ipv6 mld snooping
Global Mld Snooping Configuration
---------------------------------------------------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Disabled
Mld Snooping Version            :2
Mld Snooping Max-Member-Number   :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Report-Suppression   :Enabled
Vlan 1
-----------
Mld Snooping                    :Enabled
Mld Snooping Fast-Leave         :Disabled
Mld Snooping Report-Suppression   :Enabled
Mld Snooping Version            :2
Mld Snooping Max-Member-Number   :4096
Mld Snooping Unknown Multicast Behavior  :Flood
Mld Snooping Group Access-list   :N/A
Mld Snooping Mrouter Port        :
Mld Snooping Mrouter Port Aging Interval(sec) :255
```

Configuring Static group

The switch can build MLD Snooping Group when receiving MLD report packet on Layer 2 port of specified VLAN. We also support configure static MLD Snooping Group by specifying MLD group, Layer 2 port and VLAN.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure static group

```
Switch(config)# ipv6 mld snooping vlan 1 static-group ff0e::1234 interface eth-0-2
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch# show ipv6 mld snooping groups
VLAN  Interface  Group Address                Uptime    Expire-time
1     eth-0-2    ff0e::1234                  00:00:02  stopped
```

15.5.3 Application cases

N/A

15.6 Configuring MVR6

15.6.1 Overview

Function Introduction

Multicast VLAN Registration for IPv6 (MVR6) is designed for applications using wide-scale deployment of IPv6 multicast traffic across an Ethernet ring-based service provider network (for example, the broadcast of IPv6 multiple television channels over a service-provider network). MVR6 allows a subscriber on a port to subscribe and unsubscribe to an IPv6 multicast stream on the network-wide multicast VLAN. It allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs. MVR6 provides the ability to continuously send IPv6 multicast streams in the multicast VLAN, but to isolate the streams from the subscriber VLANs for bandwidth and security reasons.

MVR6 assumes that subscriber ports subscribe and unsubscribe (join and leave) these multicast streams by sending out MLD join and leave messages. These messages can originate from an MLD version-1-compatible host with an Ethernet connection. Although MVR6 operates on the underlying mechanism of MLD snooping, the two features operation affect with each other. One can be enabled or disabled with affecting the behavior of the other feature. If MLD snooping and MVR6 are both enabled, MVR6 reacts only to join and leave messages from IPv6 multicast groups configured under MVR6. The switch CPU identifies the MVR6 IPv6 multicast streams and their associated MAC addresses in the switch forwarding table, intercepts the MLD messages, and modifies the forwarding table to include or remove the subscriber as a receiver of the multicast stream, and the receivers must be in a different VLAN from the source. This forwarding behavior selectively allows traffic to cross between different VLANs.

15.6.2 Configuration

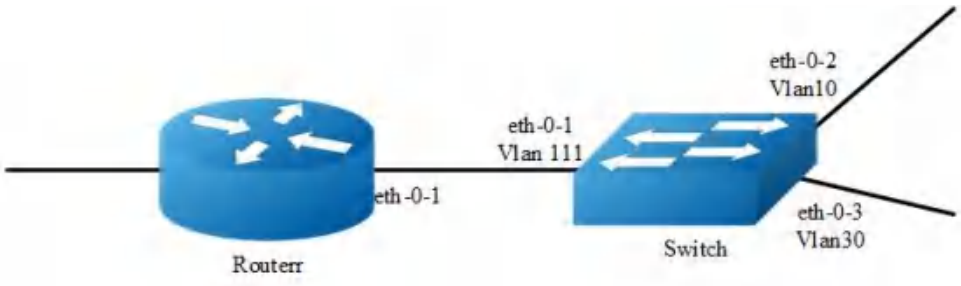


Figure 15-4 MVR6

step 1 Enter the configure mode

Configuring Switch:

```
Switch# configure terminal
```

Configuring Router:

```
Router# configure terminal
```

step 2 Enter the vlan configure mode and create VLANs

Configuring Switch:

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 111,10,30
Switch(config-vlan)# quit
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Router:

```
Router(config)# interface eth-0-1
Router(config-if)# no switchport
Router(config-if)# no shutdown
Router(config-if)# ipv6 address 2001:1::1/64
Router(config-if)# ipv6 pim sparse-mode
Router(config-if)# end
```

Interface configuration for Switch:

```
Switch(config)# interface vlan 111
Switch(config-if)# exit

Switch(config)# interface vlan 10
Switch(config-if)# exit

Switch(config)# interface vlan 30
Switch(config-if)# exit

Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan111
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport access vlan10
Switch(config-if)# exit

Switch(config)# interface eth-0-3
Switch(config-if)# switchport access vlan30
Switch(config-if)# exit
```

step 4 Enable MVR6

Eanble MVR6 in the switch, it is required that only one copy of IPv6 multicast traffic from the Router is sent to the switch, but the hosts can both receiver this IPv6 multicast traffic.

```
Switch(config)# no ipv6 multicast-routing
Switch(config)# mvr6
Switch(config)# mvr6 vlan 111
Switch(config)# mvr6 group ff0e::1234 64
Switch(config)# mvr6 source-address fe80::1111

Switch(config)# interface eth-0-1
Switch(config-if)# mvr6 type source
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# mvr6 type receiver vlan 10
```

```
Switch(config-if)# exit

Switch(config)# interface eth-0-3
Switch(config-if)# mvr6 type receiver vlan 30
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Router:

```
Router# show ipv6 mld groups
MLD Connected Group Membership
Group Address                Interface        Expires
ff0e::1234                   eth-0-2          00:03:01
ff0e::1235                   eth-0-2          00:03:01
ff0e::1236                   eth-0-2          00:03:01
ff0e::1237                   eth-0-2          00:03:01
ff0e::1238                   eth-0-2          00:03:01
……
ff0e::1273                   eth-0-2          00:03:01
```

Display the result on Switch:

```
Switch# show mvr6
MVR6 Running: TRUE
MVR6 Multicast VLAN: 111
MVR6 Source-address: fe80::111
MVR6 Max Multicast Groups: 1024
MVR6 Hw Rt Limit: 224
MVR6 Current Multicast Groups: 64


VLAN  Interface  Group Address                Uptime    Expire-time
10    eth-0-2   ff0e::1234                   00:03:23  00:02:03
10    eth-0-2   ff0e::1235                   00:03:23  00:02:03
10    eth-0-2   ff0e::1236                   00:03:23  00:02:03
10    eth-0-2   ff0e::1237                   00:03:23  00:02:03
10    eth-0-2   ff0e::1238                   00:03:23  00:02:03
10    eth-0-2   ff0e::1239                   00:03:23  00:02:03
……
10    eth-0-2   ff0e::1273                   00:03:23  00:02:03
```

16 VPN Configuration Guide

16.1 Configuring VPN

16.1.1 Overview

Function Introduction

VPN is defined as a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, where the service provider associates each interface with a VPN routing table. A VPN routing table is called a VPN routing and forwarding (VRF) table. Beginning in privileged EXEC mode, follow these steps to configure one or more VRFs.

Principle Description

N/A

16.1.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a vrf instance

```
Switch(config)# ip vrf vpn1
Switch(config-vrf)# rd 100:1
Switch(config-vrf)# router-id 1.1.1.1
Switch(config-vrf)# route-target both 100:1
Switch(config-vrf)# import map route-map
```

 Enter either an AS system number

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config-vrf)# interface eth-0-1
Switch(config-if)# no shutdown
Switch(config-if)# no switch
Switch(config-if)# ip vrf forwarding vpn1
Switch(config-if)# ip add 1.1.1.1/24
Switch(config-if)# end
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

The result of show information about the configured VRFs:

```
Switch# show ip vrf
VRF vpn1, FIB ID 1
Router ID: 1.1.1.1 (config)
Interfaces:
 eth-0-1

Switch# show ip vrf interfaces vpn1
Interface          IP-Address     VRF               Protocol
eth-0-1            1.1.1.1        vpn1              up
```

```
Switch# show ip vrf bgp brief
Name              Default RD        Interfaces
vpn1              100:1             eth-0-1

Switch# show ip vrf bgp detail
VRF vpn1; default RD 100:1
Interfaces:
 eth-0-1
VRF Table ID = 1
Export VPN route-target communities
 RT:100:1
Import VPN route-target communities
 RT:100:1
import-map: route-map
No export route-map
```

16.2 Configuring IPv4 GRE Tunnel

16.2.1 Overview

Function Introduction

Tunneling is an encapsulation technology, which uses one network protocol to encapsulate packet of another network protocol and transfer them over a virtual point to point connection. The virtual connection is called a tunnel. Tunneling refers to the whole process from data encapsulation to data transfer to data de-encapsulation.

Principle Description

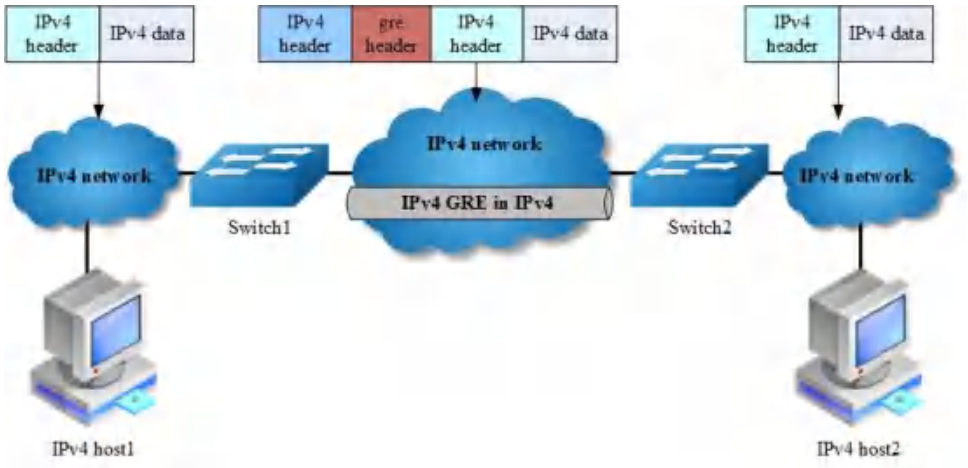


Figure 16-1 IPv4 gre over IPv4

When it is required to communicate with isolated IPv4 networks, you should create a tunnel mechanism between them. The tunnel with transmit protocol of gre connected with two isolated IPv4 island is called IPv4 gre tunnel, which is that IPv4 packets are encapsulated by gre protocol over outer IPv4 packets. Gre tunnel would add gre head in encapsulated packets, including key, sequence, checksum and so on. In order to make an implement of gre tunnel, both tunnel endpoints must support the IPv4 protocol stacks.

IPv4 gre tunnel processes packets in the following ways:

- A host in the IPv4 network sends an IPv4 packet to Switch1 at the tunnel source.
- After determining according to the routing table that the packet needs to be forwarded through the tunnel, Switch1 encapsulates the IPv4 packet with an IPv4 header and forwards it through the physical interface of the tunnel.
- Upon receiving the packet, Switch2 de-encapsulates the packet.

- Switch2 forwards the packet according to the destination address in the de-encapsulated IPv4 packet. If the destination address is the device itself, Switch2 forwards the IPv4 packet to the upper-layer protocol for processing. In the process of de-encapsulation, it would check gre key, only the matched key of packet can be processed, otherwise discarded.

The ip address of tunnel source and tunnel destination is manually assigned, and it provides point-to-point connection. By using overlay tunnels, you can communicate with isolated IPv4 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between border routers and a host.

The primary use is for stable connections that require regular secure communication between two edge routers or between an end system and an edge router, or for connection to remote IPv4 networks, gre key is alternative configuration.

16.2.2 Configuration

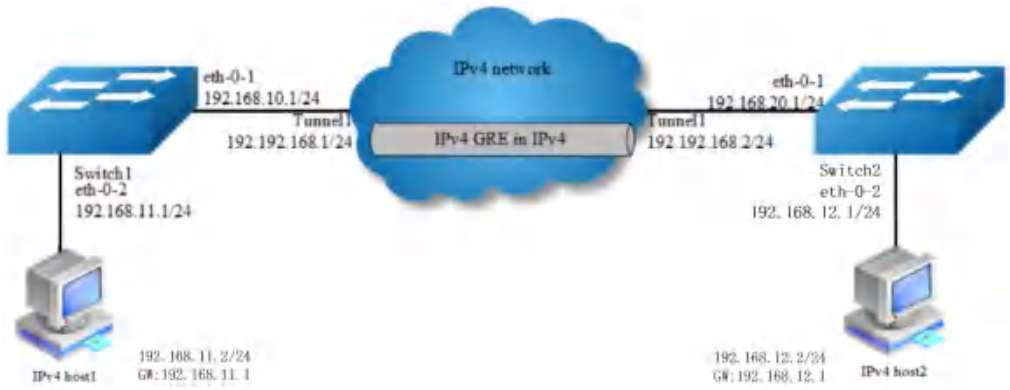


Figure 16-2 IPv4 gre Tunnel

As the topology shows, two IPv4 networks connect to the network via Switch1 and Switch2. An Ipv4 gre tunnel is required between Switch1 and Switch2, in order to connect two networks.


A reachable Ipv4 route is necessary for forwarding tunnel packet. Ipv4 address must be configured on tunnel interface; otherwise the route via this tunnel interface is invalid.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.10.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.11.1/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.20.1/24
Switch(config-if)# tunnel enable
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.12.1/24
Switch(config-if)# exit
```

step 3 Configure the tunnel interface

Tunnel interface configuration for Switch1:

```
Switch(config)# interface tunnel1
Switch(config-if)# tunnel mode gre
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel destination 192.168.20.1
Switch(config-if)# tunnel gre key 100
Switch(config-if)# ip address 192.192.168.1/24
Switch(config-if)# keepalive 5 3
Switch(config-if)# exit
```

Tunnel interface configuration for Switch2:

```
Switch(config)# interface tunnel1
Switch(config-if)# tunnel mode gre
Switch(config-if)# tunnel source eth-0-1
Switch(config-if)# tunnel destination 192.168.10.1
Switch(config-if)# tunnel gre key 100
Switch(config-if)# ip address 192.192.168.2/24
Switch(config-if)# keepalive 5 3
Switch(config-if)# exit
```

step 4 Configure the static route and arp

Configuring Switch1:

```
Switch(config)# ip route 192.168.20.0/24 192.168.10.2
Switch(config)# ip route 192.168.12.0/24 tunnel1
```

Configuring Switch2:

```
Switch(config)# ip route 192.168.10.0/24 192.168.20.2
Switch(config)# ip route 192.168.11.0/24 tunnel1
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
```

```
Index 8193 , Metric 1 , Encapsulation TUNNEL
VRF binding: not bound
Internet primary address:
 192.192.168.1/24 pointopoint 192.192.168.255
Tunnel protocol/transport GRE/IP, Status Valid
Tunnel source 192.168.10.1(eth-0-1), destination 192.168.20.1
Tunnel DSCP inherit,  Tunnel TTL 255
Tunnel GRE key enable: 100
Tunnel GRE keepalive enable, Send period: 5, Retry times: 3
0 packets input, 0 bytes
0 packets output, 0 bytes
```

Display the result on Switch2:

```
Switch# show interface tunnel1
Interface tunnel1
 Interface current state: UP
 Hardware is Tunnel
 Index 8193 , Metric 1 , Encapsulation TUNNEL
 VRF binding: not bound
 Internet primary address:
  192.192.168.2/24 pointopoint 192.192.168.255
 Tunnel protocol/transport GRE/IP, Status Valid
 Tunnel source 192.168.20.1(eth-0-1), destination 192.168.10.1
 Tunnel DSCP inherit,  Tunnel TTL 255
 Tunnel GRE key enable: 100
 Tunnel GRE keepalive enable, Send period: 5, Retry times: 3
 0 packets input, 0 bytes
 0 packets output, 0 bytes
```

16.2.3 Application cases

N/A

16.3 Configuring IPv4 ERSPAN Tunnel

16.3.1 Overview

Brief Introduction

ERSPAN (Encapsulated Remote Switch Port Analyzer) is a Layer 3 remote port mirroring technology that works by copying specified packets and sending the copied packets through a tunnel to a remote data monitoring device, allowing the user to analyse these packets (known as mirrored packets) with the data monitoring device, to for network monitoring and troubleshooting.

16.3.2 Configuration Examples

Configuring ERSPAN Single-Destination Address Mode

Topology

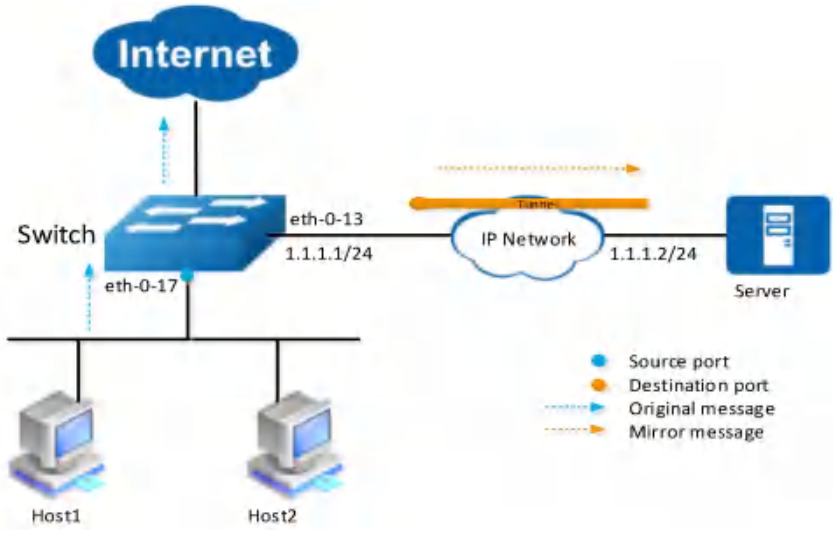


Figure 16-3 IPv4 ERSPAN Tunnel Single-Destination Address Mode

Configuration Steps

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface loopback0
Switch(config-if)# ip address 192.168.1.1/32
Switch(config-if)# exit
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit
```

step 3 Configure the tunnel interface

```
Switch(config)# interface tunnel0
Switch(config-if)# tunnel mode erspan
Switch(config-if)# tunnel source 192.168.1.1
Switch(config-if)# tunnel destination 1.1.1.2
Switch(config-if)# exit
```

step 4 Configure the source and destination ports of the local mirror group

```
Switch(config)# monitor session 1 source interface eth-0-17 both
Switch(config)# monitor session 1 destination interface tunnel0
```

step 5 Exit the configure mode

```
Switch(config)# end
```

Validation

```
Switch# show interface tunnel0
Interface tunnel0
 Interface current state: UP
```

```
Hardware is Tunnel
Index 9216 , Metric 0 , Encapsulation TUNNEL
VRF binding: not bound
Tunnel protocol/transport gre/erspan, Status Valid
Tunnel source 192.168.1.1(loopback0)
Tunnel destination 1.1.1.2
Tunnel DSCP inherit,  Tunnel TTL 254
Tunnel GRE key disable
Tunnel extend header disable
```

Configuring ERSPAN Multi-Destination Address ModeTopology

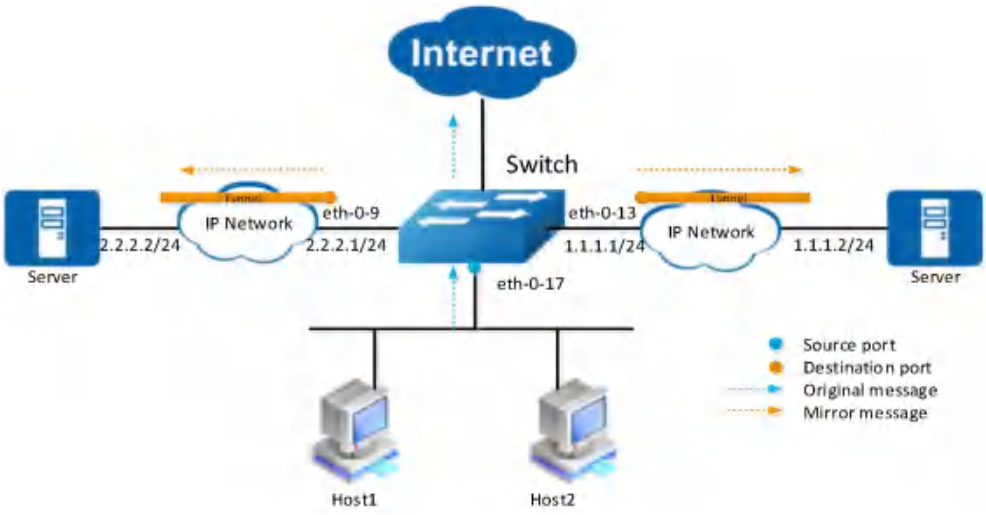


Figure 16-4 IPv4 ERSPAN Tunnel Multi-Destination Address Mode

Configuration Steps

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface loopback0
Switch(config-if)# ip address 192.168.1.1/32
Switch(config-if)# exit
Switch(config)# interface eth-0-13
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 1.1.1.1/24
Switch(config-if)# exit
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 2.2.2.1/24
Switch(config-if)# exit
```

step 3 Configure the tunnel interface

```
Switch(config)# interface tunnel0
Switch(config-if)# tunnel mode erspan ecmp-dst-gre
Switch(config-if)# tunnel source 192.168.1.1
```

```
Switch(config-if)# tunnel ecmp-destination 2.2.2.2
Switch(config-if)# tunnel ecmp-destination 1.1.1.2
Switch(config-if)# exit
```

step 4 Configure the source and destination ports of the local mirror group

```
Switch(config)# monitor session 1 source interface eth-0-17 both
Switch(config)# monitor session 1 destination interface tunnel0
```

step 5 Exit the configure mode

```
Switch(config)# end
```

Validation

```
Switch# show interface tunnel0
Interface tunnel0
 Interface current state: UP
 Hardware is Tunnel
 Index 9216 , Metric 0 , Encapsulation TUNNEL
 VRF binding: not bound
 Tunnel protocol/transport gre/ecmp-dst, Status Valid
 Tunnel source 192.168.1.1(loopback0)
 Tunnel ecmp-destination 2.2.2.2
                         1.1.1.2
 Tunnel DSCP inherit,  Tunnel TTL 254
 Tunnel GRE key disable
 Tunnel extend header disable
```

17 Reliability Configuration Guide

17.1 Configuring BHM

17.1.1 Overview

Function Introduction

BHM is a module which is used to monitor other Processes. When a monitored Process is uncontrolled, the BHM module will take measures, such as printing warning on screen, shutting all ports, or restarting the system, to help or remind users to recover the system.

The monitored Processes include RIP, RIPNG, OSPF, OSPF6, BGP, LDP, RSVP, PIM, PIM6, 802.1X, LACP MSTP, DHCP-RELAY, DHCP-RELAY6, RMON, OAM, ONM, SSH, SNMP, PTP, SSM. In addition, some system procedures are also monitored, including NSM, IMI, CHSM, HSRVD. There are three activations of BHM, including "reload system", including "reload system"/"warning", "shutdown port".

Principle Description

N/A

17.1.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable system monitor and heart-beat-monitor globally

```
Switch(config)# sysmon enable
Switch(config)# heart-beat-monitor enable
```

step 3 Reload system if a monitored PM is uncontrolled

```
Switch(config)# heart-beat-monitor reactivate reload system
```


There are three activations of BHM, including "reload system"/"warning", "shutdown port".

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show heart-beat-monitor
heart-beat-monitor enable.
heart-beat-monitor reactivation: restart system.
```

17.1.3 Application cases

N/A

17.2 Configuring EFM OAM

17.2.1 Overview

Function Introduction

This chapter contains a complete sample EFM OAM configuration. To see details on the commands used in this example, or to see the outputs of the validation commands, refer to the OAM Command Reference. To avoid repetition, some Common commands, like configure terminal, have not been listed under the commands used sections.

The main functions of Ethernet to the First Mile - Operation Administration and Maintenance (EFM-OAM) are link performance monitoring, fault detection, fault signaling and loopback signaling. OAM information is conveyed in Slow Protocol frames called OAM Protocol Data Units (OAMPDUs). OAMPDUs contain the appropriate control and status information used to monitor, test and troubleshoot OAM-enabled links.

Principle Description

Reference: IEEE 802.3ah (2004)

17.2.2 Configuration

Configuring Enable EFM

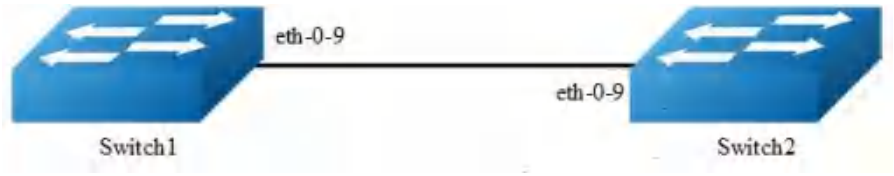


Figure 17-1 EFM

The following configurations are same on Switch1 and Switch2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and enable ethernet oam

```
Switch(config)# interface eth-0-9
Switch(config-if)# ethernet oam enable
Switch(config-if)# ethernet oam mode active
Switch(config-if)# ethernet oam link-monitor frame threshold high 10 window 50
Switch(config-if)# exit
```


ethernet oam mode can be “active” or “passive”. For example:

```
Switch(config-if)# ethernet oam mode passive
```

At least one switch among Switch1 and Switch2 should use mode active. Both switch use active can also work normally.

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

The EFM Discovery Machine State should be “send any” in both machines. This is the expected normal operating state for OAM on fully-operational links.

The various states of OAM discovery state machine are defined below.

- ACTIVE_SEND_LOCAL: A DTE configured in Active mode sends Information OAMPDUs that only contain the Local Information TLV. This state is called ACTIVE_SEND_LOCAL. While in this state, the local DTE waits for Information OAMPDUs received from the remote DTE.
- PASSIVE_WAIT: DTE configured in Passive mode waits until receiving Information OAMPDUs with Local Information TLVs before sending any Information OAMPDUs with Local Information TLVs. This state is called PASSIVE_WAIT. By waiting until first receiving an Information OAMPDU with the Local Information TLV, a Passive DTE cannot complete the OAM Discovery process when connected to another Passive DTE.
- SEND_LOCAL_REMOTE: Once the local DTE has received an Information OAMPDU with the Local Information TLV from the remote DTE, the local DTE begins sending Information OAMPDUs that contain both the Local and Remote Information TLVs. This state is called SEND_LOCAL_REMOTE.

- SEND_LOCAL_REMOTE_OK: If the local OAM client deems the settings on both the local and remote DTEs are acceptable, it enters the SEND_LOCAL_REMOTE_OK state.
- SEND_ANY: Once an OAMPDU has been received indicating the remote device is satisfied with the respective settings, the local device enters the SEND_ANY state. This is the expected normal operating state for OAM on fully operational links.
- FAULT: If OAM is reset, disabled, or the link timer expires, the Discovery process returns to the FAULT state.

Display results on Switch1:

```
Switch# show ethernet oam discovery interface eth-0-9
eth-0-9
Local client:
-------------
 Administrative configurations:
   Mode:                  active
   Unidirection:            not supported
   Link monitor:            supported(on)
   Remote Loopback:             not supported
   MIB retrieval:           not supported
   MTU Size   :             1518
 Operational status:
   Port status:             send any
   Loopback status:             no loopback
   PDU revision:            1
Remote client:
---------------
 MAC address: e6c2.47f6.7809
 PDU revision: 1
 Vendor(oui): e6 c2 47
 Administrative configurations:
   Mode:                  active
   Unidirection:            not supported
   Link monitor:            supported
   Remote Loopback:             not supported
   MIB retrieval:           not supported
   MTU Size   :             1518
```

Display results on Switch2:

```
Switch# show ethernet oam discovery interface eth-0-9
eth-0-9
Local client:
-------------
 Administrative configurations:
   Mode:                  active
   Unidirection:            not supported
   Link monitor:            supported(on)
   Remote Loopback:             not supported
   MIB retrieval:           not supported
   MTU Size   :             1518
 Operational status:
   Port status:             operational
   Loopback status:             no loopback
   PDU revision:            1
Remote client:
---------------
 MAC address: 409c.ba1a.5a09
 PDU revision: 1
```

```
Vendor(oui): 40 9c ba
Administrative configurations:
  Mode:                  active
  Unidirection:          not supported
  Link monitor:          supported
  Remote Loopback:       not supported
  MIB retrieval:         not supported
  MTU Size   :           1518
```

Configuring Remote Loopback

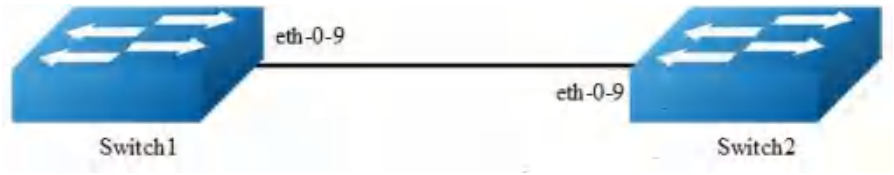


Figure 17-2 EFM

OAM remote loopback can be used for fault localization and link performance testing. In addition, an implementation may analyze loopback frames within the OAM sublayer to determine additional information about the health of the link (i.e. determine which frames are being dropped due to link errors).

The following configurations are same on Switch1 and Switch2 if there is no special description.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and enable ethernet oam remote loopback

```
Switch(config)# interface eth-0-9
Switch(config-if)# ethernet oam enable
Switch(config-if)# ethernet oam remote loopback supported
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Start remote loopback

Configure on Switch1:

```
Switch# ethernet oam remote-loopback start interface eth-0-9
```

step 5 Validation

Display results on Switch1:

```
Switch# show ethernet oam state-machine interface eth-0-9
State Machine Details:
---------------------------------
Local OAM mode:              Active
Local OAM enable:            Enable
Local link status:           OK
Local pdu status:            ANY
Local Satisfied:             True
Local Stable:                True
Remote Satisfied valid:      True
```

```
Remote Stable:                True
Local Parser State:           Discard
Local Multiplexer State:      Forward
Remote Parser State:          Loopback
Remote Multiplexer State:     Discard
```

Configuring Link Monitoring Event

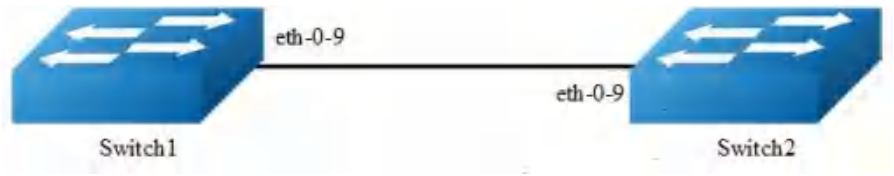


Figure 17-3 EFM

We can configure high and low threshold for link-monitoring features. We can also configure an error disable action if one of the high thresholds is exceeded.

The following configurations and validations are operated on Switch1:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the threshold for error packetes

```
Switch(config)#interface eth-0-9
Switch(config-if)# ethernet oam link-monitor frame threshold high 5000 low 200 window 500
Switch(config-if)# ethernet oam link-monitor frame-seconds threshold high 600 low 200 window 9000
```



The "ethernet oam link-monitor frame threshold" command specifies the high and low thresholds of error packets in a period. The period is defined by arguments "window 500", the unit is 100 millisecond, the default value is 1 second. In this case the high threshold is 5000 packets and the low threshold is 200 packets.

The "ethernet oam link-monitor frame-seconds threshold" command specifies the high and low thresholds of the seconds which have error packets in a period. The period is defined by arguments "window 9000", the unit is 100 millisecond, the default value is 100 second. In this case the high threshold is 600 seconds and the low threshold is 200 seconds.

step 3 Set the action when reach the threshold

When the error packets exceed the threshold configured in step 2, set the interface status to error-disable

```
Switch(config-if)# ethernet oam link-monitor high-threshold action error-disable-interface
Switch(config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch#show ethernet oam status interface eth-0-9
eth-0-9
General:
-------
  Mode:                  active
  PDU max rate:          1 packets per second
  PDU min rate:          1 packet per 1 second
  Link timeout:          10 seconds
```

```
 High threshold action:            disable interface
 Link fault action:                no action
 Dying gasp action:                no action
 Critical event action:            no action
Link Monitoring:
----------------
  Status:                    supported(on)
  Frame Error:
    Window:                  500 x 100 milliseconds
    Low threshold:           200 error frame(s)
    High threshold:          5000 error frame(s)
    Last Window Frame Errors:      0 Frame(s)
    Total Frame Errors:            0 Frame(s)
    Total Frame Errors Events:     0 Events(s)
    Relative Timestamp of the Event:  0 x 100 milliseconds
  Frame Seconds Error:
    Window:                  9000 x 100 milliseconds
    Low threshold:           200 error second(s)
    High threshold:          600 error second(s)
    Last Window Frame Second Errors:  0 error second(s)
    Total Frame Second Errors:     0 error second(s)
    Total Frame Second Errors Events: 0 Events(s)
    Relative Timestamp of the Event:  0 x 100 milliseconds
```

Configuring Remote Failure Detection

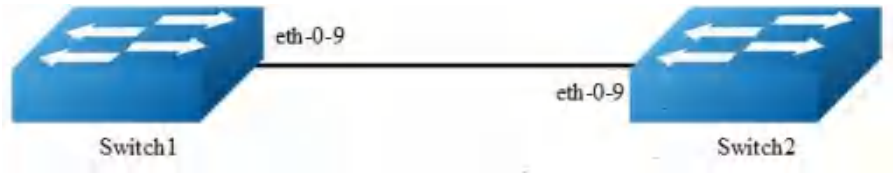


Figure 17-4 EFM

An error-disable action can be configured to occur on an interface so that if any of the critical link events (link fault, dying gasp, etc.) occurs in the remote machine, the interface is shut down.

The following configurations and validations are operated on Switch1:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set action when the remote link failure

```
Switch(config)#interface eth-0-9
Switch(config-if)# ethernet oam remote-failure critical-event dying-gasp link-fault action error-disable-interface
Switch(config-if)# exit
```

step 3 Exit the configure mode

```
Switch(config)# end
```

17.3 Configuring CFM

17.3.1 Overview

Function Introduction

CFM = Connectivity Fault Management

CFM provides the capability to detect, verify, isolate and notify connectivity failures on a Virtual Bridged LAN based on the protocol standard specified in IEEE 802.1ag. It provides for discovery and verification of paths through 802.1 bridges and LANs, and is part of the enhanced Operation, Administration and Management (OAM) features. CFM is designed to be transparent to the customer data transported by a network and to be capable of providing maximum fault coverage.

Principle Description

Reference: IEEE 802.1ag/D8.1

CFM uses standard Ethernet frames distinguished by EtherType. These CFM messages are supported:

Continuity Check messages (CC)

Multicast heartbeat messages exchanged periodically between MEPs that allow MEPs to discover other MEPs within a domain and allow MIPs to discover MEPs. It is used to detect loss of continuity (LOC) between any pair of MEPs.

Loopback messages

Unicast frames transmitted by an MEP at administrator request to verify connectivity to a particular maintenance point, indicating if a destination is reachable. A loopback message is similar to an Internet Control Message Protocol (ICMP) ping message.

Linktrace messages

Multicast frames transmitted by an MEP at administrator request to track the path (hop-by-hop) to a destination MEP/MIP. Traceroute messages are similar in concept to UDP traceroute messages.

Delay Measurement messages

A MEP sends DMM with ETH-DM request information to its peer MEP and receives DMR with ETH-DM reply information from its peer MEP to carry out two-way frame delay and delay variation measurements.

When a MEP receives 1DM frames, it will carry out one-way frame delay and delay variation measurements.

Ethernet Locked Signal messages

Ethernet Locked Signal function (ETH-LCK) is used to communicate the administrative locking of a server (sub) layer MEP and consequential interruption of data traffic forwarding towards the MEP expecting this traffic. It allows a MEP receiving frames with ETH-LCK information to differentiate between a defect condition and an administrative locking action at the server (sub) layer MEP.

Ethernet client signal fail messages

The Ethernet client signal fail function (ETH-CSF) is used by a MEP to propagate to a peer MEP the detection of a failure or defect event in an Ethernet client signal when the client itself does not support appropriate fault or defect detection or propagation mechanisms, such as ETH-CC or ETH-AIS. The ETH-CSF messages propagate in the direction from the Ethernet source-adaptation function detecting the failure or defect event to the Ethernet sink-adaptation function associated with the peer MEP. ETH-CSF is only applicable to point-to-point Ethernet transport applications.

Ethernet Frame loss measurement message

ETH-LM is used to collect counter values applicable for ingress and egress service frames where the counters maintain a count of transmitted and received data frames between a pair of MEPs.

ETH-LM is performed by sending LMM with ETH-LM information to a peer MEP and similarly receiving LMR with ETH-LM information from the peer MEP.

17.3.2 Configuration


CFM is conflict with 802.1x and mirror destination on the same port. Therefore, CFM and these functions should not be configured on the same port.

Configure CC/LB/LT/AIS/DM

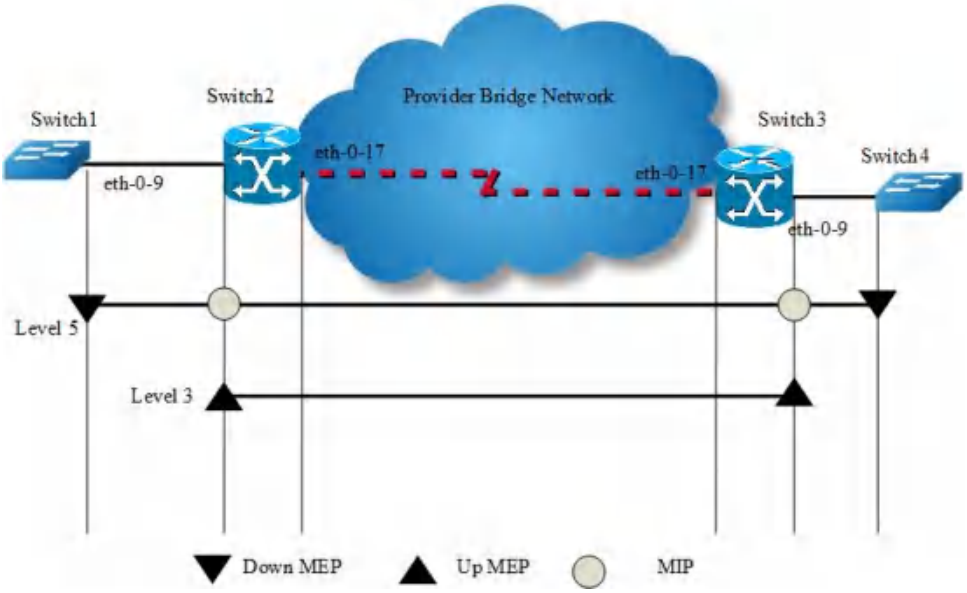


Figure 17-5 CFM

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config vlan)# vlan 30
Switch(config vlan)# exit
```

step 3 Enable CFM globally and set cfm mode to y1731

```
Switch(config)# ethernet cfm enable
Switch(config)# ethernet cfm mode y1731
```

step 4 Create the cfm domain and bind the service with a vlan

Create a domain which has the name "cust" and level 5.

```
Switch(config)# ethernet cfm domain cust level 5
Switch(config-ether-cfm)# service cst vlan 30
Switch(config-ether-cfm)# exit
```

Create a domain which has the name "provid" and level 3.

Configuring Switch2 and Switch3:

```
Switch(config)# ethernet cfm domain provid level 3
Switch(config-ether-cfm)# service cst vlan 30
Switch(config-ether-cfm)# exit
```


The range of the cfm domain level should be 0-7. The larger number indicates the higher priority. When different cfm domains have the same vlan, the packets of the domain with higher priority can pass through the domains with lower priority.

step 5 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mep down mpid 66 domain cust vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 99 domain cust vlan 30 mac d036.4567.8009
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mip level 5 vlan 30
Switch(config-if)# ethernet cfm mep up mpid 666 domain provid vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 999 domain provid vlan 30 mac 6a08.051e.bd09
Switch(config-if)# ethernet cfm ais status enable all domain provid vlan 30 level 5 multicast
Switch(config-if)# ethernet cfm server-ais status enable level 5 interval 1
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mip level 5 vlan 30
Switch(config-if)# ethernet cfm mep up mpid 999 domain provid vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 666 domain provid vlan 30 mac 0e1d.a7d7.fb09
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch4:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mep down mpid 99 domain cust vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 66 domain cust vlan 30 mac fa02.cdff.6a09
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 6 Enable continuity check

Configuring Switch1 and Switch4:

```
Switch1(config)# ethernet cfm cc enable domain cust vlan 30
```

Configuring Switch2 and Switch3:

```
Switch2(config)# ethernet cfm cc enable domain provid vlan 30
```

step 7 Alarm configuration (optional)

Suppress errors when ais packet is received and loc error.

Configuring Switch1:

```
Switch(config)# ethernet cfm ais suppress alarm enable domain cust vlan 30
```

step 8 Exit the configure mode

```
Switch(config)# end
```

step 9 Validation

MEP and MIP checks

The following command gives the connectivity details of the local machine Switch1 and Switch2 for the configured domain.

Switch1:

```
Switch# show ethernet cfm maintenance-points
######Local MEP:
MPID Direction DOMAIN LEVEL TYPE VLAN PORT    CC-Status Mac-address    RDI  Interval
--------------------------------------------------------------------------------
66  Down MEP  cust  5    MEP  30  eth-0-9  enabled  fa02.cdff.6a09 True 3.33ms
######Local MIP:
Level  VID  TYPE   PORT        MAC
------------------------------------------------------------
######Remote MEP:
MPID  LEVEL VLAN ACTIVE Remote Mac    RDI  FLAGS   STATE
------------------------------------------------------------
99  5    30  Yes   d036.4567.8009 True  Learnt   UP
```

Switch2:

```
Switch# show ethernet cfm maintenance-points
######Local MEP:
MPID Direction DOMAIN LEVEL TYPE VLAN PORT    CC-Status Mac-address    RDI
--------------------------------------------------------------------------------
666  Up MEP    provid 3    MEP  30   eth-0-9  enabled  0e1d.a7d7.fb09 False
######Local MIP:
Level  VID  TYPE   PORT        MAC
------------------------------------------------------------
5     30   MIP    eth-0-9     0e1d.a7d7.fb09
######Remote MEP:
MPID  LEVEL VLAN ACTIVE Remote Mac    RDI  FLAGS   STATE
------------------------------------------------------------
999  3    30   Yes   6a08.051e.bd09  True  Learnt   UP
```

Loopback checks

The following command is used to ping remote mep by remote mep unicast mac address on Switch1.

```
Switch# ethernet cfm loopback mac d036.4567.8009 unicast mepid 66 domain cust vlan 30
Sending 1 Ethernet CFM loopback messages, timeout is 5 seconds:
(! Pass . Fail)
!
```

```
 Loopback completed.
 ---------------------------------------
Success rate is 100 percent(1/1)
```

The following command is used to ping remote mep by multicast mac address on Switch1.

```
Switch# ethernet cfm loopback multicast mepid 66 domain cust vlan 30
Sending 1 Ethernet CFM loopback messages, timeout is 5 seconds:
(! Pass . Fail)
Host MEP: 66
Number of RMEPs that replied to mcast frame = 1
LBR received from the following
 9667.bb68.f308
 success rate is  100 (1/1)
```

The following command is used to ping remote mep by remote mep id on Switch1.

```
Switch# ethernet cfm loopback unicast rmepid 99 mepid 66 domain cust vlan 30
Sending 1 Ethernet CFM loopback messages, timeout is 5 seconds:
(! Pass . Fail)
!
 Loopback completed.
 ---------------------------------------
 Success rate is 100 percent(1/1)
```

The following command is used to ping mip by mip mac address on Switch1.

```
Switch# ethernet cfm loopback mac 0e1d.a7d7.fb09 unicast mepid 66 domain cust vlan 30
Sending 1 Ethernet CFM loopback messages, timeout is 5 seconds:
(! Pass . Fail)
!
 Loopback completed.
 ---------------------------------------
 Success rate is 100 percent(1/1)
```

RDI checks

Before clear local mep rdi, the rdi status on Switch1 is as follows:

```
Switch# show ethernet cfm maintenance-points local mep domain cust
MPID Direction DOMAIN LEVEL TYPE VLAN PORT    CC-Status Mac-address    RDI Interval
--------------------------------------------------------------------------------
66   Down MEP  cust   5  MEP  30   eth-0-9  enabled   fa02.cdff.6a09 True 3.33ms
```

ERROR checks

Before clear local mep errors, the errors on Switch1 are as follows:

```
Switch# show ethernet cfm errors domain cust
Level Vlan MPID RemoteMac      Reason                      ServiceId
5     30   66  d036.4567.8009 errorCCMdefect: rmep not found      cst
5     30   66  d036.4567.8009 errorCCMdefect: rmep not found clear cst
Time
2011/05/27 3:19:18
2011/05/27 3:19:32
```

The following command is used to clear errors on Switch1.

```
Switch# clear ethernet cfm errors domain cust
```

After clear local mep errors, the errors on Switch1 are as follows:

```
Switch# clear ethernet cfm errors domain cust
Level Vlan MPID RemoteMac     Reason                    ServiceId
```

AIS check

The following command is used to disable cc function in Switch1.

```
Switch(config)# no ethernet cfm cc enable domain cust vlan 30
```

The following command is used to disable cc function in Switch3.

```
Switch(config)# no ethernet cfm cc enable domain cust vlan 30
```

The following command is used to check ais defect condition in Switch2.

```
Switch# show ethernet cfm ais mep 666 domain cust vlan 30
AIS-Status: Enabled
AIS Period: 1
Level to transmit AIS: 7
AIS Condition: No
-------------------------------------------------------
 Configured defect condition    detected(yes/no)
-------------------------------------------------------
 unexpected-period             no
 unexpected-MEG level          no
 unexpected-MEP                no
 Mismerge                      no
 LOC                       yes
```

The following command is used to check ais reception status in Switch1.

```
Switch# show ethernet cfm ais mep 66 domain cust vlan 30
AIS-Status: Disabled
AIS Condition: Yes
```

LinkTrace checks

The following command is used to link trace remote mep by remote mep unicast mac address on Switch1.

```
Switch# ethernet cfm linktrace mac d036.4567.8009 mepid 66 domain cust vlan 30
Sending Ethernet CFM linktrace messages,TTL is 64.Per-Hop Timeout is 5 seconds:
Please wait a moment
---------------------------------
Received Hops: 1
---------------------------------
TTL                 : 63
Fowarded            : True
Terminal MEP        : False
Relay Action        : Rly FDB
Ingress Action      : IngOk
Ingress MAC address     : 0e1d.a7d7.fb09
Ingress Port ID Type    : ifName
Ingress Port ID         : eth-0-9
---------------------------------
Received Hops: 2
---------------------------------
TTL                 : 62
Fowarded            : True
Terminal MEP        : False
Relay Action        : Rly FDB
Egress Action       : EgrOk
```

```
Egress MAC address       : 6a08.051e.bd09
Egress Port ID Type      : ifName
Egress Port ID           : eth-0-9
---------------------------------
Received Hops: 3
---------------------------------
TTL                  : 61
Fowarded             : False
Terminal MEP         : True
Relay Action         : Rly Hit
Ingress Action       : IngOk
Ingress MAC address      : d036.4567.8009
Ingress Port ID Type     : ifName
Ingress Port ID          : eth-0-9
```

The following command is used to link trace remote mep by remote mep id on Switch1.

```
Switch# ethernet cfm linktrace rmepid 99 mepid 66 domain cust vlan 30
Sending Ethernet CFM linktrace messages,TTL is 64.Per-Hop Timeout is 5 seconds:
Please wait a moment
---------------------------------
Received Hops: 1
---------------------------------
TTL                  : 63
Fowarded             : True
Terminal MEP         : False
Relay Action         : Rly FDB
Ingress Action       : IngOk
Ingress MAC address      : 0e1d.a7d7.fb09
Ingress Port ID Type     : ifName
Ingress Port ID      : eth-0-9
---------------------------------
Received Hops: 2
---------------------------------
TTL                  : 62
Fowarded             : True
Terminal MEP         : False
Relay Action         : Rly FDB
Egress Action        : EgrOk
Egress MAC address       : 6a08.051e.bd09
Egress Port ID Type      : ifName
Egress Port ID           : eth-0-9
---------------------------------
Received Hops: 3
---------------------------------
TTL                  : 61
Fowarded             : False
Terminal MEP         : True
Relay Action         : Rly Hit
Ingress Action       : IngOk
Ingress MAC address      : d036.4567.8009
Ingress Port ID Type     : ifName
Ingress Port ID          : eth-0-9
```

The following command is used to link trace remote mip by remote mip unicast mac address on Switch1.

```
Switch# ethernet cfm linktrace 6a08.051e.bd09 mepid 66 domain cust vlan 30
Sending Ethernet CFM linktrace messages,TTL is 64.Per-Hop Timeout is 5 seconds:
Please wait a moment
---------------------------------
Received Hops: 1
---------------------------------
TTL                  : 63
Fowarded             : True
Terminal MEP         : False
Relay Action         : Rly FDB
Ingress Action       : IngOk
Ingress MAC address  : 0e1d.a7d7.fb09
Ingress Port ID Type : ifName
Ingress Port ID      : eth-0-9
---------------------------------
Received Hops: 2
---------------------------------
TTL                  : 62
Fowarded             : False
Terminal MEP         : False
Relay Action         : Rly Hit
Egress Action        : EgrOk
Egress MAC address   : 6a08.051e.bd09
Egress Port ID Type  : ifName
Egress Port ID       : eth-0-9
```

1DM and DMM checks

The following command is used to make two way delay and delay variation measurement on Switch1.

```
Switch# ethernet cfm dmm  rmepid 99 mepid 66 count 5 domain cust vlan 30
  Delay measurement statistics:
   DMM Packets transmitted      : 5
   Valid DMR packets received   : 5
   Index    Two-way delay    Two-way delay variation
     1       4288 usec              0 usec
     2       4312 usec             24 usec
     3       4296 usec             16 usec
     4       4320 usec             24 usec
     5       4264 usec             56 usec
   Average delay             : 4296 usec
   Average delay variation   : 24 usec
   Best case delay           : 4264 usec
   Worst case delay          : 4320 usec
```

Before make one way delay measurement, clock timer should be synchronized. The following command is used to start sending 1dm message in Switch1.

```
Switch1#ethernet cfm 1dm rmepid 99 mepid 66 count 5 domain cust vlan 30
```

The following is 1dm test result in Switch4.

```
Switch4# show ethernet cfm delaymeasurement cache
  Remote MEP     : 66
   Remote MEP vlan  : 30
   Remote MEP level : 5
   DMM Packets transmitted      : 0
   Valid DMR packets received   : 0
   Valid 1DM packets received   : 5
```

```
Index   One-way delay   One-way delay variation   Received Time
  1     16832 usec              0 usec   2011/07/19 17:27:46
  2     16176 usec            656 usec   2011/07/19 17:27:47
  3     15448 usec            728 usec   2011/07/19 17:27:48
  4     14800 usec            648 usec   2011/07/19 17:27:49
  5     15406 usec            606 usec   2011/07/19 17:27:50
Average delay             : 15732 usec
Average delay variation   : 527 usec
Best case delay           : 14800 usec
Worst case delay          : 16832 usec
```

Configure LCK

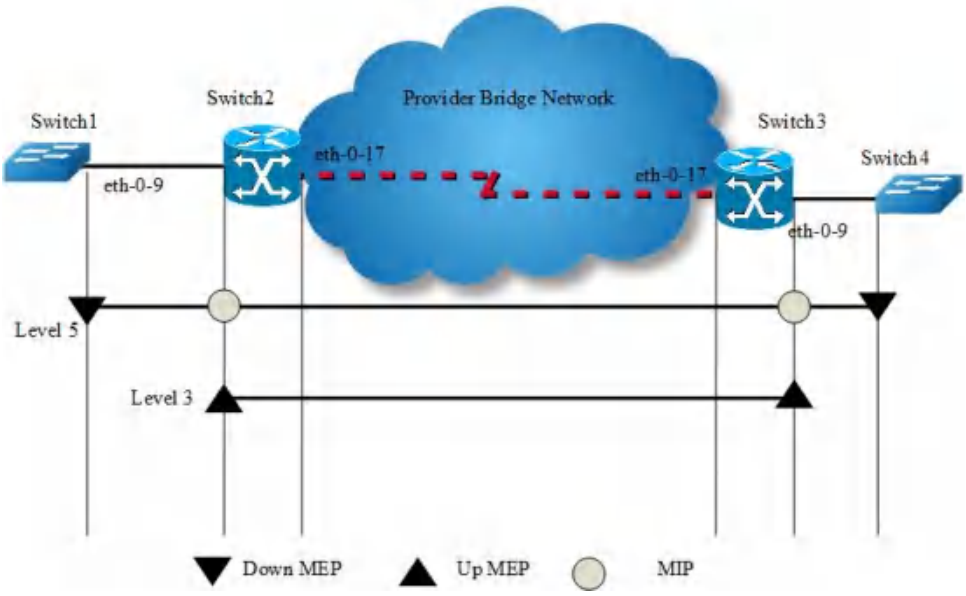


Figure 17-6 CFM

step 1 Configuration prepare

Reference to the chapter "Configure CC/LB/LT/AIS/DM".

step 2 Configure LCK

Configuring Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# ethernet cfm lck enable mep 666 domain provid vlan 30 tx-level 5 interval 1
```

step 3 Validation

The following command is used to display lck status for Switch2:

```
Switch2# show ethernet cfm lck
En-LCK Enable, Y(Yes)/N(No)
Rx-LC, Receive LCK packets and enter LCK condition, Y(Yes)/N(No)
Rx-I, The period which is gotten from LCK packets
Tx-Domain, frames with ETH-LCK information are sent to this Domain
Tx-I, Transmit Interval
--------------------------------------------------------------------------
MPID Domain      VLAN En Rx-LC Rx-I Tx-Domain    Tx-I
```

```
--------------------------------------------------------------------------
666 provid     30  Y N    N/A cust       1
```

The following command is used to display lck status for Switch1:

```
Switch1# show ethernet cfm lck
En-LCK Enable, Y(Yes)/N(No)
Rx-LC, Receive LCK packets and enter LCK condition, Y(Yes)/N(No)
Rx-I, The period which is gotten from LCK packets
Tx-Domain, frames with ETH-LCK information are sent to this Domain
Tx-I, Transmit Interval
--------------------------------------------------------------------------
MPID Domain      VLAN En Rx-LC Rx-I Tx-Domain    Tx-I
--------------------------------------------------------------------------
66  cust       30  N Y    1   N/A        N/A
```

Configure CSF

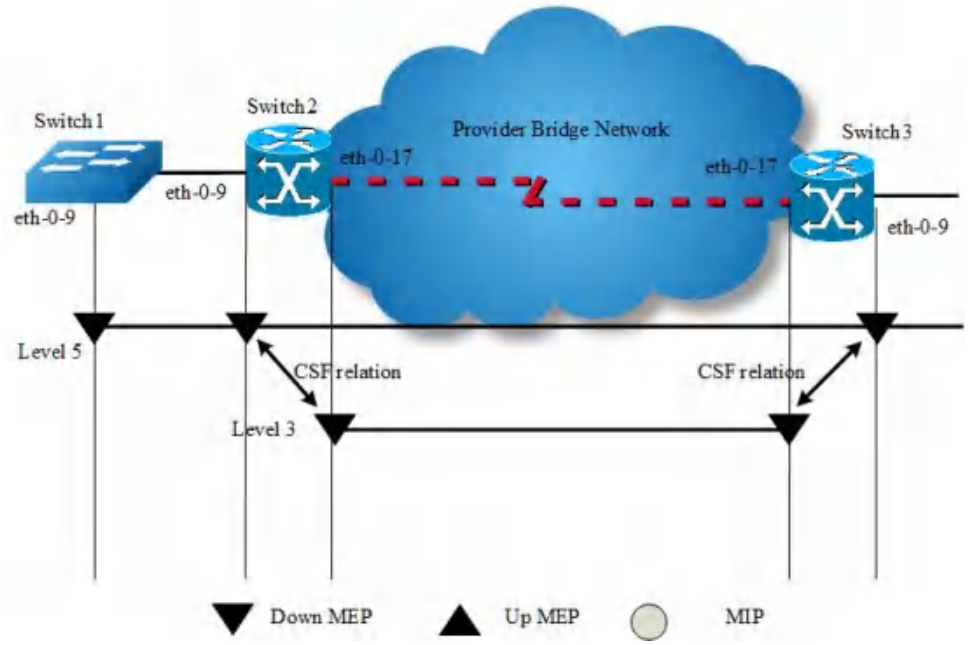


Figure 17-7 CFM CSF

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

Configuring Switch1:

```
Switch(config)# vlan database
Switch(config vlan)# vlan 30
Switch(config vlan)# exit
```

Configuring Switch2 and Switch3:

```
Switch3(config)# vlan database
Switch3(config vlan)# vlan 20,30
Switch3(config vlan)# exit
```

step 3 Enable CFM globally and set cfm mode to y1731

```
Switch(config)# ethernet cfm enable
Switch(config)# ethernet cfm mode y1731
```

step 4 Create the cfm domain and bind the service with a vlan

Create a domain which has the name "cust" and level 5.

```
Switch(config)# ethernet cfm domain cust level 5
Switch(config-ether-cfm)# service cst vlan 30
Switch(config-ether-cfm)# exit
```

Create a domain which has the name "provid" and level 3.

Configuring Switch2 and Switch3:

```
Switch(config)# ethernet cfm domain provid level 3
Switch(config-ether-cfm)# service cst vlan 20
Switch(config-ether-cfm)# exit
```

step 5 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mep down mpid 66 domain cust vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 99 domain cust vlan 30 mac d036.4567.8009
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mep down mpid 99 domain cust vlan 30 interval 1
Switch(config-if)# ethernet cfm mep crosscheck mpid 66 domain cust vlan 30 mac fa02.cdff.6a09
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# ethernet cfm mep down mpid 666 domain provid vlan 20 interval 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 30
Switch(config-if)# ethernet cfm mep down mpid 88 domain cust vlan 30 interval 1
Switch(config-if)# no shutdown
```

```
Switch(config-if)# exit

Switch(config)#interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# ethernet cfm mep down mpid 999 domain provid vlan 20 interval 1
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 6 Enable continuity check

```
Switch(config)# ethernet cfm cc enable domain cust vlan 30
```

step 7 Configure csf relation between client mep and server mep

Configuring Switch2:

```
Switch(config)# ethernet cfm csf client domain cust vlan 30 mepid 99 server domain provid vlan 20 mepid 666 interval 1
```

Configuring Switch3:

```
Switch(config)# ethernet cfm csf client domain cust vlan 30 mepid 88 server domain provid vlan 20 mepid 999 interval 1
```

step 8 Validation

The following command is used to disable cc function in Switch1.

```
Switch (config)#no ethernet cfm cc enable domain cust vlan 30
```

For Switch2, client MEP 99 will report loc error and trigger csf for reason los, therefore server MEP 666 will send CSF packet in interval 1 second. The following command is used to display csf status for Swtich2.

```
Switch# show ethernet cfm csf
CTR-Client Trigger reason, L(los)/F(fdi)/R(rdi)/D(dci) or N/A
ECC-Enter CSF Condition, Y(Yes)/N(No)
SRR-Server Rx Reason, L(los)/F(fdi)/R(rdi)/D(dci) or N/A
Tx-I, Transmit Interval
Rx-I, The period which is gotten from CSF packets
--------------------------------------------------------------------------
Client Mep                  Server Mep
MPID Cli-Domain   VLAN CTR   ECC MPID Srv-Domain   VLAN SRR   Tx-I Rx-I
--------------------------------------------------------------------------
99  cust      30  L    N  666  provid     20  N/A  1   N/A
```

For Switch3, server MEP 999 receives CSF packet and informs client MEP 99, then client MEP 88 will enter CSF condition. The following command is used to display csf status for Switch3:

```
Switch3# show ethernet cfm csf
CTR-Client Trigger reason, L(los)/F(fdi)/R(rdi)/D(dci) or N/A
ECC-Enter CSF Condition, Y(Yes)/N(No)
SRR-Server Rx Reason, L(los)/F(fdi)/R(rdi)/D(dci) or N/A
Tx-I, Transmit Interval
Rx-I, The period which is gotten from CSF packets
--------------------------------------------------------------------------
Client Mep                  Server Mep
MPID Cli-Domain   VLAN CTR   ECC MPID Srv-Domain   VLAN SRR   Tx-I Rx-I
--------------------------------------------------------------------------
88  cust      30  N/A  Y  999  provid     20  L    1   1
```

Configure Dual-Ended LM

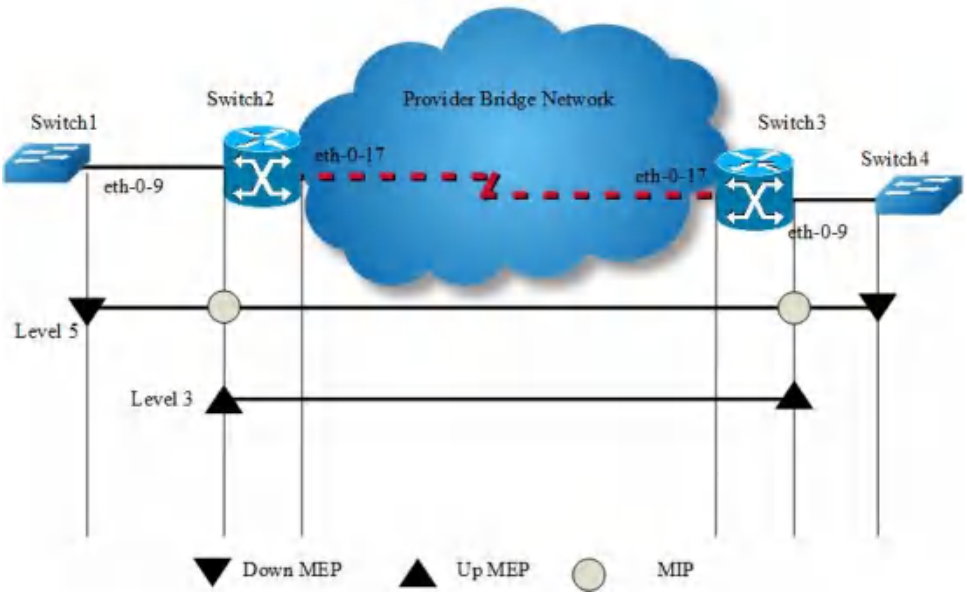


Figure 17-8 CFM

step 1 Configuration prepare

Reference to the chapter "Configure CC/LB/LT/AIS/DM".

step 2 Configure Dual-Ended LM

Configuring Switch1:

```
Switch(config)# ethernet cfm lm enable dual-ended domain cust vlan 30 mepid 66 all-cos cache-size 10
```

Configuring Switch4:

```
Switch(config)# ethernet cfm lm enable dual-ended domain cust vlan 30 mepid 99 all-cos cache-size 10
```

step 3 Validation

The following command is used to display lm status for Switch1.

```
Switch# show ethernet cfm lm domain cust vlan 30 mepid 66
DOMAIN    : cust
VLAN      : 30
MEPID     : 66
Start Time : 2013/07/16 1:36:56
End Time   : 2013/07/16 1:37:07
Notes     : 1. When the difference of Tx is less than the difference of Rx,
              the node is invalid, loss and loss ratio should be "-";
            2. When loc is reported for mep, the loss should be "-" and loss
              ratio should be 100%;
            3. When calculate average loss and loss ratio, invalid or loc nodes
              will be excluded;
Latest dual-ended loss statistics:
--------------------------------------------------------------------------------
Index Cos Local-loss Local-loss ratio Remote-loss Remote-loss ratio Time
--------------------------------------------------------------------------------
1    all      0      000.0000%       0      000.0000% 01:36:57
2    all      0      000.0000%       0      000.0000% 01:36:58
3    all      0      000.0000%       0      000.0000% 01:36:59
```

```
4    all       0      000.0000%        0      000.0000% 01:37:00
5    all       0      000.0000%        0      000.0000% 01:37:01
6    all       0      000.0000%        0      000.0000% 01:37:02
7    all       0      000.0000%        0      000.0000% 01:37:03
8    all       0      000.0000%        0      000.0000% 01:37:04
9    all       0      000.0000%        0      000.0000% 01:37:05
10   all       0      000.0000%        0      000.0000% 01:37:07
---------------------------------------------------------------------------
Maximum Local-loss  : 0        Maximum Local-loss Ratio  : 000.0000%
Minimum Local-loss  : 0        Minimum Local-loss Ratio  : 000.0000%
Average Local-loss  : 0        Average Local-loss Ratio  : 000.0000%
Maximum Remote-loss : 0        Maximum Remote-loss Ratio : 000.0000%
Minimum Remote-loss : 0        Minimum Remote-loss Ratio : 000.0000%
Average Remote-loss : 0        Average Remote-loss Ratio : 000.0000%
```

The following command is used to display lm status for Switch4.

```
Switch# show ethernet cfm lm domain cust vlan 30 mepid 99
DOMAIN    : cust
VLAN      : 30
MEPID     : 99
Start Time : 2013/07/16 1:37:11
End Time   : 2013/07/16 1:37:22
Notes     : 1. When the difference of Tx is less than the difference of Rx,
              the node is invalid, loss and loss ratio should be "-";
            2. When loc is reported for mep, the loss should be "-" and loss
              ratio should be 100%;
            3. When calculate average loss and loss ratio, invalid or loc nodes
              will be excluded;
Latest dual-ended loss statistics:
---------------------------------------------------------------------------
Index Cos Local-loss Local-loss ratio Remote-loss Remote-loss ratio Time
---------------------------------------------------------------------------
1    all       0      000.0000%        0      000.0000% 01:37:12
2    all       0      000.0000%        0      000.0000% 01:37:13
3    all       0      000.0000%        0      000.0000% 01:37:14
4    all       0      000.0000%        0      000.0000% 01:37:16
5    all       0      000.0000%        0      000.0000% 01:37:17
6    all       0      000.0000%        0      000.0000% 01:37:18
7    all       0      000.0000%        0      000.0000% 01:37:19
8    all       0      000.0000%        0      000.0000% 01:37:20
9    all       0      000.0000%        0      000.0000% 01:37:21
10   all       0      000.0000%        0      000.0000% 01:37:22
---------------------------------------------------------------------------
Maximum Local-loss  : 0        Maximum Local-loss Ratio  : 000.0000%
Minimum Local-loss  : 0        Minimum Local-loss Ratio  : 000.0000%
Average Local-loss  : 0        Average Local-loss Ratio  : 000.0000%
Maximum Remote-loss : 0        Maximum Remote-loss Ratio : 000.0000%
Minimum Remote-loss : 0        Minimum Remote-loss Ratio : 000.0000%
Average Remote-loss : 0        Average Remote-loss Ratio : 000.0000%
```

Configure Single-Ended LM

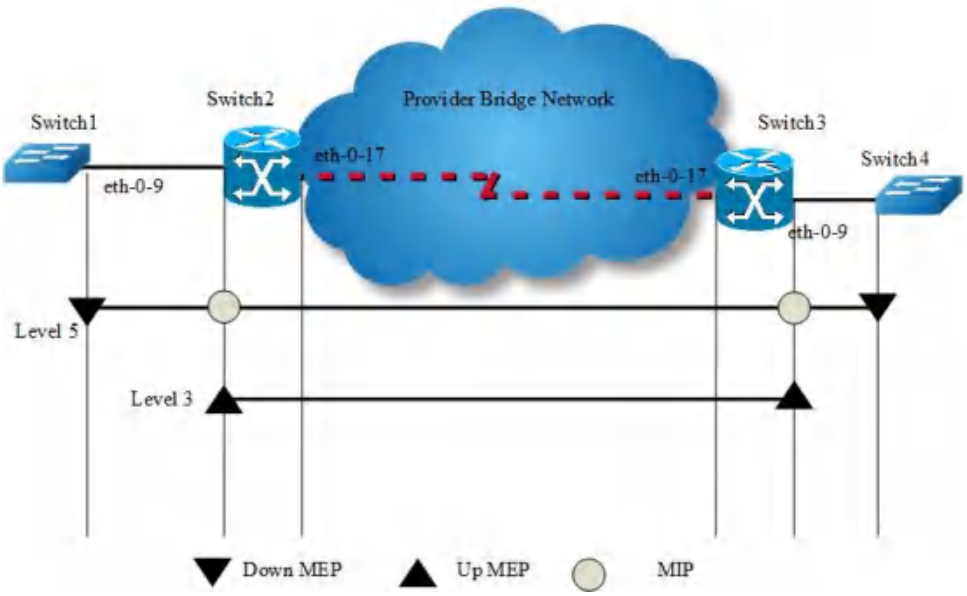


Figure 17-9 CFM

step 1 Configuration prepare

Reference to the chapter "Configure CC/LB/LT/AIS/DM".

step 2 Configure Single-Ended LM

Configuring Switch1:

```
Switch(config)# ethernet cfm lm enable single-ended domain cust vlan 30 mepid 66 all-cos
```

Configuring Switch4:

```
Switch(config)# ethernet cfm lm enable single-ended domain cust vlan 30 mepid 99 all-cos
```

step 3 Validation

The following command is used to output lmm and display lm results for Switch1.

```
Switch#  ethernet cfm lm single-ended domain cust vlan 30 rmepid 99 mepid 66 count 10
DOMAIN    : cust
VLAN      : 30
MEPID     : 66
Start Time : 2013/07/16 1:39:38
End Time   : 2013/07/16 1:39:38
Notes     : 1. When the difference of Tx is less than the difference of Rx,
              the node is invalid, loss and loss ratio should be "-";
            2. When loc is reported for mep, the loss should be "-" and loss
              ratio should be 100%;
            3. When calculate average loss and loss ratio, invalid or loc nodes
              will be excluded;
Latest single-ended loss statistics:
--------------------------------------------------------------------------------
Index Cos Local-loss Local-loss ratio Remote-loss Remote-loss ratio
--------------------------------------------------------------------------------
1    all      0     000.0000%       0     000.0000%
2    all      0     000.0000%       0     000.0000%
3    all      0     000.0000%       0     000.0000%
```

```
4   all     0     000.0000%       0     000.0000%
5   all     0     000.0000%       0     000.0000%
6   all     0     000.0000%       0     000.0000%
7   all     0     000.0000%       0     000.0000%
8   all     0     000.0000%       0     000.0000%
9   all     0     000.0000%       0     000.0000%
---------------------------------------------------------------------------
Maximum Local-loss  : 0       Maximum Local-loss Ratio  : 000.0000%
Minimum Local-loss  : 0       Minimum Local-loss Ratio  : 000.0000%
Average Local-loss  : 0       Average Local-loss Ratio  : 000.0000%
Maximum Remote-loss : 0       Maximum Remote-loss Ratio : 000.0000%
Minimum Remote-loss : 0       Minimum Remote-loss Ratio : 000.0000%
Average Remote-loss : 0       Average Remote-loss Ratio : 000.0000%
```

Configure Test

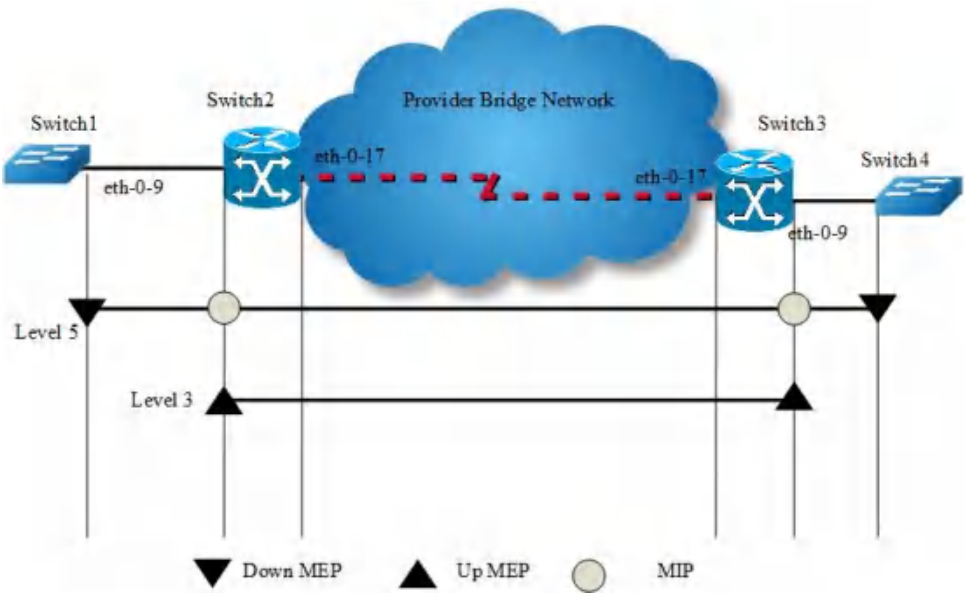


Figure 17-10 CFM

step 1 Configuration prepare

Reference to the chapter "Configure CC/LB/LT/AIS/DM".

step 2 Configure Test

Configure test transmission enable on Switch1:

```
Switch(config)# ethernet cfm tst transmission enable domain cust vlan 30 mep 66 tx-mode continuous pattern-type random packet-size 6
```

Configure test reception enable on Switch4:

```
Switch(config)# ethernet cfm tst reception enable domain cust vlan 30 mep 99
```

step 3 Validation

The following command is used to start test transmission on Switch1.

```
Switch#  ethernet cfm tst start rate 1000 time second 1
```

The following command is used to display test information on Switch1.

```
Switch#  show ethernet cfm tst
DOMAIN        : cust
VLAN          : 30
MEPID         : 66
Transmission    : Enabled
Reception       : Disabled
Status        : Non-Running
Start Time      : 06:32:48
Predict End Time : 06:33:18
Actual End Time  : 06:33:18
Packet Type     : TST
Rate          : 1000 mbps
Packet Size     : 64 bytes
Tx Number       : 29
Tx Bytes        : 1856
Rx Number       : 0
Rx Bytes        : 0
```

The following command is used to display test information on Switch4.

```
Switch#  show ethernet cfm tst
DOMAIN        : cust
VLAN          : 30
MEPID         : 99
Transmission    : Disabled
Reception       : Enabled
Status        : Non-Running
Start Time      : null
End Time        : null
Packet Type     : null
Rate          : null
Packet Size     : null
Tx Number       : 0
Tx Bytes        : 0
Rx Number       : 29
Rx Bytes        : 1856
```

17.4 Configuring CPU Traffic

17.4.1 Overview

Function Introduction

CPU traffic limit is a useful mechanism for protecting CPU from malicious flows by injecting huge volume of PDUs into switches.

CPU traffic limit provides two-level protection for CPU.

- The low-level traffic limit is performed for each reason, which is realized by queue shaping of each type of PDU.
- The high-level traffic limit is performed for all reasons, which is realized by channel shaping at CPU channel.

With this two-level protection, each PDU-to-CPU rate is limited and the overall PDU-to-CPU rate is also limited.


: The word “reason”, means this type of packets will be sent to cpu for further processing.

The description of all reason is as following.

Reason	Description

arp	Address Resolution Protocol
bpdu	Bridge Protocol Data Unit
dhcp	Dynamic Host Configuration Protocol
eapol	Extensible Authentication Protocol Over Lan
erps	Ethernet Ring Protection Switching
fwd-to-cpu	Packets forwarding to cpu
icmp-redirect	ICMP Redirect
igmp	IGMP Snooping Protocol
ip-option	Packets with IP Option
ipda	IP Destination to Router-self
ssh	SSH protocol packet
telnet	Telnet protocol packet
mlag	MLAG protocol packet
tcp	TCP protocol packet
ldp	Label Distribution Protocol
macsa-mismatch	Port Security for source mac learned
mcast-rpf-fail	Multicast with rpf fail or first multicast packet
mpls-ttl-fail	Mpls Packets with ttl fail
ip-mtu-fail	IP packet with mtu fail
ospf	Open Shortest Path First
pim	Protocol Independent Multicast
port-security-discard	Port Security for exceeding fdb maxnum
rip	Routing Information Protocol
sflow-egress	Sampled flow at egress direction
sflow-ingress	Sampled flow at ingress direction
slow-protocol	Slow Protocol (including EFM, LACP and SYNCE)
smart-link	Smart Link Protocol
ucast-ttl-fail	Unicast Packets with ttl fail
udld	Unidirectional Link Detection Protocol
vlan-security-discard	Vlan Security for exceeding fdb maxnum
vrrp	Virtual Router Redundancy Protocol
bfd-learning	BFD learning packets
dot1x-mac-bypass	Mac auth bypass packets
bgp	Border gateway protocol packet

egress-ttl-fail	Egress ttl fail packet
icmpv6	ICMPv6 packet
l2protocol-tunnel	Layer2 protocol tunnel packet
loopback-detection	lLoopback detection packet
mirror-to-cpu	Mirror to cpu packet
ndp	Neighbor discovery protocol packet
tunnel-gre-keepalive	Tunnel gre keepalive reply packet

The default rate and class configuration for all reason is as following.

reason	rate(pps)	class
arp	256	1
bpdu	64	3
dhcp	128	0
eapol	128	0
erps	128	3
fwd-to-cpu	64	0
icmp-redirect	128	0
igmp	128	2
ip-option	512	0
ipda	1000	0
ssh	64	3
telnet	64	3
mlag	1000	1
tcp	64	2
ldp	512	1
macsa-mismatch	128	0
mcast-rpf-fail	128	1
mpls-ttl-fail	64	0
ip-mtu-fail	64	0
ospf	256	1
pim	128	1
port-security-discard	128	0
rip	64	1
sflow-egress	128	0
sflow-ingress	128	0

slow-protocol	256	1
smart-link	128	2
ucast-ttl-fail	64	0
udld	128	3
vlan-security-discard	128	0
vrrp	512	1
bfd-learning	128	1
dot1x-mac-bypass	64	2
bgp	256	1
egress-ttl-fail	64	0
icmpv6	64	2
l2protocol-tunnel	1000	0
loopback-detection	64	3
mirror-to-cpu	1000	0
ndp	64	2
tunnel-gre-keepalive	64	0

Principle Description

Terminology

- PDU: Protocol Data Unit

17.4.2 Configuration

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the total rate

The default value of total rate is 2000, the unit is pps (packet-per-second)

```
Switch(config)# cpu-traffic-limit total rate 3000
```

step 3 Set the saparate rate

Use RIP packets for example:

```
Switch(config)# cpu-traffic-limit reason rip rate 500
```

step 4 Set the reason class

```
Switch(config)# cpu-traffic-limit reason rip class 3
```


The valid range of reason class is 0-3. The larger number indicates the higher priority.

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

To display the CPU Traffic Limit configuration, use following privileged EXEC commands.

```
Switch# show cpu traffic-limit
reason                rate (pps)   class
dot1x-mac-bypass      64           2
bpdu                  64           3
slow-protocol         256          1
eapol                 128          0
erps                  128          3
smart-link            128          2
udld                  128          3
loopback-detection    64           3
arp                   256          1
dhcp                  128          0
rip                   500          3
ldp                   512          1
ospf                  256          1
pim                   128          1
bgp                   256          1
vrrp                  512          1
ndp                   64           2
icmpv6                64           2
ssh                   64           3
telnet                64           3
mlag                  1000         1
tcp                   64           2
ipda                  1000         0
icmp-redirect         128          0
mcast-rpf-fail        128          1
macsa-mismatch        128          0
port-security-discard 128          0
vlan-security-discard 128          0
egress-ttl-fail       64           0
ip-mtu-fail           64           0
bfd-learning          128          1
ptp                   512          2
ip-option             512          0
tunnel-gre-keepalive  64           0
ucast-ttl-fail        64           0
mpls-ttl-fail         64           0
igmp                  128          2
sflow-ingress         128          0
sflow-egress          128          0
fwd-to-cpu            64           0
l2protocol-tunnel     1000         0
mirror-to-cpu         1000         0
Total rate:     3000 (pps)
```

To display the CPU Traffic statistics information, use following privileged EXEC commands.

```
Switch# show cpu traffic-statistics receive all
statistics rate time is 5 second(s)
reason        count(packets)   rate(pps)
dot1x-mac-bypass     0           0
bpdu                 0           0
```

```
slow-protocol        0           0
eapol                0           0
erps                 0           0
smart-link           0           0
udld                 0           0
loopback-detection   0           0
arp                  0           0
dhcp                 0           0
rip                  0           0
ldp                  0           0
ospf                 0           0
pim                  0           0
bgp                  0           0
vrrp                 0           0
rsvp                 0           0
ndp                  0           0
icmpv6               0           0
ssh                  0           0
telnet               0           0
mlag                 0           0
tcp                  0           0
ipda                 0           0
icmp-redirect        0           0
mcast-rpf-fail       0           0
macsa-mismatch       0           0
port-security-discard 0          0
vlan-security-discard 0          0
egress-ttl-fail      0           0
ip-mtu-fail          0           0
bfd-learning         0           0
ptp                  0           0
ip-option            0           0
tunnel-gre-keepalive 0           0
ucast-ttl-fail       0           0
mpls-ttl-fail        0           0
igmp                 0           0
sflow-ingress        0           0
sflow-egress         0           0
fwd-to-cpu           0           0
l2protocol-tunnel    0           0
mirror-to-cpu        0           0
mpls-tp-pwoam        0           0
other                0           0
Total                0           0
```

17.5 Configuring UDLD

17.5.1 Overview

Function Introduction

The Unidirectional Link Detection protocol is a light-weight protocol that can be used to detect and disable one-way connections before they create dangerous situations such as Spanning Tree loops or other protocol malfunctions.

Principle Description

N/A

17.5.2 Configuration

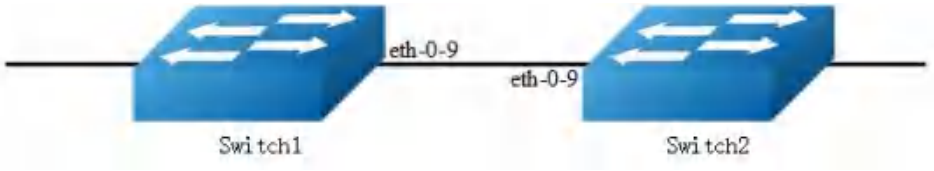


Figure 17-11 UDLD

The following configurations are same on Switch1 and Switch2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and enable udld

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# udld port
Switch(config-if)# exit
```

step 3 Enable udld globally

```
Switch(config)# udld enable
```

step 4 Set the message interval (optional)

If the message is not specified, use the default value: 15 seconds.

```
Switch(config)# udld message interval 10
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch# show udld eth-0-9
Interface eth-0-9
---
UDLD mode       : normal
Operation state  : Bidirectional
Message interval : 10
Message timeout   : 3
 Neighbor 1
 ---
 Device ID     : 4c7b.8510.ab00
 Port ID       : eth-0-9
 Device Name    : Switch
 Message interval: 10
 Message timeout : 3
 Link Status    : bidirectional
 Expiration time : 29
```

Display the result on Switch2.

```
Switch# show udld eth-0-9
Interface eth-0-9
---
UDLD mode      : normal
Operation state : Bidirectional
Message interval: 10
Message timeout : 3
 Neighbor 1
 ---
 Device ID     : 28bc.83db.8400
 Port ID       : eth-0-9
 Device Name    : Switch
 Message interval: 10
 Message timeout : 3
 Link Status    : bidirectional
 Expiration time : 23
```

17.6 Configuring ERPS

17.6.1 Overview

Function Introduction

ERPS technology increases the availability and robustness of Ethernet rings. In the event that a fiber cut occurs, ERPS converges in less than one second, often in less than 50 milliseconds.

The main idea is described as the following. ERPS operates by declaring an ERPS domain on a single ring. On that ring domain, one switch, or node, is designated the master node, while all other nodes are designated as transit nodes. One port of the master node is designated as the master node's primary port to the ring; another port is designated as the master node's secondary port to the ring. In normal operation, the master node blocks the secondary port for all non-ERPS traffic belonging to this ERPS domain, thereby avoiding a loop in the ring. Keep-alive messages are sent by the master node in a pre-set time interval. Transit nodes in the ring domain will forward the ERPS messages. Once a link failure event occurs, the master node will detect this either by receiving the link-down message sent by the node adjacent to the failed link or by the timeout of the keep-alive message. After link failure is detected, master node will open the secondary port for data traffic to re-route the traffic.

Background

In Layer 2 network, it usually use redundancy link to improve the reliability of network; redundancy link might bring ring circuits, it can use STP protocol to solve the problem of ring circuits. STP protocol which is a standard ring circuit protection protocol which is developed by IEEE and it is widely used; however STP is limited by the size of network and speed of convergence is affected by topology. The STP convergence time normally is in seconds, when the diameter of network is bigger, the convergence time is longer. Using RSTP/MSTP can reduce the convergence time but it cannot be satisfied the requirements from Voice/Video and other higher quality services. To reduce the convergence time and clean the influence of network size, ERPS protocol is created for it. ERPS is a protocol in link layer which is specialize apply to Ethernet loops, and it can prevent the broadcast which is caused by data loops; when one link is broken on Ethernet, it will use backup link to restore networking between each nodes quickly. Compared with STP protocol, ERPS protocol has quicker convergence speed of topology (lower than 50ms) and the feature that convergence time is unrelated than the amount of nodes on loops.

Principle Description

Reference: RFC 3619

Basic Conceptions

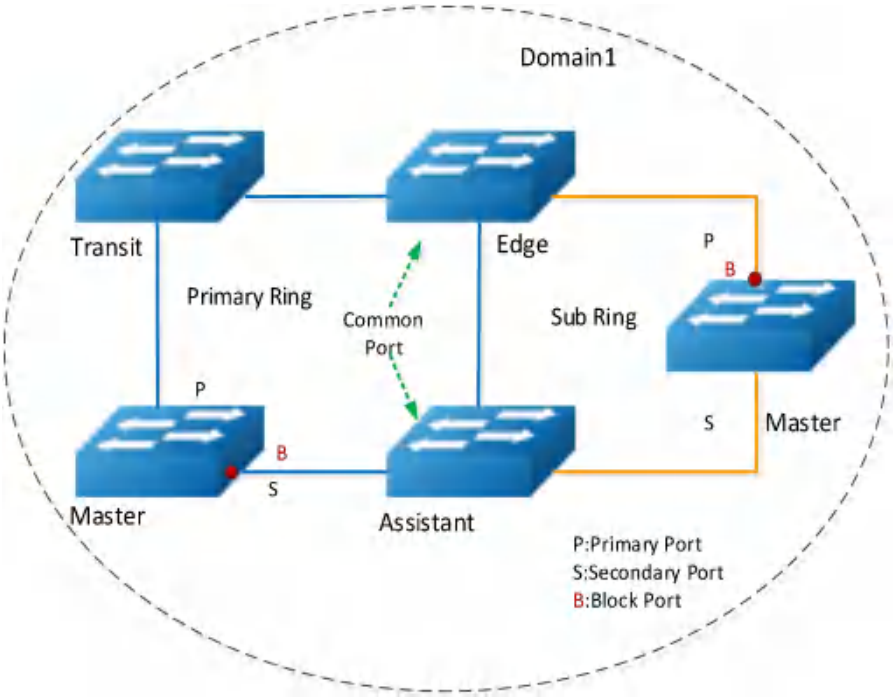


Figure 17-12 ERPS diagram

ERPS Domain:

ERPS domain is the topology range of ERPS protocol controlled. The domain ID is the unique identification of ERPS domain. The ERPS includes interconnected devices with same domain ID, control VLAN. One device can join more than one ERPS Domain.

ERPS rings

Each ERPS ring has a corresponding Ethernet topology of ring connection, which is marked by ERPS Ring ID. One ERPS domain contains one or several ERPS rings, one is primary ring and others are sub rings. In the condition of singe loop, it can configure ring as primary ring and also can configure as sub ring, it has the same affect on applications.

The type of ERPS ring:

- Single-Ring: All devices in the single-ring MUST in the same ERPS domain.
- Intersecting-Ring: All devices in the Intersecting-ring MUST in the same ERPS domain.
- Tangential-Ring: Tangential-Ring equals to two single-rings, need two domains and each ring is in its own ERPS domain.

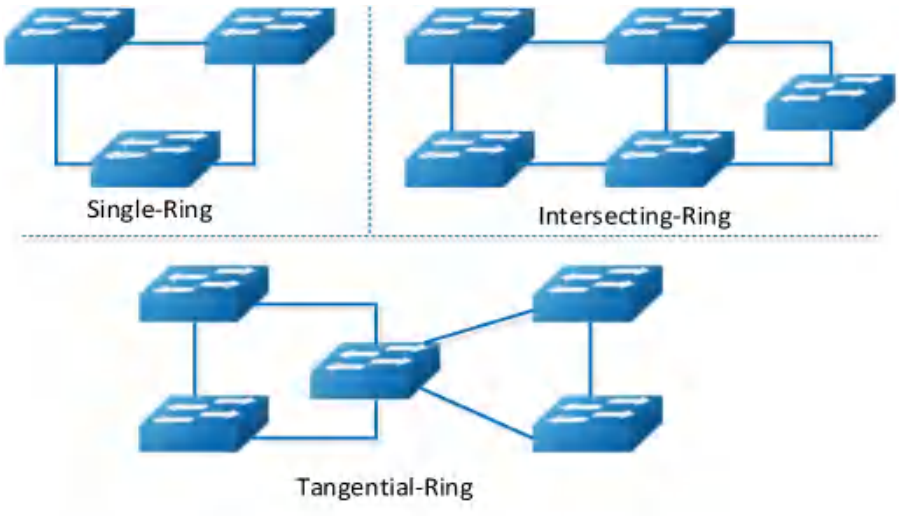


Figure 17-13 ERPS Ring type

ERPS Control VLAN:

Each ERPS domain can have two control VLAN – the main control VLAN and sub control VLAN. The ERPS protocol message of primary ring will forward in main control VLAN and ERPS protocol message of sub ring forwards in sub control VLAN. Control VLAN only transit ERPS protocol messages, data traffic will forward in Data VLAN.

Protected VLAN:

Protected VLAN is used to transmit the data packets. The protected VLAN includes ERPS and non-ERPS interface. The status of the protected VLAN is decided by its ERPS domain. Different ERPS domains on same ring have different protected VLAN. The interface status are caculated by its ERPS domain separately.

Master node:

Each switch on ring circuit is called a node. Master node is the main decision and control node on ERPS ring which initiate the detection and take action after node failure. Each ERPS ring must have a master node, and only can have one.

Transit node:

Other nodes on the ring except the master node can be called as transit node. The transit node transmit the "Hello" packet, and monitor the ERPS links directly connected, and nofity the link DOWN event to the master node.

Edge node and Assistant node

In the domain which contains primary ring and sub ring, it will generate two intersections when sub ring intersects with primary ring; one intersection is called edge node, and another is assistant node.

The Edge node and Assistant node are roles of sub ring, they are master and transit nodes on the primary ring.

Primary port and secondary port

Primary port and secondary port connect to two ports in Ethernet, one is primary port and another is secondary port. The roles of the interface is according to the user configuration. Primary port and secondary port of the master node are different. The master node send the "Hello" packets from the primary port, if the secondary port can receive this packet, it indicates that the network is healty, and the secondary port should be blocked to prevent loopback. If the secondary port can not receive the "Hello" packet in the specified time, it indicates the network has failure and the secondary port should be unblocked to ensure the network work normally. Primary port and secondary port of the transit node are same. The roles of the interface is according to the user configuration.

Common port and edge port

Edge node (assistant node) connects into two ports of sub ring, one is common port and another is edge port. Common port is the port of primary ring and sub ring's common link terminals on edge node (assistant node), edge port is the port that only connects with sub ring.

Types of ERPS protocol messages: - Hello message: is sent by master node's primary port, it examines the network looping perfectly - Link down message: when the status of happening link is changing from up to down, the message will be sent by transit node, edge node or assistant edge node of this link directly, and then notify to master node that there is a link on the ring is turning down. - Ring-up flush fdb message: is the message that the ring restores and refresh; it is sent by master node, notifies transit node, edge node or assistant node to refresh each FDB, and opens backup congestion port - Ring-down flush fdb message: ring circuit's failure refresh message; it will be sent from master node, notify transit node, edge node or assistant edge node to refresh each FDB - Edge hello message: the primary ring checks the message perfectly, it will be sent by edge node and sends to assistant edge node of the same sub ring. Sub ring will examine the link's integrity of primary ring where of the domain ring the message stays. - Major fault message: primary ring's fault notify message, if assistant edge node does not receive the edge hello message in certain time from edge node, then it will send major fault message to edge node, notifying the primary ring link's failure in the domain.

NOTE Edge node and assistant node are the roles in sub rings, these two nodes play the role as transit node on primary ring. It has no distinction between the functions on primary port and secondary port of transit node. ERPS port on primary ring join in main control VLAN and also join in sub control VLAN; ERPS port on sub ring is only join in sub control VLAN.

Working Principle

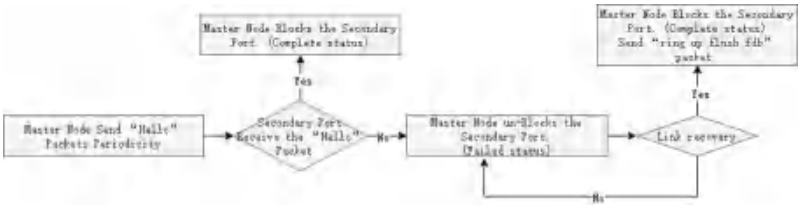


Figure 17-14 Flow chart of polling mechanism

Polling mechanism:

the master node will send hello message by its primary port periodically, it will pass through each transit node on the ring. If it can receive the hello message sent by itself, it means the ring is integrated, the master node will in complete status and block secondary port; otherwise if it cannot receive hello message within certain time (fail timer), then considers as link failure, the master node is in failed status and opens secondary port.

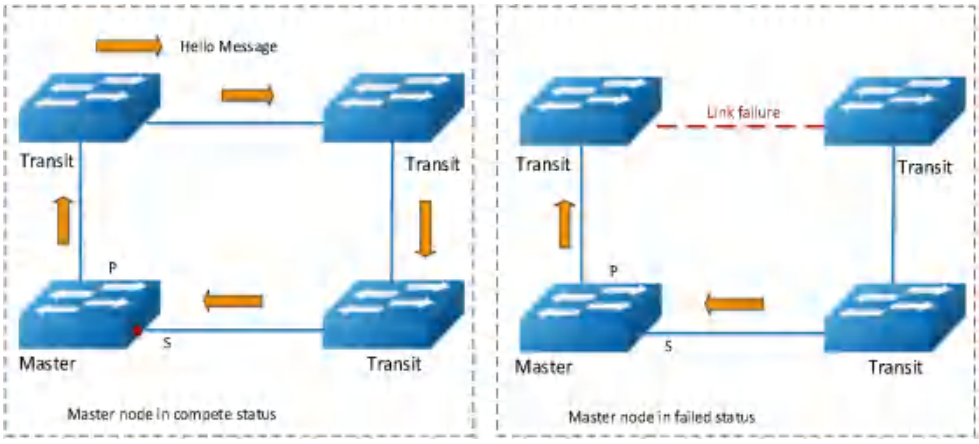


Figure 17-15 Status of Master Node



Figure 17-16 Diagram of master node's status migration

When the status of master node is migrate from complete state to failed state, it will send ring-down flush fdb message to notify other node to upgrade fdb; when the status is migrate from failed state to complete state, then it will send ring-up flush fdb message, notify other node to upgrade fdb and open temporary block port.

Transit node – failure report mechanism

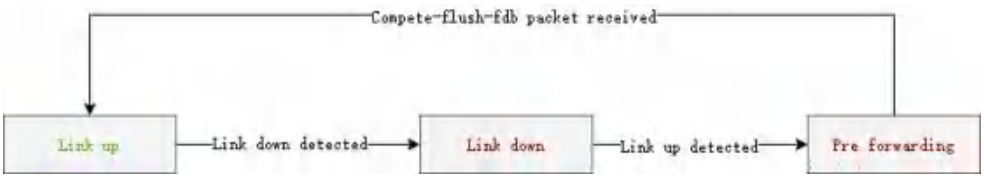


Figure 17-17 Diagram of transit node's status migration

Transit node will monitor the status of direct connected ERPS link, and report link failures to master node. Transit node has the following three statuses:

- When the primary port and secondary port from transit node are all in Up status, transit node will be link-up. When it detects link down happens on primary port or secondary port, it will migrate from link-up to link-down status, and sent link down message to notify master node about link failures.
- When the port of transit node in link-down status restores up, the transit node migrates to Pre-forwarding status, and block the restored port. The moment that primary and secondary port of transit node, master node cannot know this notification right away, so its secondary port is still open. If transit node migrates back to link-up right away, it might cause broadcast loop on ring circuit, so transit node will migrate to Pre-forwarding status first.
- When transit node in Pre-forwarding status receives the ring-up flush fdb message from master node, migrates to link-up status. If ring-up flush fdb is losing during transition, then transit node cannot receive ring-up flush fdb message within specific time, it will migrate to link-up and open temporary block port.

NOTE When the common link of primary ring and sub ring fails, it will only send link down message to master node of primary ring.

Edge node, assistant edge node – Channel Detection Mechanism of Sub Ring Protocol

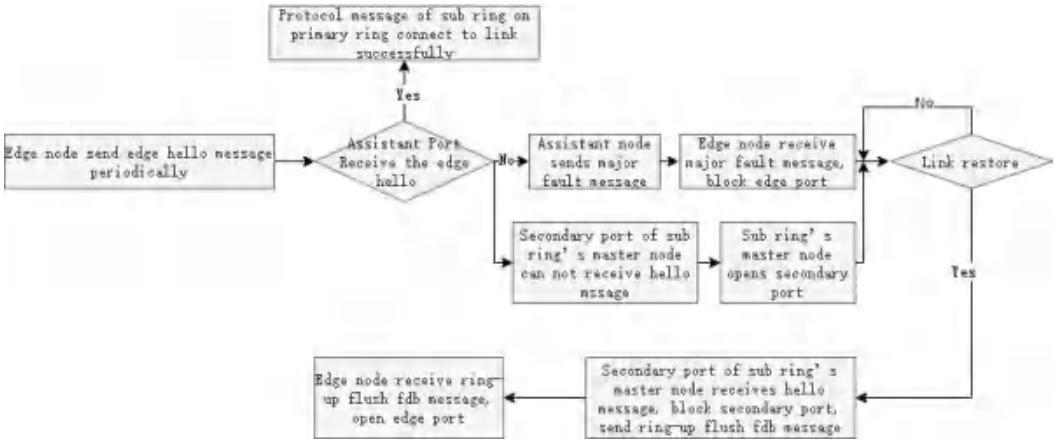


Figure 17-18 Channel detection mechanism of sub ring protocol

Channel Detection Mechanism of Sub Ring Protocol: edge node send edge hello message to assistant edge node periodically, the message with main control VLAN transits in primary ring.

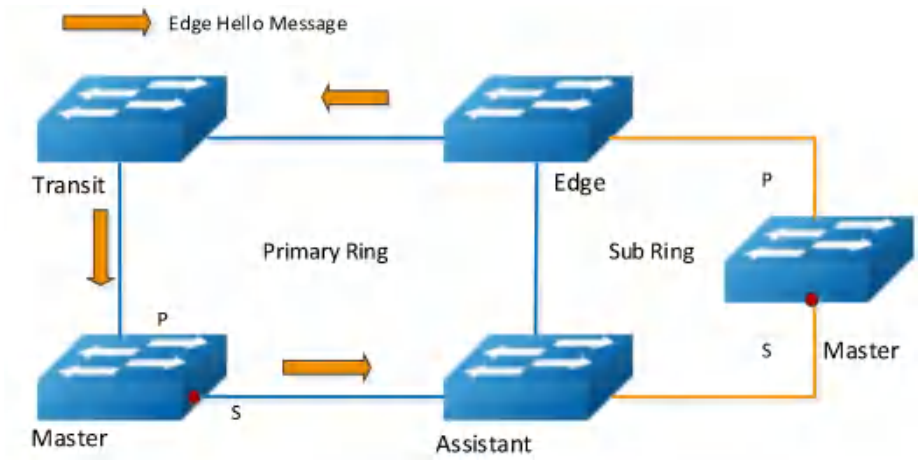


Figure 17-19 Edge node sends edge hello message

When primary link has failures, if there is only one link fails, then protocol link of sub ring is still connected. Shown as the following.

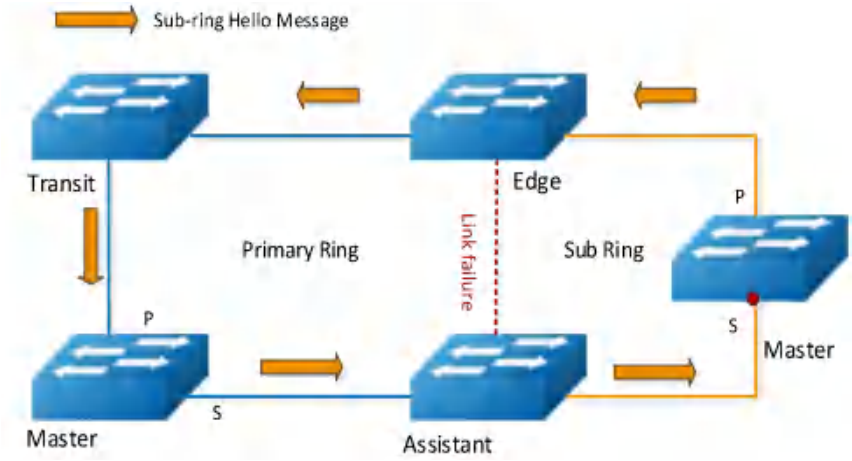


Figure 17-20 Sub ring protocol link diagram

If primary ring and sub ring's common link channel fails, and there is another link interrupts on primary ring, it will have the following response:

- Assistant edge node cannot receive edge hello message, it will send major fault message with sub control VLAN, edge node will receive major message and block its edge port:

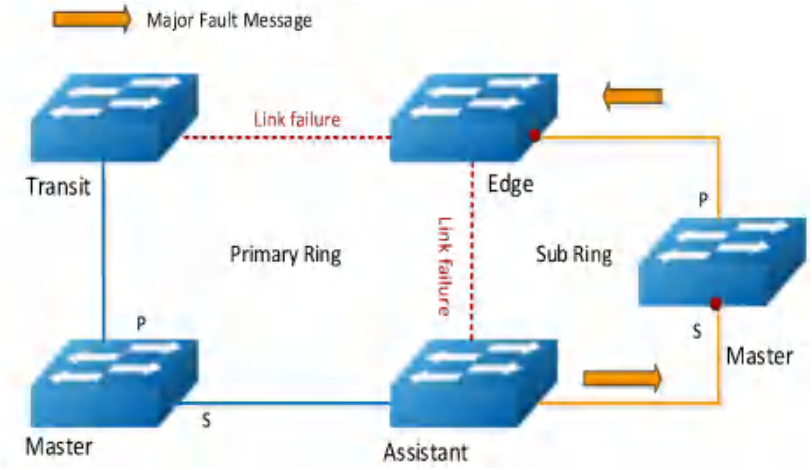


Figure 17-21 Edge node responses major fault block edge port

NOTE Edge node blocks the edge interface to prevent multi sub ring lead loop.

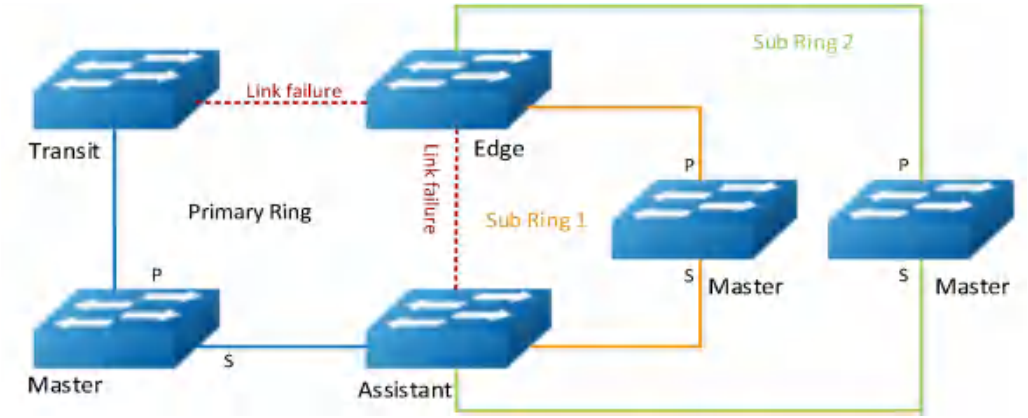


Figure 17-22 Loop in multi sub ring

If there is only one sub ring， the edge node does not need to block the edge interface. User can disable the check mechanism of the edge and assistant node by command "erps ring srpt disable".

- Secondary port of sub ring's master node does not receive hello message within specific time, sub ring's master node will in failed status, and open secondary port.

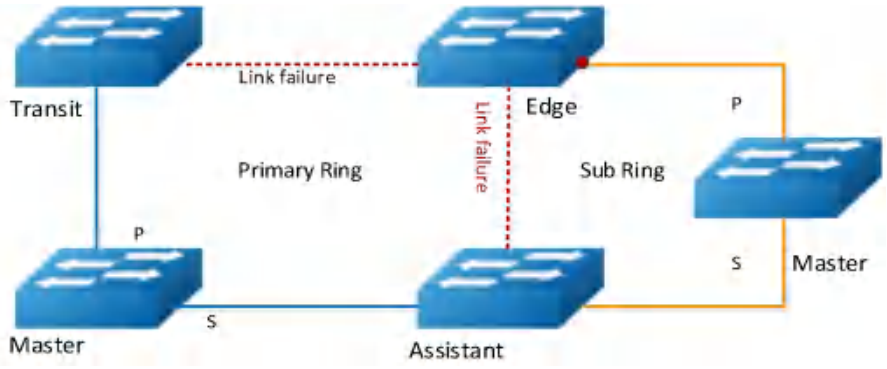


Figure 17-23 Master node of sub ring get into failed status

If sub ring protocol restores link, then secondary port of sub ring's master node receives hello message, migrates to complete status, block secondary port and sent ring-up flush fd message; after edge node receiving ring-up flush fdb message, it opens edge port. It shows in the following diagram.

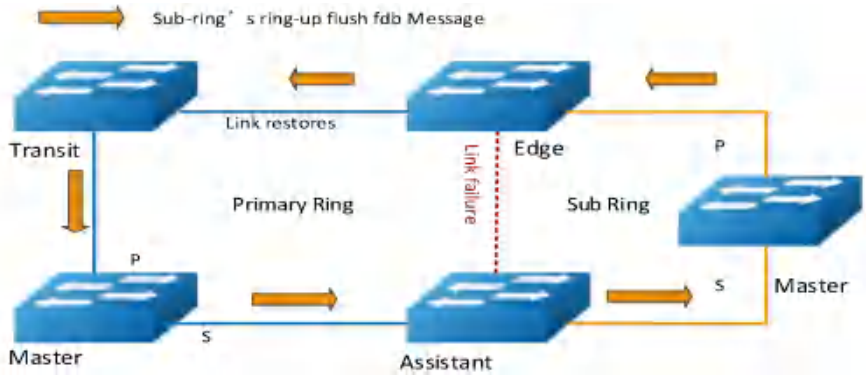


Figure 17-24 Edge node of sub ring opens edge port

Working Principle

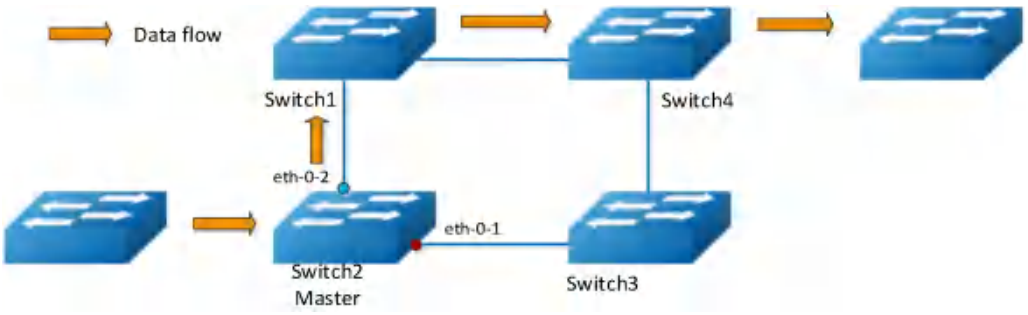


Figure 17-25 Normal ERPS networking diagram

In the figure above, in normal ERPS networking, secondary port of master node is blocked, data traffic will only forward along the orange arrows, and the bandwidth of Switch2-Switch3-Switch4 will be wasted. To improve the utilization of links, there can be two logical ERPS ring circuits in one physical loop, with these two rings corresponding a physical topology. All devices, role of ports, control vlan and

others will follow basic ERPS concepts on every ERPS ring. Every ring will detect the integrity of link independently, block or open the corresponding port; they will not affect each other. Therefore, one physical loop might have two blocked ports. Each ring needs to configure one protection instance, each instance represents a data VLAN range, and each blocked port is only valid in its ERPS instance, it can not block the VLANs which not belong to this ERPS instance. Then, traffic that is belongs to different instance can transit pass through different paths, to realize load sharing and improve the utilization of links.

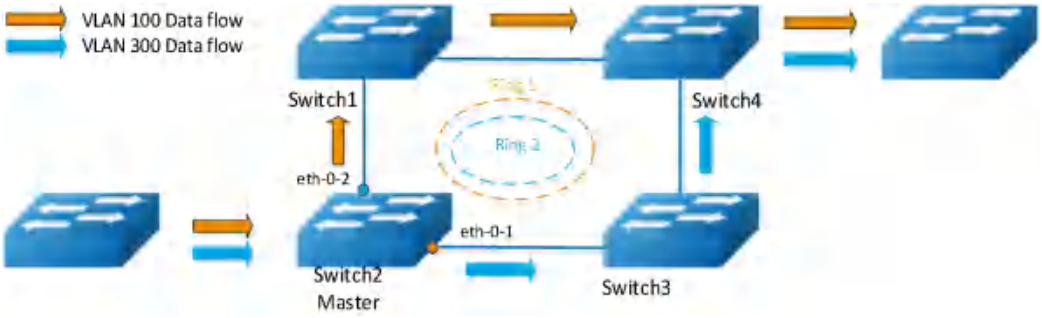


Figure 17-26 Multiple instances of network forwarding

Vlan100-200 binds with instance1, vlan300-400 binds with instance2; assign ERPS ring1 in instance1, ERPS ring2 in instance2. Ring1 and Ring2 block eth-0-1 and eth-0-2 separately, then eth-0-1 only block data message of vlan100-200, eth-0-2 only block data message of vlan300-400. Then, vlan100-200's data stream forwards through orange arrows, and vlan300-400's data stream forwards through blue arrows, to realize load sharing.

17.6.2 Configuration

ERPS is a soft-state protocol. The main requirement is to enable ERPS on desired devices, and configure the ERPS information correctly for various network topologies.

This section provides ERPS configuration examples for their typical network topologies.

Configuring ERPS for a Single-Ring Topology

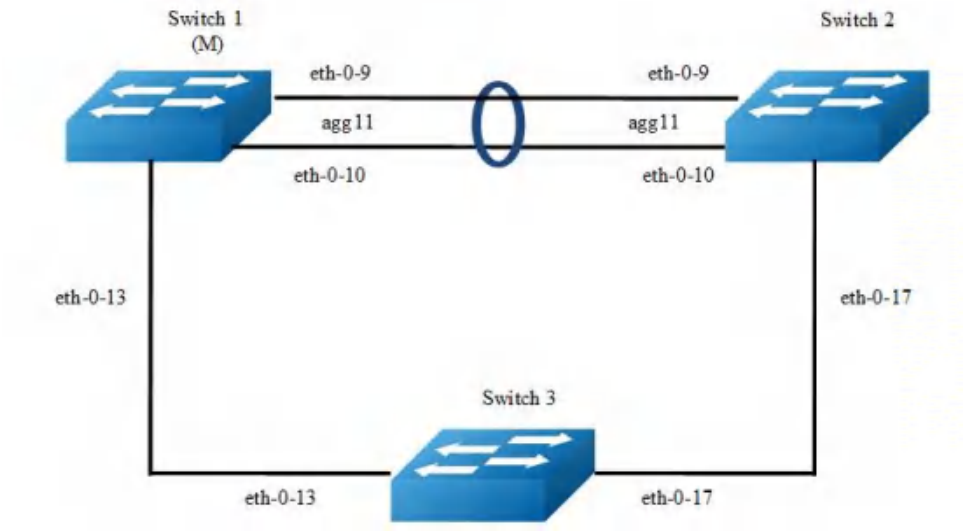


Figure 17-27 ERPS Single-Ring

Configure same ERPS domain and ring at Switch1, Switch2 and Switch3. Switch1 is configured as ERPS master node and other two switches are configured as ERPS transit nodes. Interface agg11, which has two members called eth-0-9 and eth-0-10, is configured as primary interface at Switch1 and eth-0-13 is configured as secondary interface.


The ports accessing an ERPS ring must be configured as trunk ports, permitting the traffic of data VLANs to pass through. If the switch is enabled stacking, the port of ERPS ring should not on slave stacking device.

- The ports accessing an ERPS ring must be configured as the members of the control VLAN, allowing the ERPS packets to be sent and received.
- STP on ports accessing ERPS rings must be disabled.
- Only one node can be configured as master node.
- Control VLAN must not be configured as Layer 3 interface.
- VLAN mapping must not be enabled on the ERPS ports.
- Native VLAN of a port accessing an ERPS ring must not be set as the primary control VLAN or the secondary control VLAN.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 15
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

As the topology shows， eth-0-9 and eth-0-10 of Switch1 and Switch2 join agg 11 and connect to each other directly. eth-0-13 of Switch1 and Switch3 connect to each other directly. eth-0-17 of Switch2 and Switch3 connect to each other directly.

Interface agg 11 configuration for Switch1 and Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# static-channel-group 11
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# static-channel-group 11
Switch(config-if)# exit

Switch(config)# interface agg11
Switch(config-if)# spanning-tree port disable
```

Interface eth-0-13 configuration for Switch1 and Switch3:

```
Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# spanning-tree port disable
Switch(config-if)# exit
```

Interface eth-0-17 configuration for Switch2 and Switch3:

```
Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 15
Switch(config-if)# spanning-tree port disable
Switch(config-vlan)# exit
```

step 4 Create and enable ERPS domain.

ERPS domain for Switch1:

```
Switch(config)# erps 11
Switch(config)# erps 11 primary control vlan 15
Switch(config)# erps 11 mstp instance 0
Switch(config)# erps 11 ring 1 level primary
Switch(config)# erps 11 ring 1 mode master
Switch(config)# erps 11 ring 1 primary interface agg11
Switch(config)# erps 11 ring 1 secondary interface eth-0-13
Switch(config)# erps 11 ring 1 enable
Switch(config)# erps 11 enable
```

ERPS domain for Switch2:

```
Switch(config)# erps 11
Switch(config)# erps 11 primary control vlan 15
Switch(config)# erps 11 mstp instance 0
Switch(config)# erps 11 ring 1 level primary
Switch(config)# erps 11 ring 1 mode transit
Switch(config)# erps 11 ring 1 primary interface agg11
Switch(config)# erps 11 ring 1 secondary interface eth-0-17
Switch(config)# erps 11 ring 1 enable
Switch(config)# erps 11 enable
```

ERPS domain for Switch3:

```
Switch(config)# erps 11
Switch(config)# erps 11 primary control vlan 15
Switch(config)# erps 11 mstp instance 0
Switch(config)# erps 11 ring 1 level primary
Switch(config)# erps 11 ring 1 mode transit
Switch(config)# erps 11 ring 1 primary interface eth-0-17
Switch(config)# erps 11 ring 1 secondary interface eth-0-13
Switch(config)# erps 11 ring 1 enable
Switch(config)# erps 11 enable
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch# show erps 11
ERPS domain ID: 11
ERPS domain name: ERPS0011
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 15
ERPS domain sub control VLAN ID: 0
ERPS domain hello timer interval: 1 second(s)
```

```
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: master
ERPS ring 1 node state: complete
ERPS ring 1 primary interface name: agg11        state:unblock
ERPS ring 1 secondary interface name: eth-0-13    state:block
ERPS ring 1 stats:
Sent:
 total packets:51
 hello packets:47                ring-up-flush-fdb packets:2
 ring-down-flush-fdb packets:2     link-down packets:0
 edge-hello packets:0            major-fault packets:0
Received:
 total packets:21
 hello packets:21                ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0     link-down packets:0
 edge-hello packets:0            major-fault packets:0
```

Display the result on Switch2.

```
Switch# show erps 11
ERPS domain ID: 11
ERPS domain name: ERPS0011
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 15
ERPS domain sub control VLAN ID: 0
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: transit
ERPS ring 1 node state: link up
ERPS ring 1 primary interface name: agg11        state:unblock
ERPS ring 1 secondary interface name: eth-0-17    state:unblock
ERPS ring 1 stats:
Sent:
 total packets:0
 hello packets:0                 ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0     link-down packets:0
 edge-hello packets:0            major-fault packets:0
Received:
 total packets:114
 hello packets:113               ring-up-flush-fdb packets:1
 ring-down-flush-fdb packets:0     link-down packets:0
 edge-hello packets:0            major-fault packets:0
```

Display the result on Switch3.

```
Switch# show erps 11
ERPS domain ID: 11
ERPS domain name: ERPS0011
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 15
ERPS domain sub control VLAN ID: 0
```

```
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: transit
ERPS ring 1 node state: link up
ERPS ring 1 primary interface name: eth-0-17      state:unblock
ERPS ring 1 secondary interface name: eth-0-13    state:unblock
ERPS ring 1 stats:
Sent:
 total packets:0
 hello packets:0                ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
Received:
 total packets:130
 hello packets:129              ring-up-flush-fdb packets:1
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
```

Configuring a Intersecting-Ring Topology

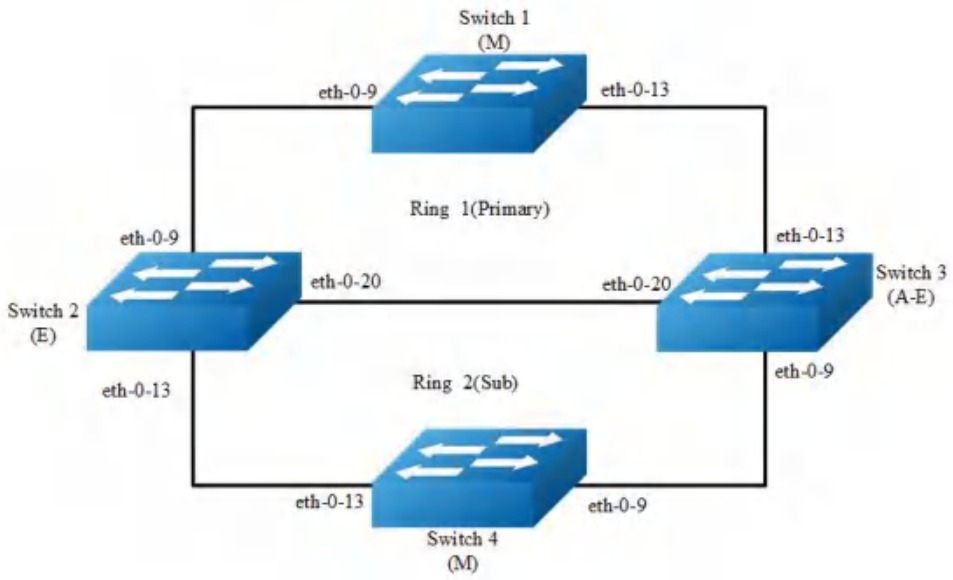


Figure 17-28 ERPS Intersecting-Ring

Configure same ERPS domain at Switch1, Switch2, Switch3 and Switch4. Switch1, Switch2 and Switch3 consist of ERPS primary ring 1 while Switch2, Switch3 and Switch4 consist of ERPS sub ring 2. Switch1 is configured as ERPS ring 1 master node and other two switches are configured as ERPS transit nodes while Switch4 is configured as ERPS ring 2 master node. In addition Switch2 is configured as edge node and Switch3 is configured as assistant-edge node.

The ports accessing an ERPS ring must be configured as trunk ports, permitting the traffic of data VLANs to pass through.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 11,12
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-9
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 11,12
Switch(config-if)# spanning-tree port disable
Switch(config-if)# exit

Switch(config)# interface eth-0-13
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 11,12
Switch(config-if)# spanning-tree port disable
Switch(config-if)# exit
```

Interface eth-0-20 configuration for Switch2 and Switch3:

```
Switch(config)# interface eth-0-20
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 11,12
Switch(config-if)# spanning-tree port disable
Switch(config-if)# exit
```

step 4 Create and enable ERPS domain.

ERPS domain for Switch1:

```
Switch(config)# erps 1
Switch(config)# erps 1 primary control vlan 11
Switch(config)# erps 1 sub control vlan 12
Switch(config)# erps 1 mstp instance 0
Switch(config)# erps 1 ring 1 level primary
Switch(config)# erps 1 ring 1 mode master
Switch(config)# erps 1 ring 1 primary interface eth-0-9
Switch(config)# erps 1 ring 1 secondary interface eth-0-13
Switch(config)# erps 1 ring 1 enable
Switch(config)# erps 1 enable
```

ERPS domain for Switch2:

```
Switch(config)# erps 1
Switch(config)# erps 1 primary control vlan 11
Switch(config)# erps 1 sub control vlan 12
Switch(config)# erps 1 mstp instance 0
Switch(config)# erps 1 ring 1 level primary
Switch(config)# erps 1 ring 1 mode transit
Switch(config)# erps 1 ring 1 primary interface eth-0-9
Switch(config)# erps 1 ring 1 secondary interface eth-0-20
Switch(config)# erps 1 ring 1 enable

Switch(config)# erps 1 ring 2 level sub
Switch(config)# erps 1 ring 2 edge-mode edge
Switch(config)# erps 1 ring 2 edge interface eth-0-13
```

```
Switch(config)# erps 1 ring 2 common interface eth-0-20
Switch(config)# erps 1 ring 2 srpt disable
Switch(config)# erps 1 ring 2 enable
Switch(config)# erps 1 enable
```

ERPS domain for Switch3:

```
Switch(config)# erps 1
Switch(config)# erps 1 primary control vlan 11
Switch(config)# erps 1 sub control vlan 12
Switch(config)# erps 1 mstp instance 0
Switch(config)# erps 1 ring 1 level primary
Switch(config)# erps 1 ring 1 mode transit
Switch(config)# erps 1 ring 1 primary interface eth-0-13
Switch(config)# erps 1 ring 1 secondary interface eth-0-20
Switch(config)# erps 1 ring 1 enable

Switch(config)# erps 1 ring 2 level sub
Switch(config)# erps 1 ring 2 edge-mode assistant-edge
Switch(config)# erps 1 ring 2 edge interface eth-0-9
Switch(config)# erps 1 ring 2 common interface eth-0-20
Switch(config)# erps 1 ring 2 enable
Switch(config)# erps 1 enable
```

ERPS domain for Switch4:

```
Switch(config)# erps 1
Switch(config)# erps 1 sub control vlan 12
Switch(config)# erps 1 mstp instance 0
Switch(config)# erps 1 ring 2 level sub
Switch(config)# erps 1 ring 2 mode master
Switch(config)# erps 1 ring 2 primary interface eth-0-9
Switch(config)# erps 1 ring 2 secondary interface eth-0-13
Switch(config)# erps 1 ring 2 enable
Switch(config)# erps 1 enable
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch# show erps 1
ERPS domain ID: 1
ERPS domain name: ERPS001
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 11
ERPS domain sub control VLAN ID: 12
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: master
ERPS ring 1 node state: complete
ERPS ring 1 primary interface name: eth-0-9       state:unblock
ERPS ring 1 secondary interface name: eth-0-13    state:block
```

```
ERPS ring 1 stats:
Sent:
 total packets:1310
 hello packets:1303              ring-up-flush-fdb packets:3
 ring-down-flush-fdb packets:4   link-down packets:0
 edge-hello packets:0            major-fault packets:0
Received:
 total packets:921
 hello packets:921               ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0   link-down packets:0
 edge-hello packets:0            major-fault packets:0
```

Display the result on Switch2.

```
Switch# show erps 1
ERPS domain ID: 1
ERPS domain name: ERPS001
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 11
ERPS domain sub control VLAN ID: 12
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: transit
ERPS ring 1 node state: link up
ERPS ring 1 primary interface name: eth-0-9     state:unblock
ERPS ring 1 secondary interface name: eth-0-20   state:unblock
ERPS ring 1 stats:
Sent:
 total packets:0
 hello packets:0                 ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0   link-down packets:0
 edge-hello packets:0            major-fault packets:0
Received:
 total packets:988
 hello packets:985               ring-up-flush-fdb packets:2
 ring-down-flush-fdb packets:1   link-down packets:0
 edge-hello packets:0            major-fault packets:0
ERPS ring ID: 2
ERPS ring level: sub
ERPS ring 2 node mode: transit
ERPS ring 2 edge node mode: edge
ERPS ring 2 node state: link up
ERPS ring 2 edge interface name: eth-0-13     state: unblock
ERPS ring 2 common interface name: eth-0-20   state: unblock
EPRS ring 2 SRPT is disabled
ERPS ring 2 stats:
Sent:
 total packets:0
 hello packets:0                 ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0   link-down packets:0
 edge-hello packets:0            major-fault packets:0
Received:
 total packets:858
```

```
 hello packets:856              ring-up-flush-fdb packets:1
 ring-down-flush-fdb packets:1      link-down packets:0
 edge-hello packets:0           major-fault packets:0
```

Display the result on Switch3.

```
Switch# show erps 1
ERPS domain ID: 1
ERPS domain name: ERPS001
ERPS domain mode: normal
ERPS domain primary control VLAN ID: 11
ERPS domain sub control VLAN ID: 12
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 1
ERPS ring level: primary
ERPS ring 1 node mode: transit
ERPS ring 1 node state: link up
ERPS ring 1 primary interface name: eth-0-13      state:unblock
ERPS ring 1 secondary interface name: eth-0-20    state:unblock
ERPS ring 1 stats:
Sent:
 total packets:0
 hello packets:0              ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
Received:
 total packets:645
 hello packets:644              ring-up-flush-fdb packets:1
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
ERPS ring ID: 2
ERPS ring level: sub
ERPS ring 2 node mode: transit
ERPS ring 2 edge node mode: assistant edge
ERPS ring 2 node state: link up
ERPS ring 2 edge interface name: eth-0-9      state: unblock
ERPS ring 2 common interface name: eth-0-20    state: unblock
ERPS ring 2 stats:
Sent:
 total packets:0
 hello packets:0              ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
Received:
 total packets:645
 hello packets:644              ring-up-flush-fdb packets:1
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
```

Display the result on Switch4.

```
Switch# show erps 1
ERPS domain ID: 1
ERPS domain name: ERPS001
ERPS domain mode: normal
```

```
ERPS domain primary control VLAN ID: 0
ERPS domain sub control VLAN ID: 12
ERPS domain hello timer interval: 1 second(s)
ERPS domain fail timer interval: 3 second(s)
ERPS domain protected mstp instance: 0
ERPS ring ID: 2
ERPS ring level: sub
ERPS ring 2 node mode: master
ERPS ring 2 node state: complete
ERPS ring 2 primary interface name: eth-0-9       state:unblock
ERPS ring 2 secondary interface name: eth-0-13    state:block
ERPS ring 2 stats:
Sent:
 total packets:814
 hello packets:810              ring-up-flush-fdb packets:2
 ring-down-flush-fdb packets:2      link-down packets:0
 edge-hello packets:0           major-fault packets:0
Received:
 total packets:774
 hello packets:774              ring-up-flush-fdb packets:0
 ring-down-flush-fdb packets:0      link-down packets:0
 edge-hello packets:0           major-fault packets:0
Switch#
```

Configuring a Tangent Rings Networking

Topology

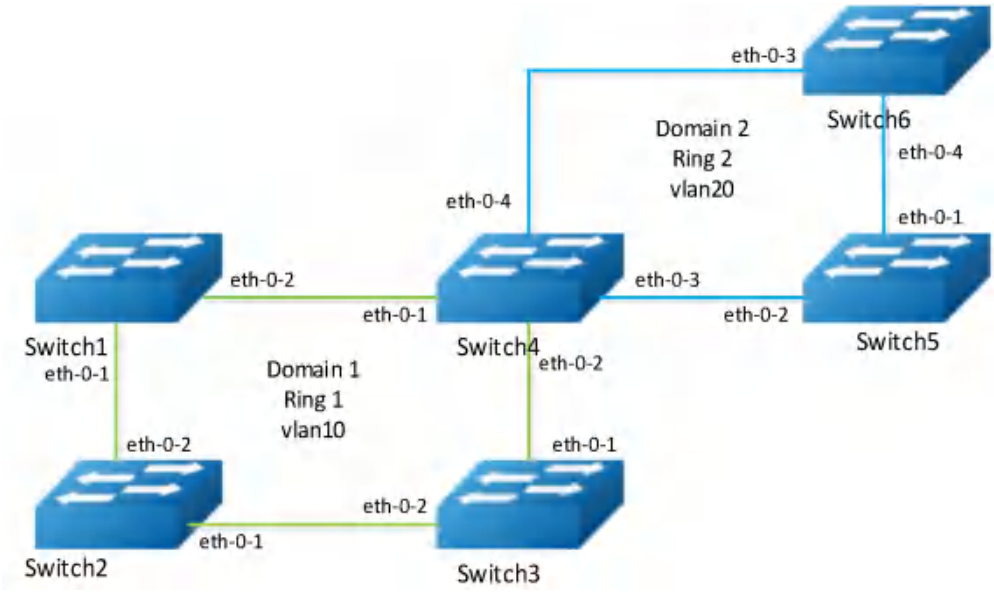


Figure 17-29 Tangency ring networking diagram

Configuration Steps Ideas

Switch 1/2/3/4/5/6 constitute into two rings, executing ERPS protocol, provide Layer 2 redundancy protection of these two rings.

- In topology, the two rings without common physical links are not fit with configuring into primary ring – sub ring mode. It can consider as two independent rings inside the two erps domains.
- Configure 1/2/3/4 as ring1 in erps domain1; 2 is the master node, eth-0-1 is primary port, eth-0-2 is secondary port.
- Configure 4/5/6 as ring2 in erps domain2; 6 is master node, eth-0-3 is primary port and eth-0-4 is secondary port.

Configuration Steps Steps

NOTE If there is no clear explanation, use "SwitchX" to declare that the configuration is same for all switches (Switch 1/2/3/4/5/6).

step 1 Enter the configure mode

```
SwitchX# configure terminal
```

step 2 Configuring VLAN 10 for Ring 1 devices (Switch 1/2/3/4)

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 10
SwitchX(config-vlan)# exit
SwitchX(config)# no ip igmp snooping vlan 10
```

NOTE ERPS packets are multicast, so we igmp snooping must be disabled for the control VLAN.

step 3 Configuring VLAN 20 for Ring 2 devices (Switch 4/5/6)

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 20
SwitchX(config-vlan)# exit
SwitchX(config)# no ip igmp snooping vlan 20
```

step 4 Configuring interface attributes

```
SwitchX(config)# interface eth-0-X
SwitchX(config-if)# switchport
SwitchX(config-if)# no shutdown
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk allowed vlan all
SwitchX(config-if)# spanning-tree port disable
SwitchX(config-if)# exit
```

NOTE According to the topology figure, the eth-0-X is eth-0-1/eth-0-2/eth-0-3/eth-0-4, the configuration is same.

step 5 Configuring interface attributes for Switch 4

```
Switch4(config)# interface eth-0-1
Switch4(config-if)# switchport trunk allowed vlan remove 20
Switch4(config-if)# exit
Switch4(config)# interface eth-0-2
Switch4(config-if)# switchport trunk allowed vlan remove 20
Switch4(config-if)# exit
Switch4(config)# interface eth-0-3
Switch4(config-if)# switchport trunk allowed vlan remove 10
Switch4(config-if)# exit
Switch4(config)# interface eth-0-4
Switch4(config-if)# switchport trunk allowed vlan remove 10
Switch4(config-if)# exit
```

NOTE According to the network requirement, remove VLAN 20 on interface eth-0-1/eth-0-2 and remove VLAN 10 on interface eth-0-3/eth-0-4.

step 6 Configuring master node switch2 in Domain1

```
Switch2(config)# erps 1
Switch2(config)# erps 1 primary control vlan 10
```

```
Switch2(config)# erps 1 ring 1 level primary
Switch2(config)# erps 1 ring 1 mode master
Switch2(config)# erps 1 ring 1 primary interface eth-0-1
Switch2(config)# erps 1 ring 1 secondary interface eth-0-2
Switch2(config)# erps 1 mstp instance 0
Switch2(config)# erps 1 ring 1 enable
Switch2(config)# end
```

step 7 Configure transit node switch1 in Domain1

```
Switch1(config)# erps 1
Switch1(config)# erps 1 primary control vlan 10
Switch1(config)# erps 1 ring 1 level primary
Switch1(config)# erps 1 ring 1 mode transit
Switch1(config)# erps 1 ring 1 primary interface eth-0-1
Switch1(config)# erps 1 ring 1 secondary interface eth-0-2
Switch1(config)# erps 1 mstp instance 0
Switch1(config)# erps 1 ring 1 enable
```

step 8 Configure transit node switch3 in Domain1

```
Switch3(config)# erps 1
Switch3(config)# erps 1 primary control vlan 10
Switch3(config)# erps 1 ring 1 level primary
Switch3(config)# erps 1 ring 1 mode transit
Switch3(config)# erps 1 ring 1 primary interface eth-0-1
Switch3(config)# erps 1 ring 1 secondary interface eth-0-2
Switch3(config)# erps 1 mstp instance 0
Switch3(config)# erps 1 ring 1 enable
```

step 9 Configure master node switch6 in Domain2

```
Switch6(config)# erps 2
Switch6(config)# erps 2 primary control vlan 20
Switch6(config)# erps 2 ring 2 level primary
Switch6(config)# erps 2 ring 2 mode master
Switch6(config)# erps 2 ring 2 primary interface eth-0-3
Switch6(config)# erps 2 ring 2 secondary interface eth-0-4
Switch6(config)# erps 2 mstp instance 0
Switch6(config)# erps 2 ring 2 enable
```

step 10 Configure transit node switch5 in Domain2

```
Switch5(config)# erps 2
Switch5(config)# erps 2 primary control vlan 10
Switch5(config)# erps 2 ring 1 level primary
Switch5(config)# erps 2 ring 1 mode transit
Switch5(config)# erps 2 ring 1 primary interface eth-0-1
Switch5(config)# erps 2 ring 1 secondary interface eth-0-2
Switch5(config)# erps 2 mstp instance 0
Switch5(config)# erps 2 ring 1 enable
```

step 11 Configuring ring1 attributes intersect node switch4

```
Switch4(config)# erps 1
Switch4(config)# erps 1 primary control vlan 10
Switch4(config)# erps 1 ring 1 level primary
Switch4(config)# erps 1 ring 1 mode transit
```

```
Switch4(config)# erps 1 ring 1 primary interface eth-0-1
Switch4(config)# erps 1 ring 1 secondary interface eth-0-2
Switch4(config)# erps 1 mstp instance 0
Switch4(config)# erps 1 ring 1 enable
```

step 12 Configuring ring2 attributes intersect node switch4

```
Switch4(config)# erps 2
Switch4(config)# erps 2 primary control vlan 20
Switch4(config)# erps 2 ring 2 level primary
Switch4(config)# erps 2 ring 2 mode transit
Switch4(config)# erps 2 ring 2 primary interface eth-0-3
Switch4(config)# erps 2 ring 2 secondary interface eth-0-4
Switch4(config)# erps 2 mstp instance 0
Switch4(config)# erps 2 ring 2 enable
```

step 13 Exit configure mode

```
SwitchX(config)# end
```

Configuring ERPS Multi sub Rings Networking

Topology

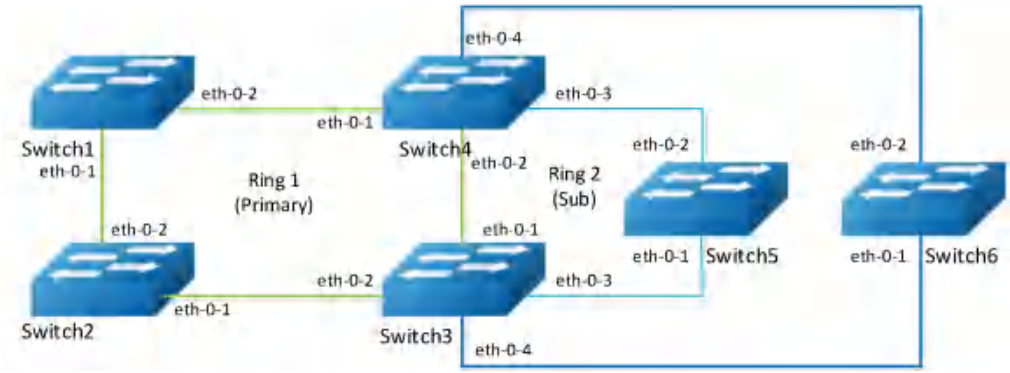


Figure 17-30 Multi sub rings networking diagram

Configuration Steps Ideas

Switch 1/2/3/4/5/6 constitute into intersect ring network executing ERPS protocol, provide Layer 2 redundancy protection.

- There are three intersect rings; therefore, considering to configure one primary ring & two sub ring mode.
- There are multi sub ring with edge and assistant node, so srpt should be enabled on one sub ring and disabled on another in order to prevent one sub ring become a network island when the assistant nodes are all blocked.

Configuration Steps Steps

NOTE If there is no clear explanation, use "SwitchX" to declare that the configuration is same for all switches (Switch 1/2/3/4/5/6).

step 1 Enter the configure mode

```
SwitchX# configure terminal
```

step 2 Configuring VLAN 10 for Primary Ring devices (Switch 1/2/3/4)

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 10,20
SwitchX(config-vlan)# exit
```

```
SwitchX(config)# no ip igmp snooping vlan 10
SwitchX(config)# no ip igmp snooping vlan 20
```

step 3 Configuring VLAN 20 for Ring 2 devices (Switch 5/6)

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 20
SwitchX(config-vlan)# exit
SwitchX(config)# no ip igmp snooping vlan 20
```

step 4 Configuring interface attributes

```
SwitchX(config)# interface eth-0-X
SwitchX(config-if)# switchport
SwitchX(config-if)# no shutdown
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk allowed vlan all
SwitchX(config-if)# spanning-tree port disable
SwitchX(config-if)# exit
```

NOTE According to the topology figure, the eth-0-X is eth-0-1/eth-0-2/eth-0-3/eth-0-4, the configuration is same.

step 5 Configuring interface attributes for assistant-edge witch 3/4

```
SwitchX(config)# interface eth-0-3
SwitchX(config-if)# switchport trunk allowed vlan remove 10
SwitchX(config)# interface eth-0-4
SwitchX(config-if)# switchport trunk allowed vlan remove 10
```

NOTE According to the network requirement, remove VLAN 10 on sub ring's interface eth-0-3/eth-0-4 of assistant-edge.

step 6 Configure master node switch2 in primary ring

```
Switch2(config)# erps 1
Switch2(config)# erps 1 primary control vlan 10
Switch2(config)# erps 1 ring 1 level primary
Switch2(config)# erps 1 ring 1 mode master
Switch2(config)# erps 1 ring 1 primary interface eth-0-1
Switch2(config)# erps 1 ring 1 secondary interface eth-0-2
Switch2(config)# erps 1 sub control vlan 20
Switch2(config)# erps 1 mstp instance 0
Switch2(config)# erps 1 ring 1 enable
```

step 7 Configure transit node switch 1

```
Switch1(config)# erps 1
Switch1(config)# erps 1 primary control vlan 10
Switch1(config)# erps 1 ring 1 level primary
Switch1(config)# erps 1 ring 1 mode transit
Switch1(config)# erps 1 ring 1 primary interface eth-0-1
Switch1(config)# erps 1 ring 1 secondary interface eth-0-2
Switch1(config)# erps 1 sub control vlan 20
Switch1(config)# erps 1 mstp instance 0
Switch1(config)# erps 1 ring 1 enable
```

step 8 Configurring primary ring1 attributes assistant-edge switch4

```
Switch4(config)# erps 1
Switch4(config)# erps 1 primary control vlan 10
Switch4(config)# erps 1 sub control vlan 20
Switch4(config)# erps 1 ring 1 level primary
Switch4(config)# erps 1 ring 1 mode master
Switch4(config)# erps 1 ring 1 primary interface eth-0-1
Switch4(config)# erps 1 ring 1 secondary interface eth-0-2
Switch4(config)# erps 1 mstp instance 0
Switch4(config)# erps 1 ring 1 enable
```

step 9 Configurring sub ring2 attributes assistant-edge switch4

```
Switch4(config)# erps 1 ring 2 level sub
Switch4(config)# erps 1 ring 2 edge-mode edge
Switch4(config)# erps 1 ring 2 edge interface eth-0-3
Switch4(config)# erps 1 ring 2 common interface eth-0-2
Switch4(config)# erps 1 ring 2 srpt disable
Switch4(config)# erps 1 ring 2 enable
```

step 10 Configurring sub ring3 attributes assistant-edge switch4

```
Switch4(config)# erps 1 ring 3 level sub
Switch4(config)# erps 1 ring 3 edge-mode assistant-edge
Switch4(config)# erps 1 ring 3 edge interface eth-0-4
Switch4(config)# erps 1 ring 3 common interface eth-0-2
Switch4(config)# erps 1 ring 3 enable
```

step 11 Configurring primary ring1 attributes assistant-edge switch3

```
Switch3(config)# erps 1
Switch3(config)# erps 1 primary control vlan 10
Switch3(config)# erps 1 sub control vlan 20
Switch3(config)# erps 1 ring 1 level primary
Switch3(config)# erps 1 ring 1 mode master
Switch3(config)# erps 1 ring 1 primary interface eth-0-1
Switch3(config)# erps 1 ring 1 secondary interface eth-0-2
Switch3(config)# erps 1 mstp instance 0
Switch3(config)# erps 1 ring 1 enable
```

step 11 Configurring sub ring2 attributes assistant-edge switch3

```
Switch3(config)# erps 1 ring 2 level sub
Switch3(config)# erps 1 ring 2 edge-mode assistant-edge
Switch3(config)# erps 1 ring 2 edge interface eth-0-3
Switch3(config)# erps 1 ring 2 common interface eth-0-1
Switch3(config)# erps 1 ring 2 srpt disable
Switch3(config)# erps 1 ring 2 enable
```

step 12 Configurring sub ring3 attributes assistant-edge switch3

```
Switch3(config)# erps 1 ring 3 level sub
Switch3(config)# erps 1 ring 3 edge-mode edge
Switch3(config)# erps 1 ring 3 edge interface eth-0-4
Switch3(config)# erps 1 ring 3 common interface eth-0-1
Switch3(config)# erps 1 ring 3 enable
```

step 13 Configure master node switch5 in sub ring

```
Switch5(config)# erps 1
Switch5(config)# erps 1 sub control vlan 20
Switch5(config)# erps 1 ring 2 level sub
Switch5(config)# erps 1 ring 2 mode master
Switch5(config)# erps 1 ring 2 primary interface eth-0-1
Switch5(config)# erps 1 ring 2 secondary interface eth-0-2
Switch5(config)# erps 1 mstp instance 0
Switch5(config)# erps 1 ring 2 enable
```

step 14 Configure master node switch6 in sub ring

```
Switch6(config)# erps 1
Switch6(config)# erps 1 sub control vlan 20
Switch6(config)# erps 1 ring 3 level sub
Switch6(config)# erps 1 ring 3 mode master
Switch6(config)# erps 1 ring 3 primary interface eth-0-2
Switch6(config)# erps 1 ring 3 secondary interface eth-0-1
Switch6(config)# erps 1 mstp instance 0
Switch6(config)# erps 1 ring 3 enable
```

step 15 Exit configure mode

```
SwitchX(config)# end
```

Configuring ERPS Multi-instance Networking

Topology

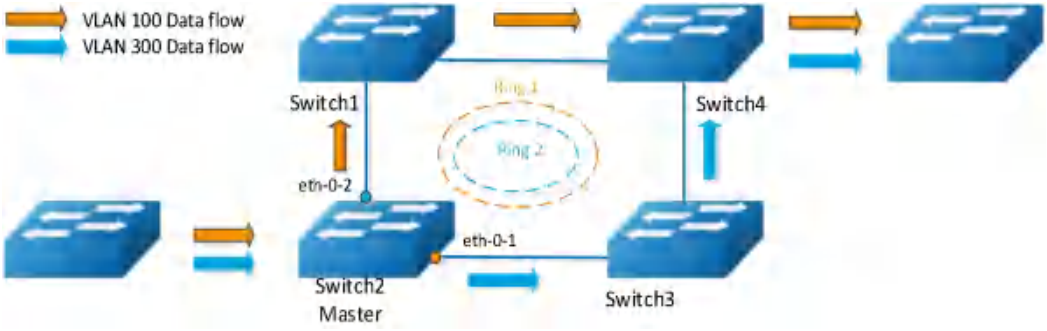


Figure 17-31 Multiple instances networking diagram

Configuration Steps Ideas

Switch 1/2/3/4 constitute into ring network and execute ERPS protocol, provide Layer 2 redundancy protection. Meanwhile, configures multiple instances, the traffic from switch2 forwards to switch4, data stream of vlan100-200 follows the forwarding route of switch2-1-4, and data stream of vlan300-400 follows the forwarding route of switch2-3-4. This can realize load sharing, and it will improve the utilization of links.

- Configure switch as the master node of domain 1 ring1 and domain 2 ring2, ring1's secondary port as eth-0-1, and ring2's secondary port as eth-0-2
- Configure instance instance1 binds with vlan100-200, instance2 binds with vlan300-400
- Assign domain1 is in instance1, domain2 is in instance2.

Configuration Steps Steps

NOTE If there is no clear explanation, use "SwitchX" to declare that the configuration is same for all switches (Switch 1/2/3/4).

step 1 Enter the configure mode

```
SwitchX# configure terminal
```

step 2 Configuring VLAN 10/20 for Control VLAN

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 10,20,100-200,300-400
SwitchX(config-vlan)# exit
SwitchX(config)# no ip igmp snooping vlan 10
SwitchX(config)# no ip igmp snooping vlan 20
```

step 3 Configuring VLAN 100-200/300-400 for Data flow

```
SwitchX(config)# vlan database
SwitchX(config-vlan)# vlan 100-200,300-400
SwitchX(config-vlan)# exit
```

step 4 Configuring interface attributes

```
SwitchX(config)# interface eth-0-1
SwitchX(config-if)# switchport
SwitchX(config-if)# no shutdown
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk allowed vlan all
SwitchX(config-if)# spanning-tree port disable
SwitchX(config-if)# exit
SwitchX(config)# interface eth-0-2
SwitchX(config-if)# switchport
SwitchX(config-if)# no shutdown
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk allowed vlan all
SwitchX(config-if)# spanning-tree port disable
SwitchX(config-if)# exit
```

step 5 Configuring MSTP instance

```
SwitchX(config)# spanning-tree mode mstp
SwitchX(config)# spanning-tree mst configuration
SwitchX(config-mst)# instance 1 vlan 10,100-200
SwitchX(config-mst)# instance 2 vlan 20,300-400
```

NOTE It is suggested to add control VLAN of ERPS ring into protection instance of rings.

step 6 Configurring ring1 on master node switch2

```
Switch2(config)# erps 1
Switch2(config)# erps 1 primary control vlan 10
Switch2(config)# erps 1 ring 1 level primary
Switch2(config)# erps 1 ring 1 mode master
Switch2(config)# erps 1 ring 1 primary interface eth-0-2
Switch2(config)# erps 1 ring 1 secondary interface eth-0-1
Switch2(config)# erps 1 mstp instance 1
Switch2(config)# erps 1 ring 1 enable
```

step 7 Configurring ring2 on master node switch2

```
Switch2(config)# erps 2
Switch2(config)# erps 2 primary control vlan 20
Switch2(config)# erps 2 ring 2 level primary
```

```
Switch2(config)# erps 2 ring 2 mode master
Switch2(config)# erps 2 ring 2 primary interface eth-0-1
Switch2(config)# erps 2 ring 2 secondary interface eth-0-2
Switch2(config)# erps 2 mstp instance 2
Switch2(config)# erps 2 ring 2 enable
```

step 8 Configurring ring1 on transit node switch1

```
Switch1(config)# erps 1
Switch1(config)# erps 1 primary control vlan 10
Switch1(config)# erps 1 ring 1 level primary
Switch1(config)# erps 1 ring 1 mode transit
Switch1(config)# erps 1 ring 1 primary interface eth-0-1
Switch1(config)# erps 1 ring 1 secondary interface eth-0-2
Switch1(config)# erps 1 mstp instance 1
Switch1(config)# erps 1 ring 1 enable
```

step 9 Configurring ring2 on transit node switch1

```
Switch1(config)# erps 2
Switch1(config)# erps 2 primary control vlan 20
Switch1(config)# erps 2 ring 2 level primary
Switch1(config)# erps 2 ring 2 mode transit
Switch1(config)# erps 2 ring 2 primary interface eth-0-1
Switch1(config)# erps 2 ring 2 secondary interface eth-0-2
Switch1(config)# erps 2 mstp instance 2
Switch1(config)# erps 2 ring 2 enable
```

step 10 Configurring ring1 on transit node switch3

```
Switch3(config)# erps 1
Switch3(config)# erps 1 primary control vlan 10
Switch3(config)# erps 1 ring 1 level primary
Switch3(config)# erps 1 ring 1 mode transit
Switch3(config)# erps 1 ring 1 primary interface eth-0-1
Switch3(config)# erps 1 ring 1 secondary interface eth-0-2
Switch3(config)# erps 1 mstp instance 1
Switch3(config)# erps 1 ring 1 enable
```

step 11 Configurring ring2 on transit node switch3

```
Switch3(config)# erps 2
Switch3(config)# erps 2 primary control vlan 20
Switch3(config)# erps 2 ring 2 level primary
Switch3(config)# erps 2 ring 2 mode transit
Switch3(config)# erps 2 ring 2 primary interface eth-0-1
Switch3(config)# erps 2 ring 2 secondary interface eth-0-2
Switch3(config)# erps 2 mstp instance 2
Switch3(config)# erps 2 ring 2 enable
```

step 12 Configurring ring1 on transit node switch4

```
Switch4(config)# erps 1
Switch4(config)# erps 1 primary control vlan 10
Switch4(config)# erps 1 ring 1 level primary
Switch4(config)# erps 1 ring 1 mode transit
Switch4(config)# erps 1 ring 1 primary interface eth-0-1
Switch4(config)# erps 1 ring 1 secondary interface eth-0-2
```

```
Switch4(config)# erps 1 mstp instance 1
Switch4(config)# erps 1 ring 1 enable
```

step 13 Configurring ring2 on transit node switch4

```
Switch4(config)# erps 2
Switch4(config)# erps 2 primary control vlan 20
Switch4(config)# erps 2 ring 2 level primary
Switch4(config)# erps 2 ring 2 mode transit
Switch4(config)# erps 2 ring 2 primary interface eth-0-1
Switch4(config)# erps 2 ring 2 secondary interface eth-0-2
Switch4(config)# erps 2 mstp instance 2
Switch4(config)# erps 2 ring 2 enable
```

step 14 Exit configure mode

```
SwitchX(config)# end
```

17.7 Configuring Smart Link

17.7.1 Overview

Function Introduction

The Smart Link is a simple but practical technology of fast link protection. It is a solution specific to dual uplink networking to fulfill redundancy and fast migration of active and standby links.

Every smart-link group is included a pair of a layer 2 interfaces where one interface is configured to act as a standby to the other. The feature provides an alternative solution to the STP. Users can disable STP and still retain basic link redundancy. The feature also support load-balancing so than both interfaces simultaneously forward the traffic.

Principle Description

N/A

17.7.2 Configuration

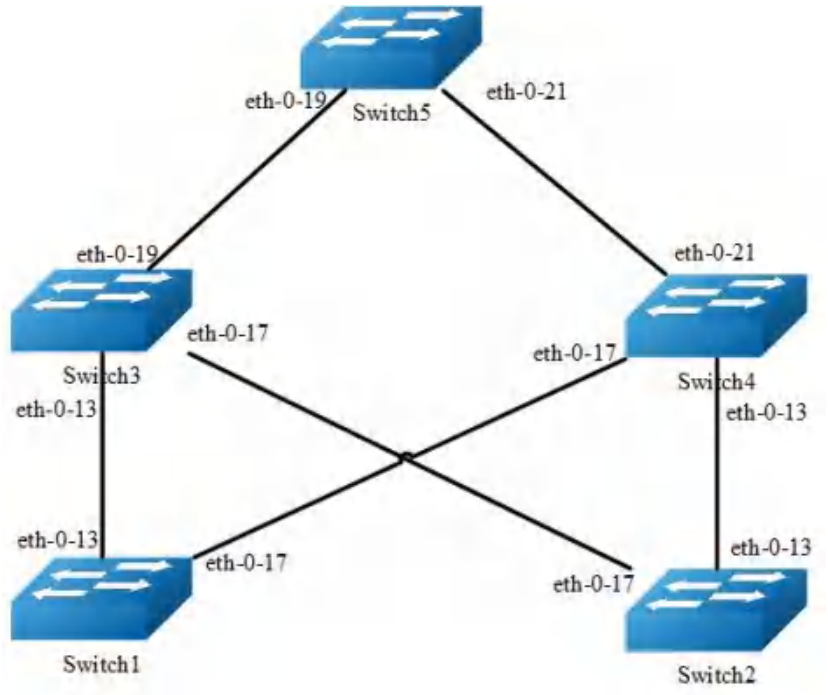


Figure 17-32 Smart-Link Typical Topology

The figure above is a typical smart-link application. The Switch1 and Switch2 are configured smart-link group. Switch3, Switch4 and Switch5 are configured smart-link flush receiver.

To configure smart-link group, some configuration should be configured before it.

- VLANs should be configured.
- MSTP instance should be configured.
- Spanning-tree should be disabled in the interface.
- About above configurations, please see the related references.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2-20
Switch(config-vlan)# exit
```

step 3 Set the spanning tree mode and create mstp instance

Create the mstp instance on Switch1 and Switch2:

```
Switch(config)# spanning-tree mode mstp
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 1
Switch(config-mst)# instance 2 vlan 2
Switch(config-mst)# instance 3 vlan 3
Switch(config-mst)# instance 10 vlan 10
Switch(config-mst)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1 and Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# spanning-tree port disable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# spanning-tree port disable
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch3 and Switch4:

```
Switch(config)# interface eth-0-13
Switch(config-if)# switchport mode trunk
Switch(config-if)# no shutdown
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# smart-link flush receive control-vlan 10 password simple test
```

```
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# no shutdown
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# smart-link flush receive control-vlan 10 password simple test
Switch (config-if)# exit
```

Interface eth-0-19 configuration for Switch3:

```
Switch(config)# interface eth-0-19
Switch(config-if)# switchport mode trunk
Switch(config-if)# no shutdown
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# exit
```

Interface eth-0-21 configuration for Switch4:

```
Switch(config)# interface eth-0-21
Switch(config-if)# switchport mode trunk
Switch(config-if)# no shutdown
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# exit
```

Interface configuration for Switch5:

```
Switch(config)# interface eth-0-19
Switch(config-if)# switchport mode trunk
Switch(config-if)# no shutdown
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# smart-link flush receive control-vlan 10 password simple test
Switch(config-if)# exit

Switch(config)# interface eth-0-21
Switch(config-if)# switchport mode trunk
Switch(config-if)# no shutdown
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# smart-ink flush receive control-vlan 10 password simple test
Switch(config-if)# exit
```

step 5 Create smart link group and set the attributes of the group

Create smart link group on Switch1 and Switch2:

```
Switch(config)# smart-link group 1
Switch(config-smlk-group)# interface eth-0-13 master
Switch(config-smlk-group)# interface eth-0-17 slave
Switch(config-smlk-group)# protected mstp instance 1
Switch(config-smlk-group)# protected mstp instance 2
Switch(config-smlk-group)# protected mstp instance 3
Switch(config-smlk-group)# protected mstp instance 10
Switch(config-smlk-group)# load-balance instance 3
Switch(config-smlk-group)# restore time 40
Switch(config-smlk-group)# restore enable
Switch(config-smlk-group)# flush send control-vlan 10 password simple test
Switch(config-smlk-group)# group enable
Switch(config-smlk-group)# exit
```

step 6 Disable the smart link relay function

Configure on Switch5:

```
Switch(config)# no smart-link relay enable
```

step 7 Exit the configure mode

```
Switch(config)# end
```

step 8 Validation

Display the result on Switch1.

```
Switch1# show smart-link group 1
Smart-link group 1 information:
 The smart-link group was enabled.
 ============================================================
 Auto-restore:
  state      time         count   Last-time
  enabled    40           0     N/A
 ============================================================
 Protected instance: 1  2  3
 Load balance instance: 3
 Flush sender , Control-vlan ID: 10   Password:test
 ============================================================
 INTERFACE:
 Role   Member    DownCount Last-Down-Time  FlushCount Last-Flush-Time
 MASTER eth-0-13   0      N/A              0    N/A
 SLAVE  eth-0-17   0      N/A              0    N/A
 ============================================================
 Instance states in the member interfaces:
  A - ACTIVE ,  B -BLOCK  , D-The interface is link-down
 Map-instance-ID   MASTER(eth-0-13)   SLAVE(eth-0-17)
     1          A             B
     2          A             B
     3          B             A
```

Display the result on Switch2.

```
Switch# show smart-link group 1
Smart-link group 1 information:
 The smart-link group was enabled.
 ============================================================
 Auto-restore:
  state      time         count   Last-time
  enabled    40           0     N/A
 ============================================================
 Protected instance: 1  2  3
 Load balance instance: 3
 Flush sender , Control-vlan ID: 10   Password:test
 ============================================================
 INTERFACE:
 Role   Member    DownCount Last-Down-Time  FlushCount Last-Flush-Time
 MASTER eth-0-13   0      N/A              0    N/A
 SLAVE  eth-0-17   0      N/A              0    N/A
 ============================================================
 Instance states in the member interfaces:
  A - ACTIVE ,  B -BLOCK  , D-The interface is link-down
```

```
Map-instance-ID  MASTER(eth-0-13)   SLAVE(eth-0-17)
   1         A           B
   2         A           B
   3         B           A
```

Display the result on Switch3.

```
Switch# show smart-link
 Relay smart-link flush packet is enabled
 Smart-link flush receiver interface:
   eth-0-13   control-vlan:10   password:test
   eth-0-17   control-vlan:10   password:test
Smart-link received flush packet number:0
Smart-link processed flush packet number:0
Smart link Group Number is 0.
```

Display the result on Switch4.

```
Switch# show smart-link
 Relay smart-link flush packet is enabled
 Smart-link flush receiver interface:
   eth-0-13   control-vlan:10   password:test
   eth-0-17   control-vlan:10   password:test
Smart-link received flush packet number:0
Smart-link processed flush packet number:0
Smart link Group Number is 0.
```

Display the result on Switch5.

```
Switch# show smart-link
 Relay smart-link flush packet is disabled
 Smart-link flush receiver interface:
   eth-0-21   control-vlan:10   password: test
   eth-0-19   control-vlan:10   password:test
Smart-link received flush packet number:0
Smart-link processed flush packet number:0
 Smart link Group Number is 0.
```

17.7.3 Application cases

N/A

17.8 Configuring Multi-Link

17.8.1 Overview

Function Introduction

The Multi-Link is a simple but practical technology of fast link protection. It is a solution specific to multi-uplink networking to fulfill redundancy and fast migration of between links.

The feature is like smart link, but links extend to four instead of two.

Principle Description

N/A

17.8.2 Configuration

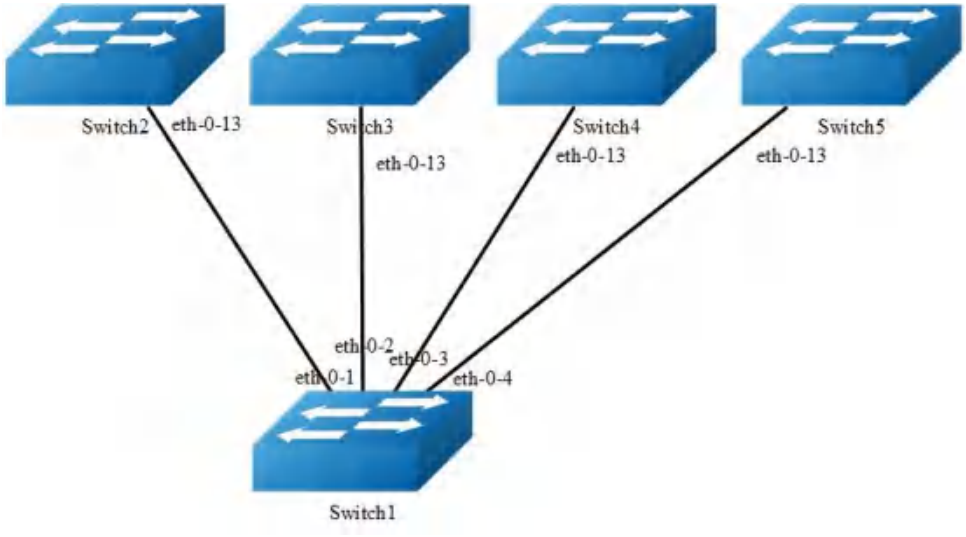


Figure 17-33 Multi-Link Typical Topology

The figure above is a typical multi-link application. The Switch1 are configured multi-link group. Switch2, Switch3, Switch4 and Switch5 are configured multi-link flush receiver.

To configure Multi-link group, some configuration should be configured before it.

- VLANs should be configured.
- MSTP instance should be configured.
- Spanning-tree should be disabled in the interface.
- About above configurations, please see the related references.
- The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config- vlan)# vlan 2-10
Switch(config- vlan)# exit
```

step 3 Set the spanning tree mode and create mstp instance

```
Switch(config)# spanning-tree mode mstp
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 1
Switch(config-mst)# instance 2 vlan 2
Switch(config-mst)# instance 3 vlan 3
Switch(config-mst)# instance 4 vlan 4-10
Switch(config-mst)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface range eth-0-1 - 4
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
```

```
Switch(config-if)# spanning-tree port disable
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch1 ~ 5:

```
Switch(config)# interface eth-0-13
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# multi-link flush receive control-vlan 10 password simple test
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 5 Create multi link group and set the attributes of the group

Create multi link group on Switch1:

```
Switch(config)# multi-link group 1
Switch(config-multilk-group)# interface eth-0-1 priority 1
Switch(config-multilk-group)# interface eth-0-2 priority 2
Switch(config-multilk-group)# interface eth-0-3 priority 3
Switch(config-multilk-group)# interface eth-0-4 priority 4
Switch(config-multilk-group)# protected mstp instance 1
Switch(config-multilk-group)# protected mstp instance 2
Switch(config-multilk-group)# protected mstp instance 3
Switch(config-multilk-group)# protected mstp instance 4
Switch(config-multilk-group)# load-balance instance 2 priority 2
Switch(config-multilk-group)# load-balance instance 3 priority 3
Switch(config-multilk-group)# load-balance instance 4 priority 4
Switch(config-multilk-group)# restore time 40
Switch(config-multilk-group)# restore enable
Switch(config-multilk-group)# flush send control-vlan 10 password simple test
Switch(config-multilk-group)# group enable
Switch(config-multilk-group)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1.

```
Switch# show multi-link group 1
Multi-link group 1 information:
 The multi-link group was enabled.
 ============================================================
 Auto-restore:
  state      time         count   Last-time
  enabled    40             0     N/A
 ============================================================
 Protected instance: 1  2  3  4
 Load balance instance: 2(to P2)  3(to P3)  4(to P4)
 Flush sender , Control-vlan ID: 10   Password:test
 ============================================================
 INTERFACE:
 Role   Member    DownCount Last-Down-Time     FlushCount Last-Flush-Time
 PRI1   eth-0-1   0         N/A               1          2016/09/05,07:13:24
 PRI2   eth-0-2   0         N/A               1          2016/09/05,07:13:24
```

```
PRI3  eth-0-3  0     N/A          1     2016/09/05,07:13:24
PRI4  eth-0-4  0     N/A          1     2016/09/05,07:13:24
============================================================
Instance states in the member interfaces:
 A - ACTIVE , B -BLOCK , D-The interface is link-down
Map-instance-ID  P1(eth-0-1 )  P2(eth-0-2 )  P3(eth-0-3 )  P4(eth-0-4 )
1          A       B       B       B
2          B       A       B       B
3          B       B       A       B
4          B       B       B       A
```

Display the result on Switch2~5.

```
Switch# show multi-link
Relay multi-link flush packet is enabled
Multi-link flush receiver interface:
  eth-0-13  control-vlan:10  password:test
Multi-link received flush packet number:0
Multi-link processed flush packet number:0
Multi-link tcn is disabled
Multi-link tcn query count   :2
Multi-link tcn query interval :10
Multi-link Group Number is 0.
```

17.8.3 Application cases

Configuring Multi-Link Enhance

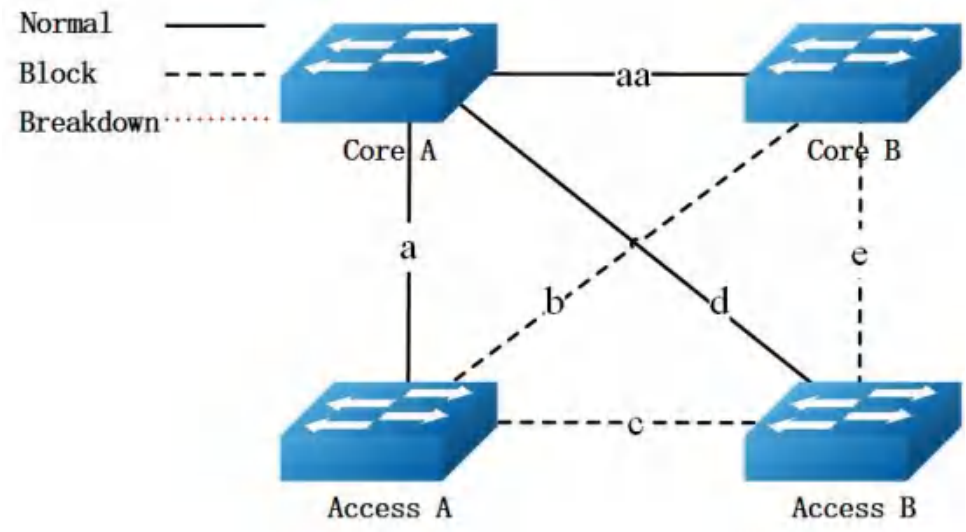
There is an enhanced method to improve the ability of multi-link to protect link. When all the interfaces of multi-link group are down, you can enable another interface to send the enhance packet to peer which makes the instance state of one interface to change from block to active. It would avoid the switch being the state of islet.

When 2 multi-link group on different switches backup for each other, multi-link members on one switch is blocked and can not protect the traffic.

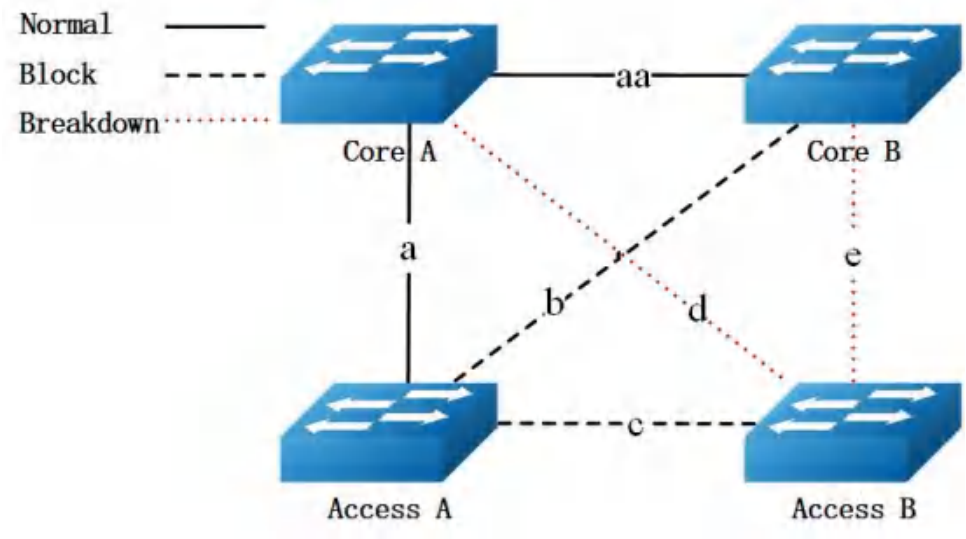
In this example:

- Core switch A and B, Access switch A and B, make up a full-match topology.
- Enable multi-link on Access switch A, the priority for link a/b/c is 1/2/3.
- Enable multi-link on Access switch B, the priority for link d/e is 1/2.

In normal condition, link b/c/e are block, link a/d are active. As the following figure shows:



When link d/e are break down, the only out going link for Access switch B is link c, which is between Access switch A and Access switch B.



Because link c is blocked, the Access switch B is the state of islet. As the following figure shows:

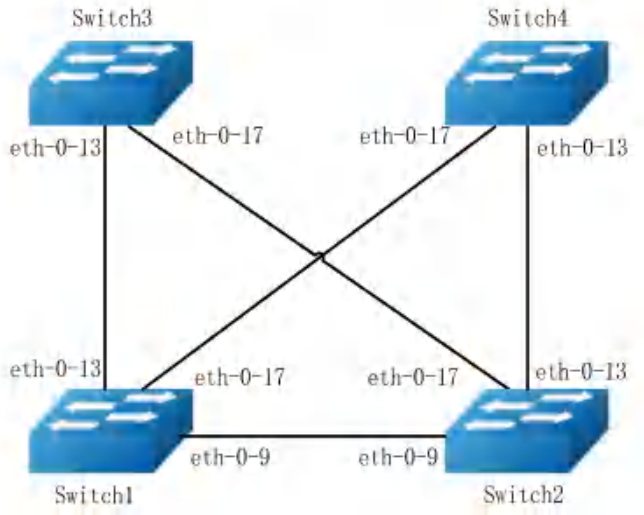


Figure 17-34 Multilink-enhance Typical Topology

The figure above is a typical multi-link application. The Switch1, 2 are configured multi-link group. Switch1 has the interface which receives the multilink-enhance packets. And , Switch2 has the interface which sends the multilink-enhance packets.

To configure multi-link group, some configuration should be configured before it.

- VLANs should be configured.
- MSTP instance should be configured.
- Spanning-tree should be disabled in the interface.
- About above configurations, please see the related references.
- It should configure the control vlan and password of flush sending before setting the multilink-enhance interface.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config- vlan)# vlan 10,20,30,40
Switch(config- vlan)# exit
```

step 3 Set the spanning tree mode and create mstp instance

```
Switch(config)# spanning-tree mode mstp
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 1,10,30
Switch(config-mst)# instance 2 vlan 20,40
Switch(config-mst)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch1(config)# interface eth-0-9
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan all
Switch1(config-if)# spanning-tree port disable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit

Switch1(config)# interface eth-0-13
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan all
Switch1(config-if)# spanning-tree port disable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit

Switch1(config)# interface eth-0-17
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan all
Switch1(config-if)# spanning-tree port disable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-13
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-17
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# multi-link flush receive control-vlan 30 password simple a
Switch(config-if)#exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-13
Switch(config-if)# multi-link flush receive control-vlan 30 password simple a
```

```
Switch(config-if)#exit

Switch(config)# interface eth-0-17
Switch(config-if)# multi-link flush receive control-vlan 30 password simple b
Switch(config-if)#exit
```

Interface configuration for Switch4:

```
Switch(config)# interface eth-0-13
Switch(config-if)# multi-link flush receive control-vlan 30 password simple b
Switch(config-if)#exit

Switch(config)# interface eth-0-17
Switch(config-if)# multi-link flush receive control-vlan 30 password simple a
Switch(config-if)#exit
```

step 5 Create multi link group and set the attributes of the group

Create multi link group on Switch1:

```
Switch(config)# multi-link group 1
Switch(config-multilk-group)# interface eth-0-13 priority 1
Switch(config-multilk-group)# interface eth-0-17 priority 2
Switch(config-multilk-group)# interface eth-0-9 priority 3
Switch(config-multilk-group)# protected mstp instance 1
Switch(config-multilk-group)# protected mstp instance 2
Switch(config-multilk-group)# flush send control-vlan 30 password simple a
Switch(config-multilk-group)# multilink-enhance receive control-vlan 10 password b interface eth-0-9
Switch(config-multilk-group)# group enable
Switch(config-multilk-group)# end
```

Create multi link group on Switch2:

```
Switch(config)# multi-link group 1
Switch(config-multilk-group)# interface eth-0-13 priority 1
Switch(config-multilk-group)# interface eth-0-17 priority 2
Switch(config-multilk-group)# protected mstp instance 1
Switch(config-multilk-group)# protected mstp instance 2
Switch(config-multilk-group)# flush send control-vlan 10 password simple b
Switch(config-multilk-group)# multilink-enhance interface eth-0-9
Switch(config-multilk-group)# group enable
Switch(config-multilk-group)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1.

```
Switch1# show multi-link group 1
Multi-link group 1 information:
The multi-link group was enabled.
============================================================
Auto-restore:
 state     time        count   Last-time
 disabled  60          0       N/A
============================================================
Protected instance: 1  2
```

```
Load balance instance:
Flush sender , Control-vlan ID: 30   Password: a
============================================================
INTERFACE:
Role  Member   DownCount Last-Down-Time     FlushCount Last-Flush-Time
PRI1  eth-0-13 0       N/A              5       2017/05/15,07:50:11
PRI2  eth-0-17 0       N/A              0       N/A
PRI3  eth-0-9  1       2017/05/15,07:48:46 5       2017/05/15,07:50:11
PRI4  N/A      0       N/A              0       N/A
============================================================
Instance states in the member interfaces:
 A-ACTIVE ,  B-BLOCK , A(E)-ENHANCE_ACTIVE   D-The interface is link-down
Map-instance-ID  P1(eth-0-13)   P2(eth-0-17)   P3(eth-0-9)    P4(N/A)
1            A           B           B           D
2            A           B           B           D

Switch# show multi-link
Relay multi-link flush packet is enabled
Multi-link enhance receiver interface:
  eth-0-9   control-vlan:10   password:b
Multi-link received flush packet number : 0
Multi-link processed flush packet number: 0
Multi-link received enhance packet number : 4
Multi-link processed enhance packet number: 4
Multi-link tcn is disabled
Multi-link tcn query count   : 2
Multi-link tcn query interval : 10
Multi-link Group Number is 1.
 Group-ID   State     Pri-1     Pri-2     Pri-3     Pri-4
 1          enabled   eth-0-13  eth-0-17  eth-0-9   N/A
```

Display the result on Switch2.

```
Switch# show multi-link group1
Multi-link group 1 information:
The multi-link group was enabled.
============================================================
Auto-restore:
 state     time         count    Last-time
 disabled  60           0        N/A
============================================================
Protected instance: 1  2
Load balance instance:
Flush sender , Control-vlan ID: 10   Password: b
Multilk enhance interface: eth-0-9, Control-vlan ID: 10  Password: b
============================================================
INTERFACE:
Role  Member   DownCount Last-Down-Time     FlushCount Last-Flush-Time
PRI1  eth-0-13 1       2017/05/15,07:49:15 0       N/A
PRI2  eth-0-17 2       2017/05/15,07:50:03 3       2017/05/15,07:50:11
PRI3  N/A      0       N/A              0       N/A
PRI4  N/A      0       N/A              0       N/A
============================================================
ENHANCE INTERFACE:
Role  Member   DownCount Last-Down-Time     EnhanceCount Last-SendEnhance-Ti
me
```

```
M-En  eth-0-9  0      N/A           0        N/A

============================================================
Instance states in the member interfaces:
 A-ACTIVE ,  B-BLOCK , A(E)-ENHANCE_ACTIVE  D-The interface is link-down
Map-instance-ID  P1(eth-0-13)  P2(eth-0-17)  P3(N/A)      P4(N/A)
1          A          B          D          D
2          A          B          D          D

Switch# show multi-link
Relay multi-link flush packet is enabled
Multi-link received flush packet number : 0
Multi-link processed flush packet number: 0
Multi-link received enhance packet number : 0
Multi-link processed enhance packet number: 0
Multi-link tcn is disabled
Multi-link tcn query count   : 2
Multi-link tcn query interval : 10
Multi-link Group Number is 1.
 Group-ID   State     Pri-1     Pri-2     Pri-3     Pri-4
 1        enabled    eth-0-13  eth-0-17  N/A       N/A
```

17.9 Configuring Monitor Link

17.9.1 Overview

Function Introduction

Monitor Link is a port collaboration function. Monitor Link usually works together with Layer 2 topology protocols. The idea is to monitor the states of uplink ports and adapt the up/down state of downlink ports to the up/down state of uplink ports, triggering link switchover on the downstream switch in time.

Principle Description

N/A

17.9.2 Configuration

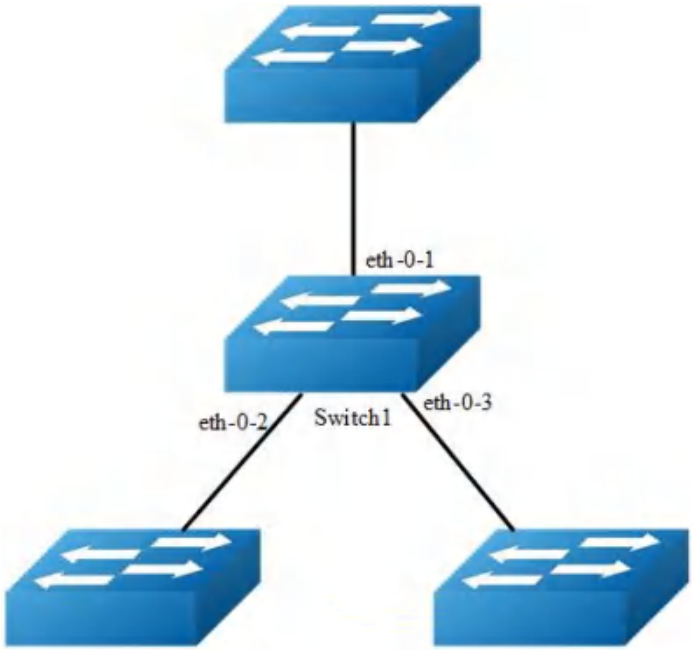


Figure 17-35 monitor link

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and turn on the interface

```
Switch(config)# interface range eth-0-1 - 3
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

step 3 Create multi link group and set the attributes of the group

```
Switch(config)# monitor-link group 1
Switch(config-mtlk-group)# monitor-link uplink interface eth-0-1
Switch(config-mtlk-group)# monitor-link downlink interface eth-0-2
Switch(config-mtlk-group)# monitor-link downlink interface eth-0-3
Switch(config-mtlk-group)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

```
Switch# show monitor-link group
Group Id: 1
Monitor link status: UP
Role    Member  Last-up-time    Last-down-time      upcount  downcount
UpLk 1  eth-0-1 2011/07/15,02:07:31 2011/07/15,02:07:31   2       1
DwLk 1  eth-0-2 2011/07/15,02:07:34 2011/07/15,02:07:31   1       1
DwLk 2  eth-0-3 N/A                 N/A                   0       0
```

17.9.3 Application cases

N/A

17.10 Configuring VRRP

17.10.1 Overview

Function Introduction

This chapter provides an overview of Virtual Router Redundancy Protocol (VRRP) and its implementation. VRRP eliminates the risk of a single point of failure inherent in a static default routing environment. It specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. One of the major advantages of VRRP is that it makes default path available without requiring configuration of dynamic routing on every end-host.


MD5 authentication is not yet supported for VRRP.

Principle Description

The VRRP module is based on: RFC 3768 (VRRP): Knight, S., et.al "Virtual Router Redundancy Protocol (VRRP)"

Terminology

- **Backup Router:** VRRP router that back up an IP address. It assumes forwarding responsibility for the virtual IP address if the Master fails.
- **Critical IP:** The IP address that the VRRP router send/receive messages on for a particular session.
- **IP Address Owner:** The VRRP Router that has the virtual router's IP address (es) as real interface address (es). This is the router that, when up, will respond to packets addressed to one of these IP addresses for ICMP pings, TCP connections, etc.
- **Master Router:** The VRRP router that owns the IP address (i.e., is being backed up), and which is the default router for forwarding for that IP address.
- **Virtual IP:** The IP address back up by a VRRP session.
- **Virtual Router:** A router managed by VRRP that acts as a default router for hosts on a shared LAN. It consists of a Virtual Router Identifier and a set of associated IP addresses across a common LAN. A VRRP Router might backup one or more virtual routers.
- **VRRP Router:** A router runs the Virtual Router Redundancy Protocol. It might participate in one or more virtual routers.

Typically, end hosts are connected to the enterprise network through a single router (first hop router) that is in the same Local Area Network (LAN) segment. The most popular method of configuration for the end hosts is to statically configure this router as their default gateway. This minimizes configuration and processing overhead. The main problem with this configuration method is that it produces a single point of failure if this first hop router fails.

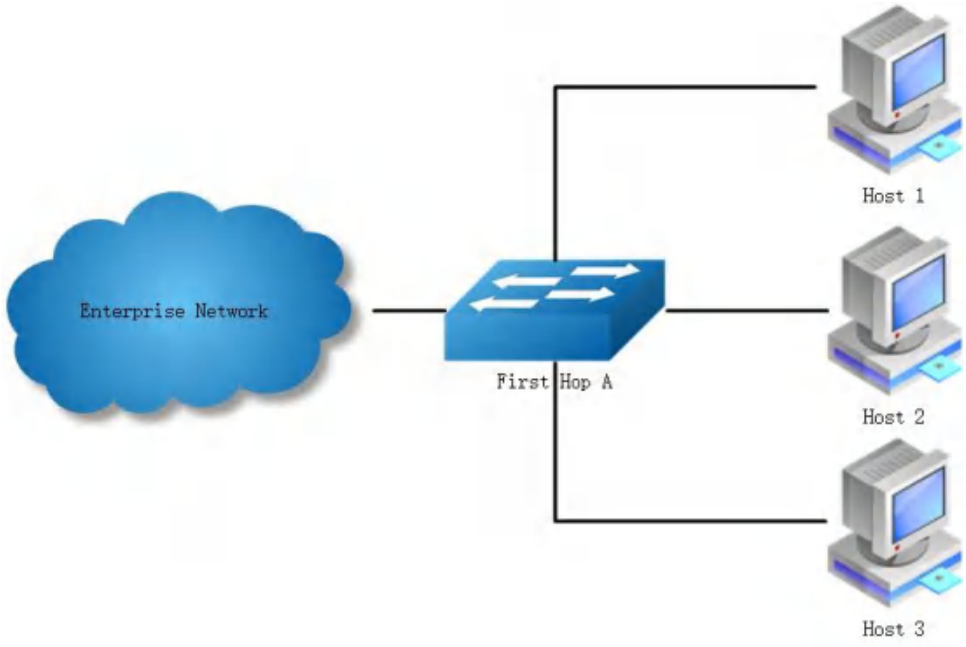


Figure 17-36 Without VRRP

The Virtual Router Redundancy Protocol attempts to solve this problem by introducing the concept of a virtual router, composed of two or more VRRP routers on the same subnet. The concept of a virtual IP address is also introduced, which is the address that end hosts configure as their default gateway. Only one router (called the master) forward packets on the behalf of this IP address. In the event that the Master router fails, one of the other routers (Backup) assumes forwarding responsibility for it.

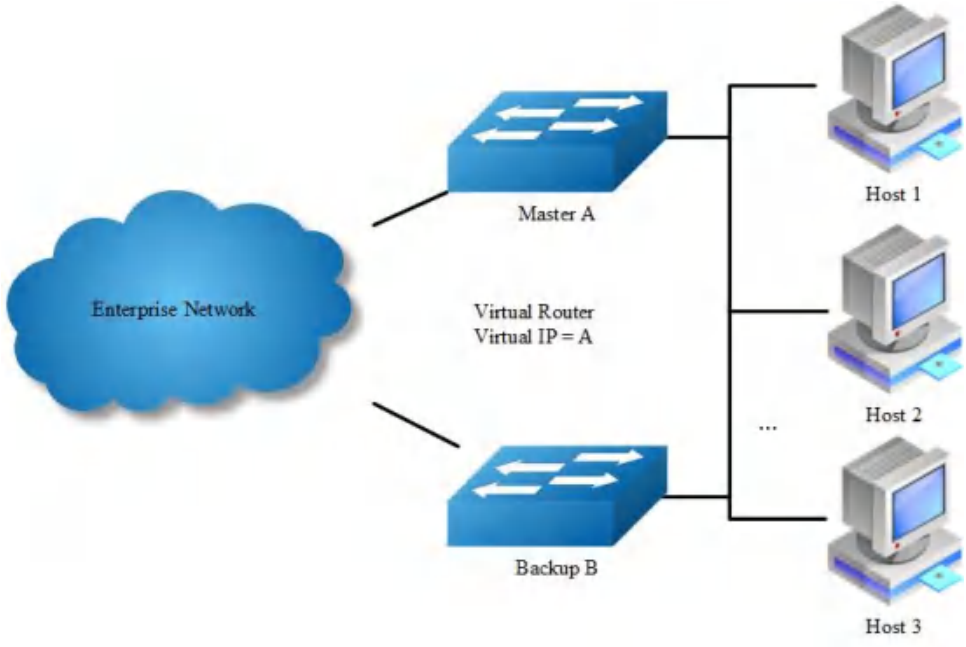


Figure 17-37 With VRRP

At first glance, the configuration outlined in might not seem very useful, as it doubles the cost and leaves one router idle at all times. This, however, can be avoided by creating two virtual routers and splitting the traffic between them.

17.10.2 Configuration

Configuring VRRP (One Virtual Router)

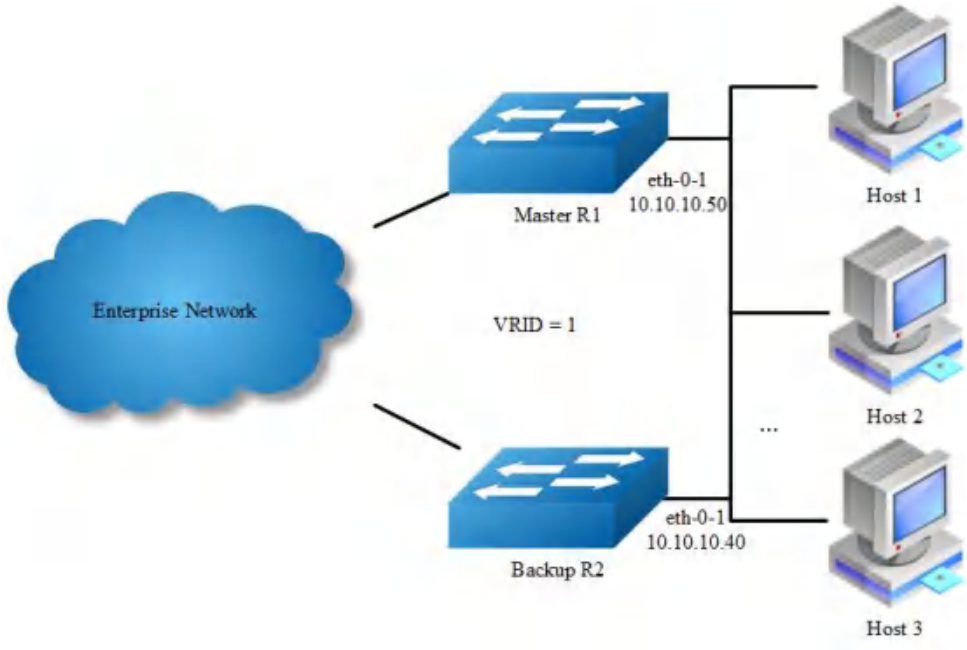


Figure 17-38 VRRP with one virtual router

In this configuration the end-hosts install a default route to the IP address of virtual router 1(VRID = 1) and both routers R1 and R2 run VRRP. R1 is configured to be the Master for virtual router 1 (VRID = 1) and R2 as a Backup for virtual router 1. If R1 fails, R2 will take over virtual router 1 and its IP addresses, and provide uninterrupted service for the hosts. Configuring only one virtual router, doubles the cost and leaves R2 idle at all times.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.50/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.40/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an instance of vrrp

```
Switch(config)# router vrrp 1
Switch(config-router)# virtual-ip 10.10.10.60
Switch(config-router)# interface eth-0-1
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
```

step 4 Set the priority (optional)

Set the priority on R1. R1 use the default value if the priority is not configured.

```
Switch(config-router)# priority 200
```

step 5 Enable vrrp and Exit the vrrp configure mode

```
Switch(config-router)# enable
Switch(config-router)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on R1.

```
Switch# show vrrp
vrrp session count: 1
vrrp version     : 2
VRID <1>
 State           : Master
 Virtual IP      : 10.10.10.60(Not IP owner)
 Interface       : eth-0-1
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Uniform-mac     : -
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
 Conf pri        : 200        Run pri   : 200
 Master router ip   : 10.10.10.50
 Master priority    : 200
 Master advt timer  : 5 second(s)
 Master down timer  : 15 second(s)
 Preempt delay      : 0 second(s)
 Learn master mode  : FALSE
```

Display the result on R2.

```
Switch# show vrrp
vrrp session count: 1
vrrp version     : 2
VRID <1>
 State           : Backup
 Virtual IP      : 10.10.10.60(Not IP owner)
 Interface       : eth-0-1
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Uniform-mac     : -
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
 Conf pri        : 100        Run pri   : 100
 Master router ip   : 10.10.10.50
 Master priority    : 200
 Master advt timer  : 5 second(s)
 Master down timer  : 16 second(s)
```

```
Preempt delay     : 0 second(s)
Learn master mode  : FALSE
```

Configuring VRRP (Two Virtual Router)

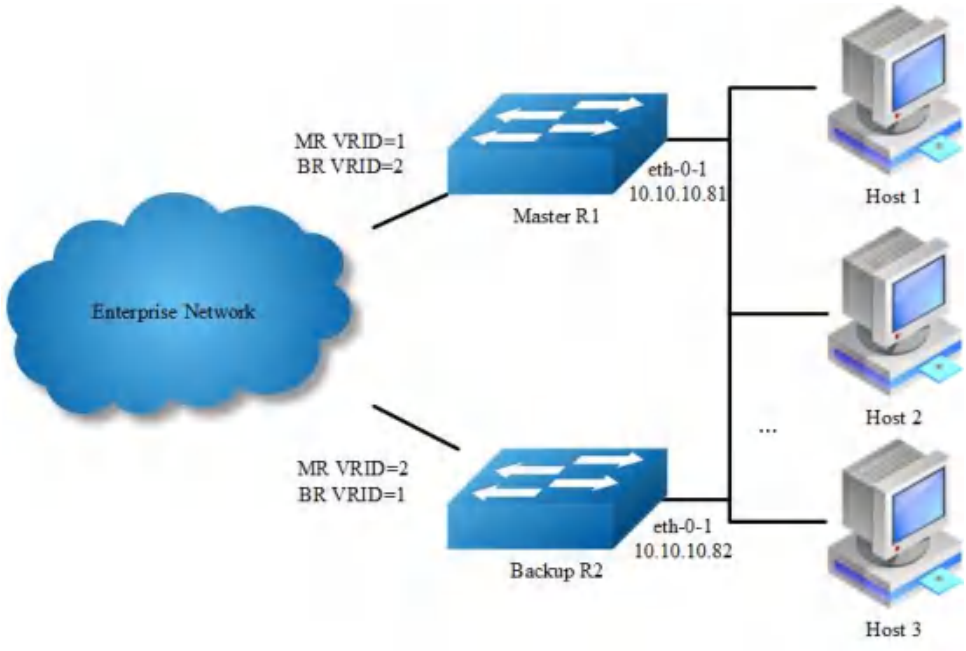


Figure 17-39 VRRP with two virtual router

In the one virtual router example earlier, R2 is not backed up by R1. This example illustrates how to backup R2 by configuring a second virtual router.

In this configuration, R1 and R2 are two virtual routers and the hosts split their traffic between R1 and R2. R1 and R2 function as backups for each other.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.81/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.82/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an instance of vrrp

Configuring R1:

```
Switch(config)# router vrrp 1
Switch(config-router)# virtual-ip 10.10.10.81
Switch(config-router)# interface eth-0-1
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# enable
Switch(config-router)# exit

Switch(config)# router vrrp 2
Switch(config-router)# virtual-ip 10.10.10.82
Switch(config-router)# interface eth-0-1
Switch(config-router)# priority 200
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# enable
Switch(config-router)# exit
```

Configuring R2:

```
Switch(config)# router vrrp 1
Switch(config-router)# virtual-ip 10.10.10.81
Switch(config-router)# interface eth-0-1
Switch(config-router)# priority 200
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# enable
Switch(config-router)# exit

Switch(config)# router vrrp 2
Switch(config-router)# virtual-ip 10.10.10.82
Switch(config-router)# interface eth-0-1
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# enable
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Display the result on R1.

```
Switch# show vrrp
vrrp session count: 2
vrrp version     : 2
VRID <1>
 State           : Master
 Virtual IP      : 10.10.10.81(IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Uniform-mac     : -
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
```

```
 Conf pri         : 100         Run pri  : 255
 Master router ip   : 10.10.10.81
 Master priority    : 255
 Master advt timer  : 5 second(s)
 Master down timer  : 15 second(s)
 Preempt delay      : 0 second(s)
 Learn master mode  : FALSE
VRID <2>
 State            : Backup
 Virtual IP       : 10.10.10.82(Not IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0102
 VRF              : Default
 Uniform-mac      : -
 Advt timer       : 5 second(s)
 Preempt mode     : TRUE
 Conf pri         : 200         Run pri  : 200
 Master router ip   : 10.10.10.82
 Master priority    : 255
 Master advt timer  : 5 second(s)
 Master down timer  : 15 second(s)
 Preempt delay      : 0 second(s)
 Learn master mode  : FALSE
```

Display the result on R2.

```
Switch# show vrrp
vrrp session count: 2
vrrp version     : 2
VRID <1>
 State            : Backup
 Virtual IP       : 10.10.10.81(Not IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0101
 VRF              : Default
 Uniform-mac      : -
 Advt timer       : 5 second(s)
 Preempt mode     : TRUE
 Conf pri         : 200         Run pri  : 200
 Master router ip   : 10.10.10.81
 Master priority    : 255
 Master advt timer  : 5 second(s)
 Master down timer  : 15 second(s)
 Preempt delay      : 0 second(s)
 Learn master mode  : FALSE
VRID <2>
 State            : Master
 Virtual IP       : 10.10.10.82(IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0102
 VRF              : Default
 Uniform-mac      : -
 Advt timer       : 5 second(s)
 Preempt mode     : TRUE
 Conf pri         : 100         Run pri  : 255
 Master router ip   : 10.10.10.82
```

```
Master priority    : 255
Master advt timer  : 5 second(s)
Master down timer  : 15 second(s)
Preempt delay      : 0 second(s)
Learn master mode  : FALSE
```

VRRP Circuit Failover

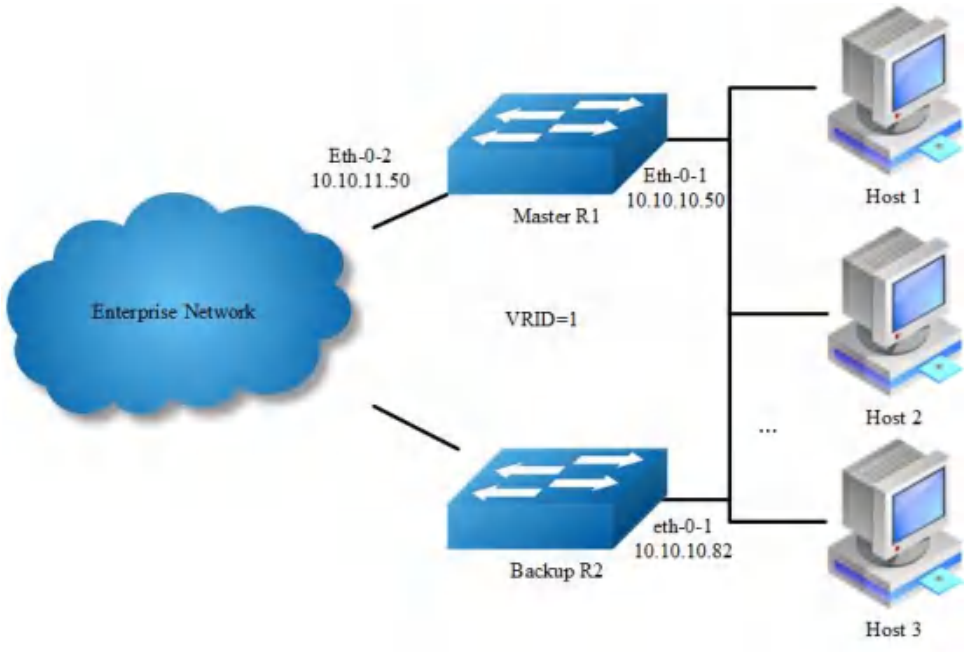


Figure 17-40 VRRP Circuit Failover

The need for VRRP Circuit Failover arose because VRRPv2 was unable to track the gateway interface status. The VRRP Circuit Failover feature provides a dynamic failover of an entire circuit in the event that one member of the group fails. It introduces the concept of a circuit, where two or more Virtual Routers on a single system can be grouped. In the event that a failure occurs and one of the Virtual Routers performs the Master to Backup transition, the other Virtual Routers in the group are notified and are forced into the Master to Backup transition, so that both incoming and outgoing packets are routed through the same gateway router, eliminating the problem for Firewall/NAT environments. The following scenario explains this feature.

To configure VRRP Circuit Failover, each circuit is configured to have a corresponding priority-delta value, which is passed to VRRP when a failure occurs. The priority of each Virtual Router on the circuit is decremented by the priority delta value causing the VR Master to VR Backup transition.

In this example, two routers R1 and R2 are configured as backup routers with different priorities. The priority-delta value is configured to be greater than the difference of both the priorities. R1 is configured to have a priority of 100 and R2 has a priority of 90. R1 with a greater priority is the Virtual Router Master. The priority-delta value is 20, greater than 10 (100 minus 90). On R1 when the external interface eth1 fails, the priority of R1 becomes 80 (100 minus 20). Since R2 has a greater priority (90) than R1, R2 becomes the VR Master and routing of packages continues without interruption.

When this VR Backup (R1) is up again, it regains its original priority (100) and becomes the VR Master again.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.50/24
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.11.50/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.40/24
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an track object to monitor the link state

Configuring R1:

```
Switch(config)# track 10 interface eth-0-2 linkstate
```

To get more information about track, please reference to the "Configuring Track" chapter.

step 4 Create an instance of vrrp

Configuring R1:

```
Switch(config)# router vrrp 1
Switch(config-router)# virtual-ip 10.10.10.1
Switch(config-router)# interface eth-0-1
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# priority 100
Switch(config-router)# track 10 decrement 20
Switch(config-router)# enable
```

Configuring R2:

```
Switch(config)# router vrrp 1
Switch(config-router)# virtual-ip 10.10.10.1
Switch(config-router)# interface eth-0-1
Switch(config-router)# preempt-mode true
Switch(config-router)# advertisement-interval 5
Switch(config-router)# priority 90
Switch(config-router)# enable
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on R1.

```
Switch# show vrrp
vrrp session count: 1
```

```
vrrp version     : 2
VRID <1>
 State           : Master
 Virtual IP      : 10.10.10.1(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Uniform-mac     : -
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
 Conf pri        : 100        Run pri  : 100
 Track Object    : 10         Decre pri : 20
 Decre pri       : 20
 Master router ip  : 10.10.10.50
 Master priority   : 100
 Master advt timer  : 5 second(s)
 Master down timer  : 16 second(s)
 Preempt delay     : 0 second(s)
 Learn master mode  : FALSE
```

Display the result on R2.

```
Switch# show vrrp
vrrp session count: 1
vrrp version     : 2
VRID <1>
 State           : Backup
 Virtual IP      : 10.10.10.1(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Uniform-mac     : -
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
 Conf pri        : 90         Run pri  : 90
 Master router ip  : 10.10.10.50
 Master priority   : 100
 Master advt timer  : 5 second(s)
 Master down timer  : 16 second(s)
 Preempt delay     : 0 second(s)
 Learn master mode  : FALSE
```

Configuring IPv6 VRRP(One Virtual Router)

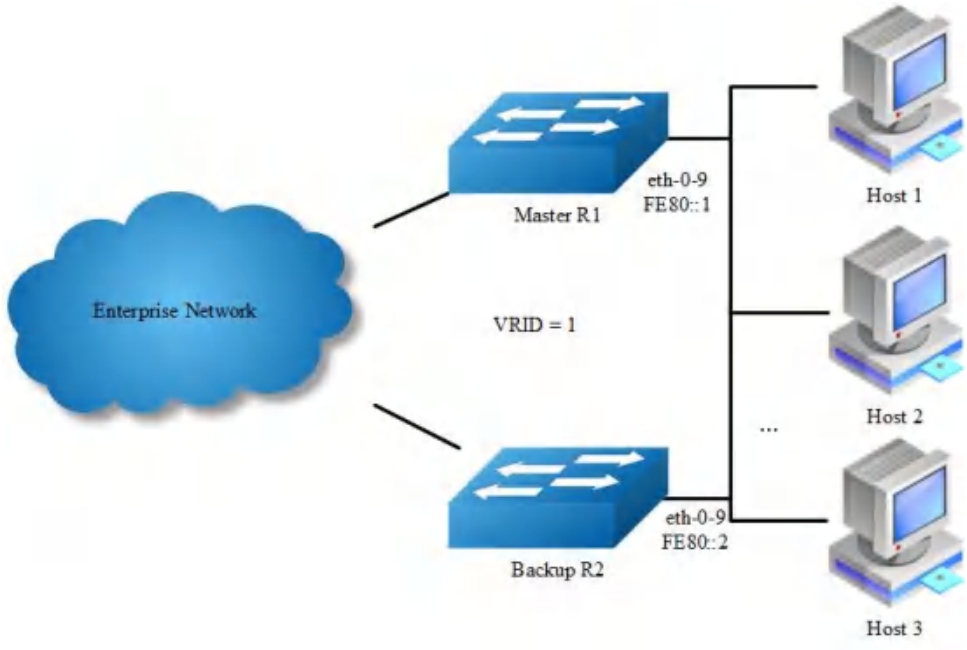


Figure 17-41 IPv6 VRRP with one virtual router

In this configuration the end-hosts install a default route to the IP address of virtual router 1(VRID = 1) and both routers R1 and R2 run VRRP. R1 is configured to be the Master for virtual router 1 (VRID = 1) and R2 as a Backup for virtual router 1. If R1 fails, R2 will take over virtual router 1 and its IP addresses, and provide uninterrupted service for the hosts. Configuring only one virtual router, doubles the cost and leaves R2 idle at all times.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode and enable IPv6

```
Switch# configure terminal
Switch(config)# ipv6 enable
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address fe80::1 link-local
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address fe80::2 link-local
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an instance of IPv6 VRRP

```
Switch(config)# router ipv6 vrrp 1
Switch(config-router)# virtual-ipv6 fe80::1 link-local
Switch(config-router)# interface eth-0-9
```

step 4 Set the priority (optional)

Set the priority on R1. R2 use the default value 100 if the priority is not configured.

```
Switch(config-router)# priority 200
```

step 5 Enable IPv6 VRRP and Exit the router configure mode

```
Switch(config-router)# enable
Switch(config-router)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on R1.

```
Switch# show ipv6 vrrp 1
VRID <1>
 State           : Master
 Virtual IP      : fe80::1(IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0201
 VRF             : Default
 Advt timer      : 1 second(s)
 Preempt mode    : TRUE
 Conf pri        : 200        Run pri  : 255
 Master router ip  : fe80::1
 Master priority   : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 3 second(s)
 Preempt delay     : 0 second(s)
```

Display the result on R2.

```
Switch# show ipv6 vrrp 1
VRID <1>
 State           : Backup
 Virtual IP      : fe80::1(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0201
 VRF             : Default
 Advt timer      : 1 second(s)
 Preempt mode    : TRUE
 Conf pri        : 100        Run pri  : 100
 Master router ip  : fe80::1
 Master priority   : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 4 second(s)
 Preempt delay     : 0 second(s)
```

Configuring IPv6 VRRP (Two Virtual Router)

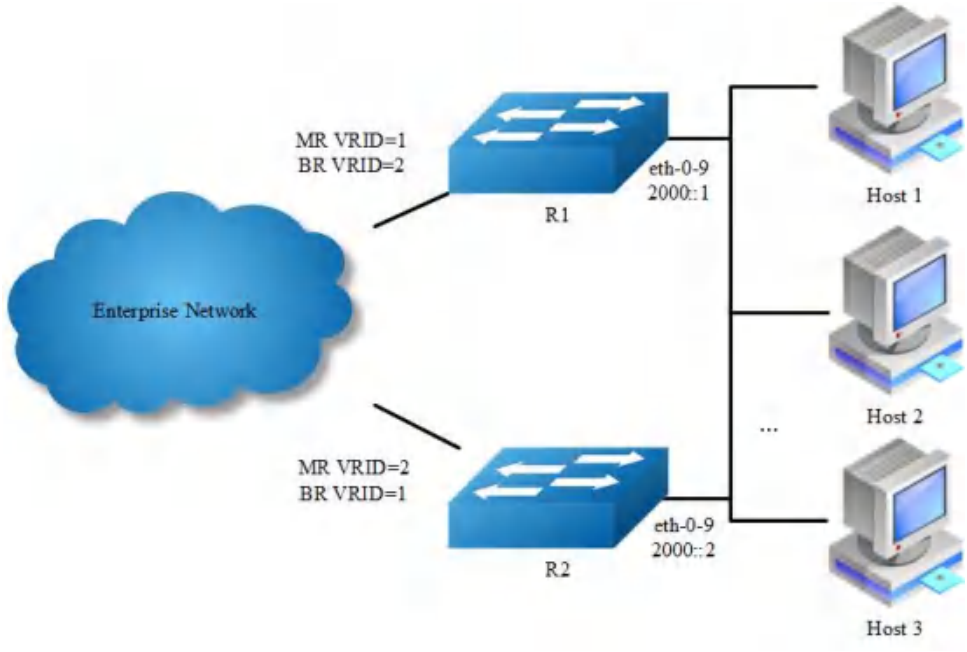


Figure 17-42 IPv6 VRRP with two virtual router

In the one virtual router example earlier, R2 is not backed up by R1. This example illustrates how to backup R2 by configuring a second virtual router.

In this configuration, R1 and R2 are two virtual routers and the hosts split their traffic between R1 and R2. R1 and R2 function as backups for each other.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode and enable IPv6

```
Switch# configure terminal
Switch(config)# ipv6 enable
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2000::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2000::2/64
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an instance of IPv6 VRRP

Configuring R1:

```
Switch(config)# router ipv6 vrrp 1
Switch(config-router)# interface eth-0-9
```

```
Switch(config-router)# priority 200
Switch(config-router)# virtual-ipv6 fe80::1 link-local
Switch(config-router)# virtual-ipv6 2000::1
Switch(config-router)# enable
Switch(config-router)# exit

Switch(config)# router ipv6 vrrp 2
Switch(config-router)# interface eth-0-9
Switch(config-router)# virtual-ipv6 fe80::3 link-local
Switch(config-router)# virtual-ipv6 2000::2
Switch(config-router)# enable
Switch(config-router)# exit
```

Configuring R2:

```
Switch(config)# router ipv6 vrrp 1
Switch(config-router)# interface eth-0-9
Switch(config-router)# virtual-ipv6 fe80::1 link-local
Switch(config-router)# virtual-ipv6 2000::1
Switch(config-router)# enable
Switch(config-router)# exit

Switch(config)# router vrrp 2
Switch(config-router)# interface eth-0-9
Switch(config-router)# priority 200
Switch(config-router)# virtual-ipv6 fe80::3 link-local
Switch(config-router)# virtual-ipv6 2000::2
Switch(config-router)# enable
Switch(config-router)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Display the result on R1.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 2
VRID <1>
 State           : Master
 Virtual IP      : fe80::1(Not IP owner)
                   2000::1(IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0201
 VRF             : Default
 Advt timer      : 1 second(s)
 Preempt mode    : TRUE
 Conf pri        : 200         Run pri  : 255
 Master router ip  : fe80::48cc:69ff:fec8:5b00
 Master priority   : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 3 second(s)
 Preempt delay     : 0 second(s)
VRID <2>
 State           : Backup
 Virtual IP      : fe80::3(Not IP owner)
```

```
                2000::2(Not IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0202
 VRF              : Default
 Advt timer       : 1 second(s)
 Preempt mode     : TRUE
 Conf pri         : 100          Run pri   : 100
 Master router ip   : fe80::b002:86ff:febc:3700
 Master priority    : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 4 second(s)
 Preempt delay      : 0 second(s)
```

Display the result on R2.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 2
VRID <1>
 State            : Backup
 Virtual IP       : fe80::1(Not IP owner)
                2000::1(Not IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0201
 VRF              : Default
 Advt timer       : 1 second(s)
 Preempt mode     : TRUE
 Conf pri         : 100          Run pri   : 100
 Master router ip   : fe80::48cc:69ff:fec8:5b00
 Master priority    : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 4 second(s)
 Preempt delay      : 0 second(s)
VRID <2>
 State            : Master
 Virtual IP       : fe80::3(Not IP owner)
                2000::2(IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0202
 VRF              : Default
 Advt timer       : 1 second(s)
 Preempt mode     : TRUE
 Conf pri         : 200          Run pri   : 255
 Master router ip   : fe80::b002:86ff:febc:3700
 Master priority    : 255
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 3 second(s)
 Preempt delay      : 0 second(s)
```

IPv6 VRRP Circuit Failover

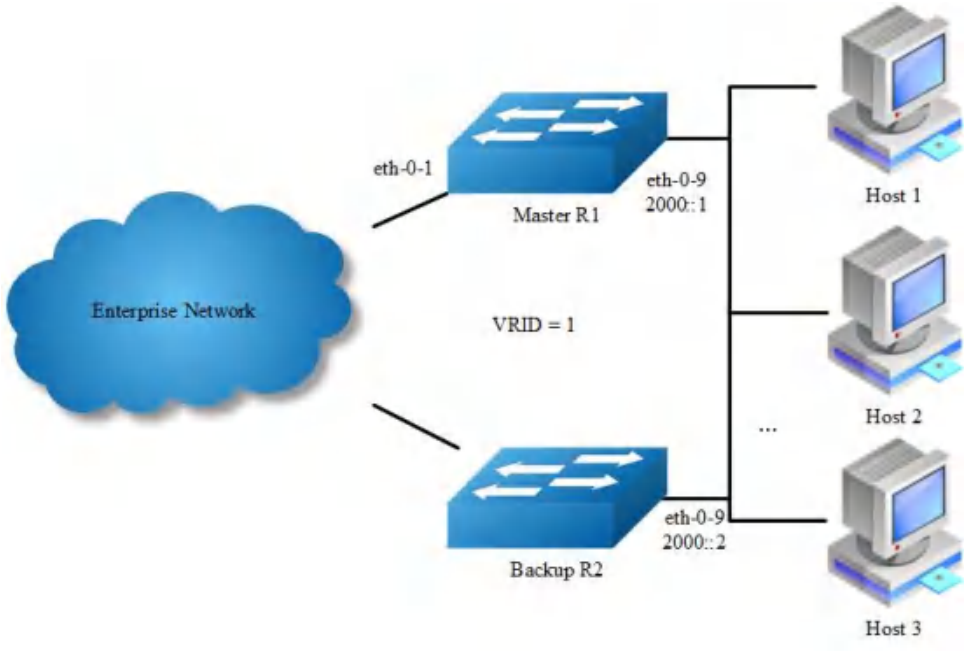


Figure 17-43 IPv6 VRRP Circuit Failover

The need for VRRP Circuit Failover arose because VRRPv2 was unable to track the gateway interface status. The VRRP Circuit Failover feature provides a dynamic failover of an entire circuit in the event that one member of the group fails. It introduces the concept of a circuit, where two or more Virtual Routers on a single system can be grouped. In the event that a failure occurs and one of the Virtual Routers performs the Master to Backup transition, the other Virtual Routers in the group are notified and are forced into the Master to Backup transition, so that both incoming and outgoing packets are routed through the same gateway router, eliminating the problem for Firewall/NAT environments. The following scenario explains this feature.

To configure VRRP Circuit Failover, each circuit is configured to have a corresponding priority-delta value, which is passed to VRRP when a failure occurs. The priority of each Virtual Router on the circuit is decremented by the priority delta value causing the VR Master to VR Backup transition.

In this example, two routers R1 and R2 are configured as backup routers with different priorities. The priority-delta value is configured to be greater than the difference of both the priorities. R1 is configured to have a priority of 100 and R2 has a priority of 90. R1 with a greater priority is the Virtual Router Master. The priority-delta value is 20, greater than 10 (100 minus 90). On R1 when the external interface eth1 fails, the priority of R1 becomes 80 (100 minus 20). Since R2 has a greater priority (90) than R1, R2 becomes the VR Master and routing of packages continues without interruption.

When this VR Backup (R1) is up again, it regains its original priority (100) and becomes the VR Master again.

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode and enable IPv6

```
Switch# configure terminal
Switch(config)# ipv6 enable
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for R1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2000::1/64
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-1
```

```
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for R2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2000::2/64
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 3 Create an track object to monitor the link state

Configuring R1:

```
Switch(config)# track 10 interface eth-0-1 linkstate
```

To get more information about track, please reference to the "Configuring Track" chapter.

step 4 Create an instance of IPv6 VRRP

Configuring R1:

```
Switch(config)# router ipv6 vrrp 1
Switch(config-router)# interface eth-0-9
Switch(config-router)# virtual-ipv6 fe80::1 link-local
Switch(config-router)# virtual-ipv6 2000::3
Switch(config-router)# track 10 decrement 20
Switch(config-router)# enable
```

Configuring R2:

```
Switch(config)# router ipv6 vrrp 1
Switch(config-router)# interface eth-0-9
Switch(config-router)# priority 90
Switch(config-router)# virtual-ipv6 fe80::1 link-local
Switch(config-router)# virtual-ipv6 2000::3
Switch(config-router)# enable
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on R1.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 1
VRID <1>
 State            : Master
 Virtual IP       : fe80::1(Not IP owner)
                    2000::3(Not IP owner)
 Interface        : eth-0-9
 VMAC             : 0000.5e00.0201
 VRF              : Default
 Advt timer       : 1 second(s)
 Preempt mode     : TRUE
 Conf pri         : 100          Run pri   : 100
 Track Object     : 10           Decre pri : 20
 Master router ip   : fe80::48cc:69ff:fec8:5b00
```

```
Master priority    : 100
Master advt timer  : 100 centi-second(s)
Master down timer  : 4 second(s)
Preempt delay      : 0 second(s)
```

Display the result on R2.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 1
VRID <1>
 State           : Backup
 Virtual IP      : fe80::1(Not IP owner)
                   2000::3(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0201
 VRF             : Default
 Advt timer      : 1 second(s)
 Preempt mode    : TRUE
 Conf pri        : 90          Run pri   : 90
 Master router ip   : fe80::48cc:69ff:fec8:5b00
 Master priority    : 100
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 4 second(s)
 Preempt delay      : 0 second(s)
```

step 7 Shutdown port eth-0-1 of R1, Exit the configure mode

```
Switch# configure terminal
Switch(config)# interface eth-0-1
Switch(config-if)# shutdown
Switch(config-if)# end
```

step 8 Validation, R1 change to backup and R2 change to master

Display the result on R1.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 1
VRID <1>
 State           : Backup
 Virtual IP      : fe80::1(Not IP owner)
                   2000::3(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0201
 VRF             : Default
 Advt timer      : 1 second(s)
 Preempt mode    : TRUE
 Conf pri        : 100         Run pri   : 80
 Track Object    : 10          Decre pri : 20
 Master router ip   : fe80::b002:86ff:febc:3700
 Master priority    : 90
 Master advt timer  : 100 centi-second(s)
 Master down timer  : 4 second(s)
 Preempt delay      : 0 second(s)
```

Display the result on R2.

```
Switch# show ipv6 vrrp
IPv6 vrrp session count: 1
```

```
VRID <1>
State          : Master
Virtual IP     : fe80::1(Not IP owner)
                 2000::3(Not IP owner)
Interface      : eth-0-9
VMAC           : 0000.5e00.0201
VRF            : Default
Advt timer     : 1 second(s)
Preempt mode   : TRUE
Conf pri       : 90        Run pri  : 90
Master router ip  : fe80::b002:86ff:febc:3700
Master priority   : 90
Master advt timer : 100 centi-second(s)
Master down timer : 4 second(s)
Preempt delay     : 0 second(s)
```

17.10.3 Application cases

N/A

17.11 Configuring Track

17.11.1 Overview

Function Introduction

Track is used for link the functional modules and monitor modules. Track builds a system structure with 3 levels: "functional modules – Track – monitor modules".

Track can shield the difference of the monitor modules and provide an unitized API for the functional modules.

The following monitor modules are supported:

- IP SLA
- interface states
- bfd states

The following functional modules are supported:

- Static route
- VRRP

Track makes a communication for the functional modules and monitor modules. When link states or network performance is changed, the monitor modules can detect the event and notify the track module; therefore track will change its owner states and notify the related functional modules.

Principle Description

N/A

17.11.2 Configuration

Configuring IP SLA for interfaces in the VRF

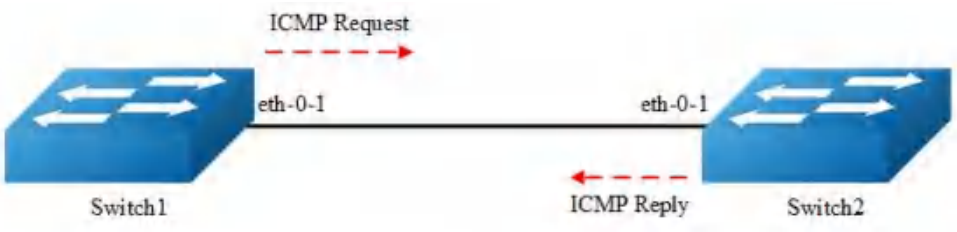


Figure 17-44 IP SLA

IP SLA (Service Level Agreement) is a network performance measurement and diagnostics tool that uses active monitoring. Active monitoring is the generation of traffic in a reliable and predictable manner to measure network performance. Every IP SLA operation maintains an operation return-code value. This return code is interpreted by the tracking process. The return code can return OK, Over Threshold, and several other return codes. Different operations can have different return-code values, so only values common to all operation types are used. In IP SLA, use icmp echo to check state or reachability of a route.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create a vrf instance

```
Switch(config)# ip vrf vpn1
Switch(config-vrf)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip vrf forwarding vpn1
Switch(config-if)# ip address 192.168.0.2/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip vrf forwarding vpn1
Switch(config-if)# ip address 192.168.0.1/24
Switch(config-if)# exit
```

step 4 Create ip sla and set the attributes

Configuring Switch1:

```
Switch(config)# ip sla monitor 1
Switch(config-ipsla)# type icmp-echo 192.168.0.1
Switch(config-ipsla)# frequency 35
Switch(config-ipsla)# timeout 6
Switch(config-ipsla)# threshold 3000
Switch(config-ipsla)# ttl 65
Switch(config-ipsla)# tos 1
Switch(config-ipsla)# data-size 29
Switch(config-ipsla)# data-pattern abababab
Switch(config-ipsla)# fail-percent 90
Switch(config-ipsla)# packets-per-test 4
Switch(config-ipsla)# interval 9
Switch(config-ipsla)# statistics packet 10
Switch(config-ipsla)# statistics test 3
Switch(config-ipsla)# vrf vpn1
Switch(config-ipsla)# exit
```

Parameters for ip sla:

- frequency:Time between 2 probes. Valid range is 1-4800 second, default value is 60 seconds.
- timeout:Timeout value for icmp reply. Valid range is 1-4800 second, default value is 5 seconds.
- threshold: Timeout value for icmp threshold. Valid range is 1-4800000 millisecond, default value is 5000 millisecond.
- packets-per-test:Packet number for each probe. Valid range is 1-10, default value is 3.
- interval:Time between 2 packets. Valid range is 1-4800 second, default value is 6 seconds.
- statistics packet:Packet number for statistics. Valid range is 0-1000, default value is 50.
- statistics test probe number for statistics. Valid range is 0-10, default value is 5

step 5 Enable ip sla

Configuring Switch1:

```
Switch(config)# ip sla monitor schedule 1
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1.

```
Switch# sho ip sla monitor 1
Entry 1
Type              : Echo
Admin state       : Disable
Destination address  : 192.168.0.1
Frequency         : 35s
Timeout           : 6s
Threshold         : 3000ms
Interval          : 9s
Packet per test     : 4
TTL               : 65
TOS               : 1
Data Size         : 29 bytes
Fail Percent      : 90%
Packet Item Cnt     : 10
Test Item Cnt       : 3
Vrf               : vpn1
Return code       : Unknown
```

Configuring IP SLA for Layer3 interfaces

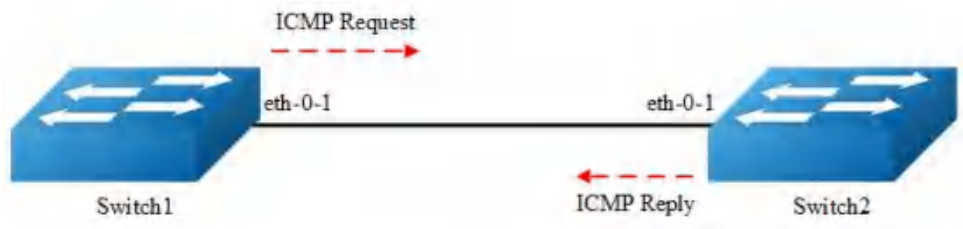


Figure 17-45 IP SLA

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.2/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.1/24
Switch(config-if)# exit
```

step 3 Create ip sla and set the attributes

Configuring Switch1:

```
Switch(config)# ip sla monitor 1
Switch(config-ipsla)# type icmp-echo  192.168.0.1
Switch(config-ipsla)# frequency 10
Switch(config-ipsla)# timeout 5
Switch(config-ipsla)# exit
```

step 4 Enable ip sla

Configuring Switch1:

```
Switch(config)# ip sla monitor schedule 1
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch# show ip sla monitor
Entry 1
  Type                : Echo
  Admin state         : Enable
  Destination address : 192.168.0.1
  Frequency           : 10 seconds
  Timeout             : 5 seconds
  Threshold           : 5 seconds
  Running Frequency   : 8 seconds
Return code           : OK

Switch# ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.846 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.643 ms
```

```
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.978 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.640 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=0.704 ms
```

Shutdown the interface eth-0-1 on Switch2.

```
Switch(config)# interface eth-0-1
Switch(config-if)# shutdown
```

Display the result on Switch1 again.

```
Switch# show ip sla monitor
Entry 1
  Type                  : Echo
  Admin state           : Enable
  Destination address   : 192.168.0.1
  Frequency             : 10 seconds
  Timeout               : 5 seconds
  Threshold             : 5 seconds
  Running Frequency     : 9 seconds
  Running Timeout       : 4 seconds
  Running Threshold     : 4 seconds
Return code             : Timeout
```

Configuring IP SLA for outgongin interface of static route

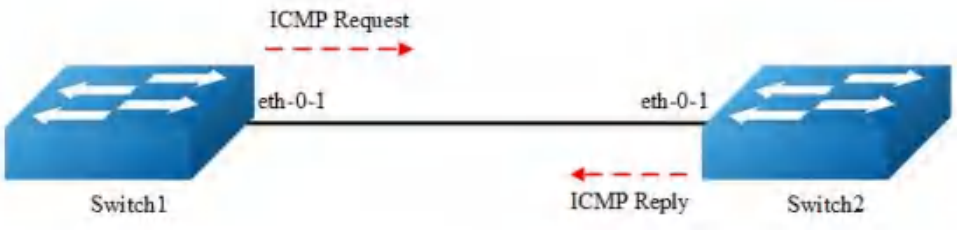


Figure 17-46 IP SLA

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.2/24n
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.1/24
Switch(config-if)# exit
```

```
Switch(config)# interface loopback 1
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# exit
```

step 3 Create ip sla and set the attributes

Configuring Switch1:

```
Switch(config)# ip sla monitor 2
Switch(config-ipsla)# type icmp-echo 1.1.1.1
Switch(config-ipsla)# frequency 10
Switch(config-ipsla)# timeout 5
Switch(config-ipsla)# exit
```

step 4 Enable ip sla

Configuring Switch1:

```
Switch(config)# ip sla monitor schedule 2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch# show ip sla monitor 2
Entry 2
  Type                : Echo
  Admin state         : Enable
  Destination address : 1.1.1.1
  Frequency           : 10 seconds
  Timeout             : 5 seconds
  Threshold           : 5 seconds
  Running Frequency   : 1 seconds
  Return code         : Unreachable
Switch# ping 1.1.1.1
connect: Network is unreachable
```

Create a static route on Switch1

```
Switch#configure terminal
Switch(config)# ip route 1.1.1.1/32 192.168.0.1
Switch(config)# end
```

Display the result on Switch1 again.

```
Switch# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=1.63 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=0.661 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=64 time=0.762 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=64 time=0.942 ms


Entry 2
  Type                : Echo
  Admin state         : Enable
```

```
Destination address          : 1.1.1.1
Frequency                    : 10 seconds
Timeout                      : 5 seconds
Threshold                    : 5 seconds
Running Frequency            : 8 seconds
Return code                  : OK
```

Configuring track interface linkstate

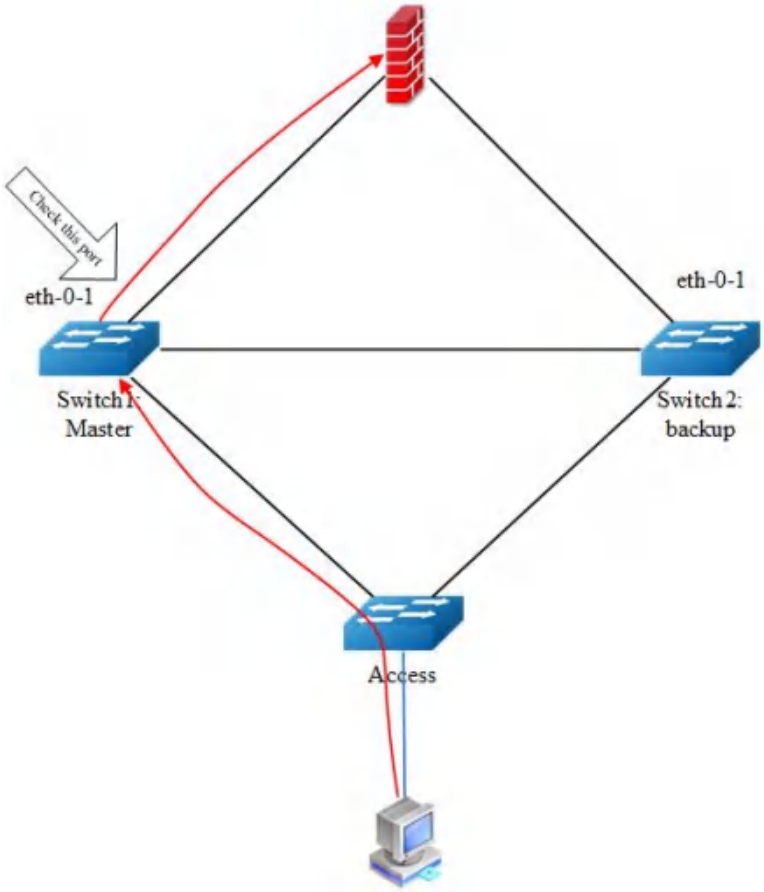


Figure 17-47 Track interface

Before the introduction of track feature, the VRRP had a simple tracking mechanism that allowed you to track the interface link state only. If the link state of the interface went down, the VRRP priority of the router was reduced, allowing another VRRP router with a higher priority to become active. The Track feature separates the tracking mechanism from VRRP and creates a separate standalone tracking process that can be used by other processes in future. This feature allows tracking of other objects in addition to the interface link state. VRRP can now register its interest in tracking objects and then be notified when the tracked object changes state. TRACK is a separate standalone tracking process that can be used by other processes as well as VRRP. This feature allows tracking of other objects in addition to the interface link state.

Configuring Switch1:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Create track and set the attributes

```
Switch(config)# track 1 interface eth-0-1 linkstate
Switch(config-track)# delay up 30
```

```
Switch(config-track)# delay down 30
Switch(config-track)# exit
```

[]Parameters for track:

- delay up: After the interface states is up, the track will wait for a cycle before restore the states. Valid range is 1-180 second. The default configuration is restore without delay.
- delay down: After the interface states is down, the track will wait for a cycle before change the states. Valid range is 1-180 second. The default configuration is change without delay.

If the track is using bfd or ip sla, the "delay up" and "delay down" is similar as using interface states.

step 3 Exit the configure mode

```
Switch(config)# end
```

step 4 Validation

```
Switch#show track
Track 2
  Type            : Interface Link state
  Interface       : eth-0-1
  State           : down
  Delay up        : 30 seconds
Delay down        : 30 seconds
```

Configuring track ip sla reachability

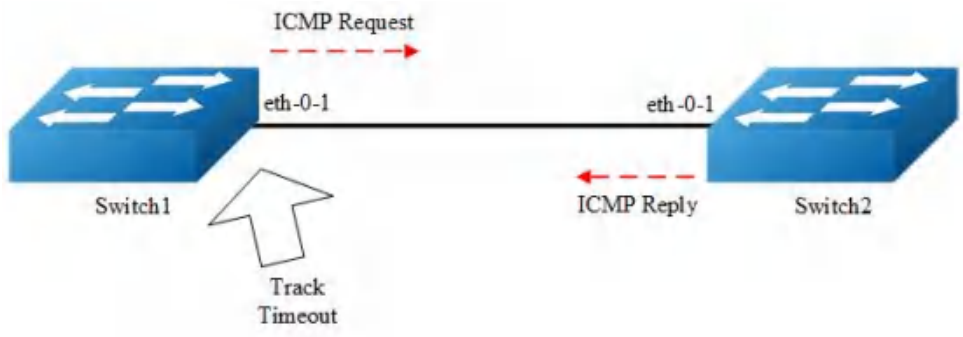


Figure 17-48 Track ip sla

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.2/24
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.1/24
```

step 3 Create ip sla and enable it

Configuring Switch1:

```
Switch(config)# ip sla monitor 1
Switch(config-ipsla)# type icmp-echo  192.168.0.1
Switch(config-ipsla)# frequency 10
Switch(config-ipsla)# timeout 5
Switch(config-ipsla)# threshold 1
Switch(config-ipsla)# exit
Switch(config)# ip sla monitor schedule 1
```

step 4 Create track and set the attributes

Configuring Switch1:

```
Switch(config)# track 1 rtr 1 reachability
Switch(config-track)# delay up 30
Switch(config-track)# delay down 30
Switch(config-track)#exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch#show track
Track 1
  Type              : Response Time Reporter(RTR) Reachability
  RTR entry number    : 1
  State             : up
  Delay up          : 30 seconds
Delay down          : 30 seconds
```

Configuring track ip sla state

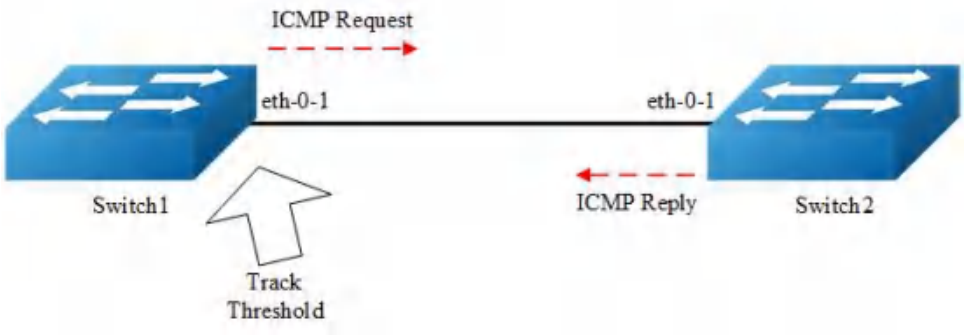


Figure 17-49 Track ip sla

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.2/24
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.0.1/24
```

step 3 Create ip sla and enable it

Configuring Switch1:

```
Switch(config)# ip sla monitor 1
Switch(config-ipsla)# type icmp-echo  192.168.0.1
Switch(config-ipsla)# frequency 10
Switch(config-ipsla)# timeout 5
Switch(config-ipsla)# threshold 1
Switch(config-ipsla)# exit
Switch(config)# ip sla monitor schedule 1
```

step 4 Create track and set the attributes

Configuring Switch1:

```
Switch(config)# track 1 rtr 1 state
Switch(config-track)# delay up 30
Switch(config-track)# delay down 30
Switch(config-track)#exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show track
Track 1
  Type              : Response Time Reporter(RTR) State
  RTR entry number    : 1
  State             : up
  Delay up          : 30 seconds
Delay down          : 30 seconds
```

Configuring track bfd

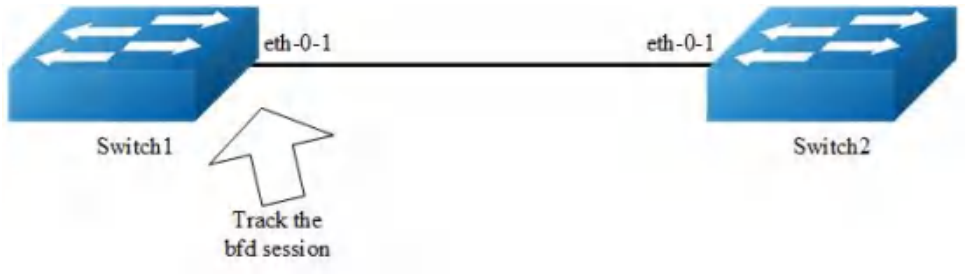


Figure 17-50 Track bfd

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# quit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# quit
```

step 4 Create track and set the attributes

Configuring Switch1:

```
Switch(config)# track 1 bfd source interface eth-0-1 destination 9.9.9.2
Switch(config-track)# delay up 30
Switch(config-track)# delay down 30
Switch(config-track)# exit
```

Configuring Switch2:

```
Switch(config)# track 1 bfd source interface eth-0-1 destination 9.9.9.1
Switch(config-track)# delay up 30
Switch(config-track)# delay down 30
Switch(config-track)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

Display the result on Switch1.

```
Switch #show track
Track 1
```

```
  Type              : BFD state
  Source interface  : eth-0-1
  Destination IP    : 9.9.9.2
  BFD Local discr   : 1
  State             : up
```

Display the result on Switch2.

```
Switch # show track
Track 1
  Type              : BFD state
  Source interface  : eth-0-1
  Destination IP    : 9.9.9.1
  BFD Local discr   : 1
  State             : up
```

Configuring track for vrrp

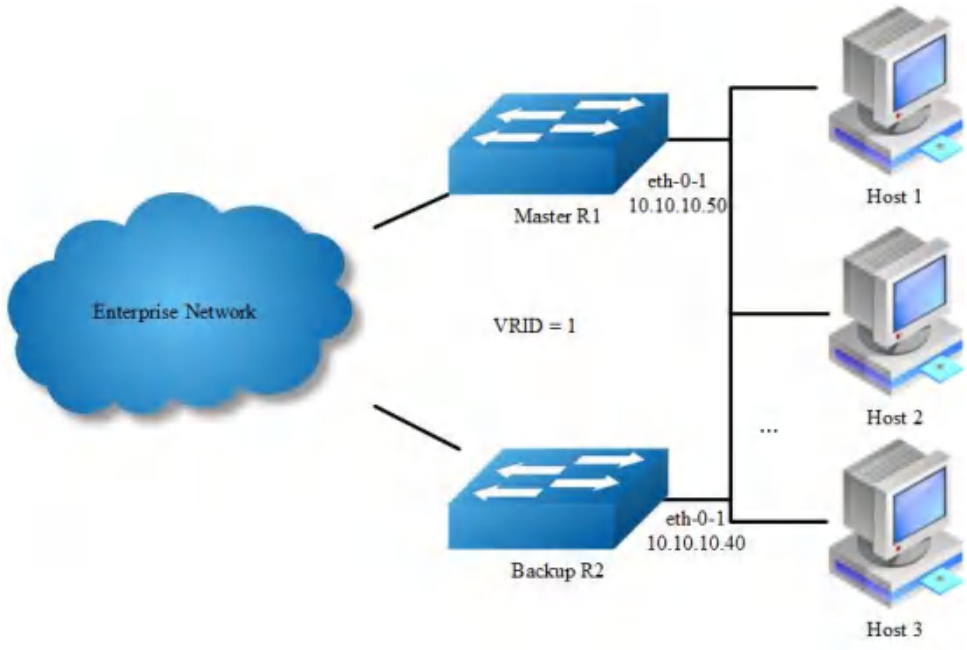


Figure 17-51 VRRP Track

step 1 Check current configuration

Reference to chapter "Configuring VRRP" - "Configuring VRRP (One Virtual Router)"

Display the configuration on R1.

```
interface eth-0-1
 no switchport
 ip address 10.10.10.50/24
!
router vrrp 1
 interface eth-0-1
 virtual-ip 10.10.10.60
 advertisement-interval 5
 enable
```

Display the configuration on R2.

```
interface eth-0-1
 no switchport
 ip address 10.10.10.40/24
!
router vrrp 1
 interface eth-0-1
 priority 200
 virtual-ip 10.10.10.60
 advertisement-interval 5
 enable
```

step 2 Create track and set the attributes

Create track on Switch1

Bind to interface linkstate

```
Switch(config)# track 1 interface eth-0-1 linkstate
Switch(config-track)# exit
```

Bind to CFM (Please reference to "CFM" part in this guide)

```
Switch(config)# track 1 cfm domain cust service cst
Switch(config-track)# exit
```

Note: Use the following example to monitoring the RDI state in CCM packets

```
Switch(config)# track 1 cfm domain cust service cst rdi-trigger
```

step 3 Apply track for vrrp

Apply track on Switch1

```
Switch(config)# router vrrp 1
Switch(config-router)# disable
Switch(config-router)# track 1 decrement 30
Switch(config-router)# enable
```

step 4 Validation

Display the result on Switch1.

```
Switch# show vrrp
vrrp session count: 1
VRID <1>
 State           : Backup
 Virtual IP      : 10.10.10.60(Not IP owner)
 Interface       : eth-0-9
 VMAC            : 0000.5e00.0101
 VRF             : Default
 Advt timer      : 5 second(s)
 Preempt mode    : TRUE
 Conf pri        : Unset       Run pri  : 100
 Increased pri   : 0
 Track Object    : 1
 Decre pri       : 30
 Master router ip  : 10.10.10.40
 Master priority   : 200
 Master advt timer  : 5 second(s)
 Master down timer  : 16 second(s)
 Preempt delay     : 0 second(s)
```

```
Learn master mode   : FALSE
BFD session state   : UNSET
```

Configuring track for static route



Figure 17-52 Static Route Track

The following configuration should be operated on all switches if the switch ID is not specified.:

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)#interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.1.10/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)#interface eth-0-1
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 192.168.1.11/24
Switch(config-if)# exit
```

step 3 Create ip sla and enable it

Configuring Switch1:

```
Switch(config)# ip sla monitor 1
Switch(config-ipsla)# type icmp-echo 192.168.1.11
Switch(config-ipsla)# exit
Switch(config)# ip sla monitor schedule 1
```

step 4 Create track and set the attributes

Configuring Switch1:

```
Switch(config)# track 1 rtr 1 reachability
Switch(config-track)# exit
```

step 5 Apply track for static route

```
Switch(config)#ip route 10.10.10.0/24 192.168.1.11 track 1
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1.

```
Switch# show ip sla monitor 1
Entry 1
  Type             : Echo
  Admin state      : Enable
  Destination address : 192.168.1.11
  Frequency        : 60 seconds
  Timeout          : 5 seconds
  Threshold        : 5 seconds
  Running Frequency  : 49 seconds
Return code      : OK

Switch# show track 1
Track 1
  Type             : Response Time Reporter(RTR) Reachability
  RTR entry number  : 1
  State            : up
Switch# show ip route static
S      10.10.10.0/24 [1/0] via 192.168.1.11, eth-0-1
```

Shutdown the interface eth-0-1 on Switch2.

```
Switch(config)# interface eth-0-1
Switch(config-if)# shutdown
```

Display the result on Switch1 again.

```
Switch# show ip sla monitor 1
Entry 1
  Type             : Echo
  Admin state      : Enable
  Destination address : 192.168.1.11
  Frequency        : 60 seconds
  Timeout          : 5 seconds
  Threshold        : 5 seconds
  Running Frequency  : 8 seconds
Return code      : Timeout
Switch# show track 1
Track 1
  Type             : Response Time Reporter(RTR) Reachability
  RTR entry number  : 1
  State            : down
Switch# show ip route static
Switch#
```

17.11.3 Application cases

N/A

17.12 Configuring IP BFD

17.12.1 Overview

Function Introduction

An increasingly important feature of networking equipment is the rapid detection of communication failures between adjacent systems, in order to more quickly establish alternative paths. Detection can come fairly quickly in certain circumstances when data link hardware comes into play (such as Synchronous Optical Network (SONET) alarms). However, there are media that do not provide this kind of

signaling (such as Ethernet), and some media may not detect certain kinds of failures in the path, for example, failing interfaces or forwarding engine components.

Networks use relatively slow "Hello" mechanisms, usually in routing protocols, to detect failures when there is no hardware signaling to help out. The time to detect failures ("Detection Times") available in the existing protocols is no better than a second, which is far too long for some applications and represents a great deal of lost data at gigabit rates. Furthermore, routing protocol Hellos are of no help when those routing protocols are not in use, and the semantics of detection are subtly different – they detect a failure in the path between the two routing protocol engines.

The goal of Bidirectional Forwarding Detection (BFD) is to provide low-overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data link(s), and, to the extent possible, the forwarding engines themselves.

An additional goal is to provide a single mechanism that can be used for aliveness detection over any media, at any protocol layer, with a wide range of Detection Times and overhead, to avoid a proliferation of different methods.


If ethernet CFM mep is configured on an physical port and CFM LM is enabled, at the same time, IP BFD is configured on an vlan interface and the former physical port is a member of the vlan, IP BFD can't work normally. If CFM LM is disabled, IP BFD can work normally.

Principle Description

Reference RFC 5880 Bidirectional Forwarding Detection (BFD)

17.12.2 Configuration

Configure BFD single hop

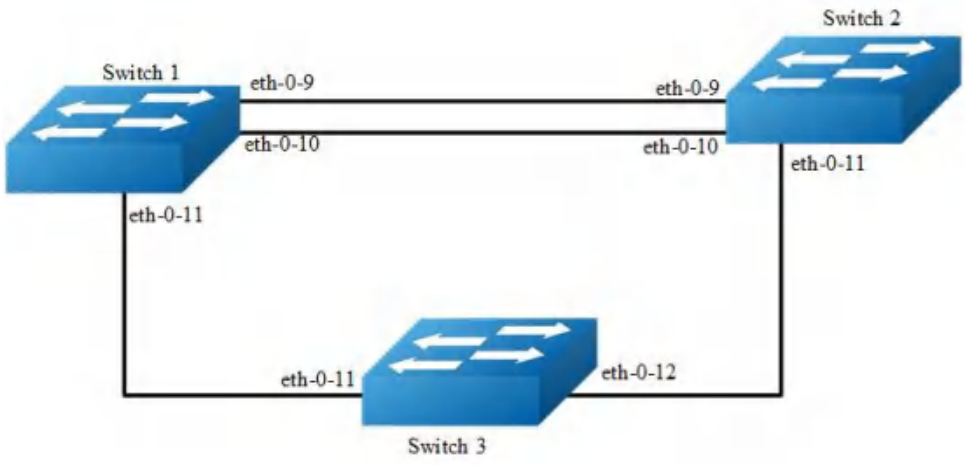


Figure 17-53 BFD single hop

This topology and configuration is for three BFD session ,one session based on static configuration with static route,next session based on OSPF ,and last session relate vrrp.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
```

```
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# bfd interval mintx 3 minrx 3 multiplier 3
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.1/24
Switch(config-if)# bfd interval mintx 3 minrx 3 multiplier 3

Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.1/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# bfd interval mintx 3 minrx 3 multiplier 3
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 10.10.10.2/24
Switch(config-if)# bfd interval mintx 3 minrx 3 multiplier 3
Switch(config-if)# ip ospf bfd
Switch(config-if)# exit

Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.2/24
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-11
Switch(config-if)# no shutdown
Switch(config-if)#exit

Switch(config)# interface eth-0-12
Switch(config-if)# no shutdown
Switch(config-if)#exit
```

step 3 Configuring ospf

Configuring Switch1:

```
Switch(config)# router ospf
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# exit
```

Configuring Switch2:

```
Switch(config)# router ospf
Switch(config-router)# network 10.10.10.0/24 area 0
Switch(config-router)# exit
```

step 4 Configuring vrrp

Configuring Switch1:

```
Switch(config)#router vrrp 1
Switch(config-router)#virtual-ip 11.11.11.100
Switch(config-router)# priority 120
Switch(config-router)#interface eth-0-11
Switch(config-router)# enable
Switch(config-router)# exit
```

Configuring Switch2:

```
switch(config)#bfd test peer-ip 11.11.11.1 interface eth-0-11 auto
switch(config)#track 1 bfd session test
Switch(config)#router vrrp 1
Switch(config-router)#virtual-ip 11.11.11.100
Switch(config-router)# priority 100
Switch(config-router)#interface eth-0-11
Switch(config-router)# track 1 increment 50
Switch(config-router)# enable
Switch(config-router)# exit
```

step 5 Configuring static route

Configuring Switch1:

```
Switch(config)# bfd test peer-ip 9.9.9.2 interface eth-0-9 auto
Switch(config)# ip route 1.1.1.0/24 9.9.9.2 bind bfd test
```

Configuring Switch2:

```
Switch(config)# bfd test peer-ip 9.9.9.1 interface eth-0-9 auto
Switch(config)# ip route 2.2.2.0/24 9.9.9.1 bind bfd test
```

step 6 Exit configure mode

```
Switch(config)# end
```

step 7 Validation

Display Switch1 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD  RD  TYPE ST  UP-Time   Remote-Addr    vrf
1   1   S-DD U   00:01:05  9.9.9.2       default
2   2   S-DD U   00:00:25  10.10.10.2    default
Number of Sessions:   2
```

Display Switch2 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD  RD  TYPE ST  UP-Time   Remote-Addr     vrf
1   1   S-DD   U   00:01:27  9.9.9.1      default
2   2   S-DD   U   00:00:46  10.10.10.1   default
3   3   S-DD   U   00:00:25  11.11.11.1   default
Number of Sessions:   3
```

Configuring BFD multi-hop

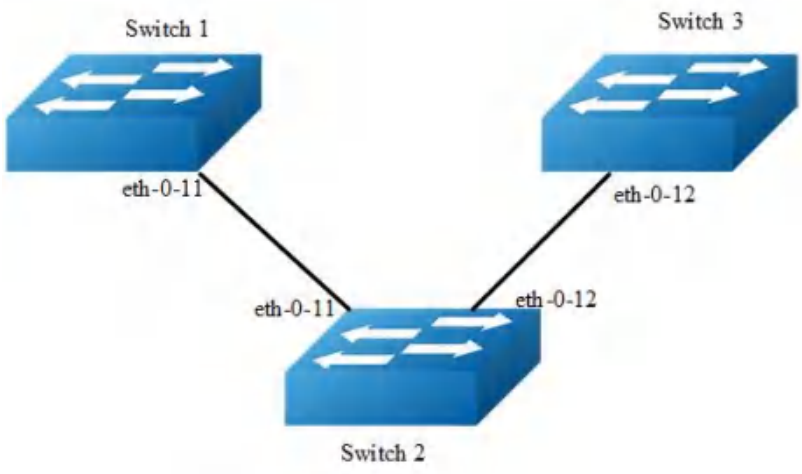


Figure 17-54 BFD multi hop

This topology and configuration is for one BFD session which is based on static multiple bfd for static route,

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.1/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.2/24
Switch(config-if)# exit
```

```
Switch(config)#interface eth-0-12
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)#ip address 12.12.12.1/24
Switch(config-if)#exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-12
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)#ip address 12.12.12.2/24
Switch(config-if)#exit
```

step 3 Configuring static route

Configuring Switch1:

```
Switch1(config)#ip route 12.12.12.2/24 11.11.11.2
Switch1(config)# bfd test peer-ip 12.12.12.2 source-ip 11.11.11.1 local 10 remote 20
Switch1(config)# ip route 192.168.1.1/24 12.12.12.2 bind bfd test
```

Configuring Switch2:

```
Switch3(config)#ip route 11.11.11.1/24 12.12.12.1
Switch3(config)#bfd test peer-ip 11.11.11.1 source-ip 12.12.12.2 local 20 remote 10
Switch3(config)#ip route 2.2.2.2/24 11.11.11.1 bind bfd test
```

step 6 Exit configure mode

```
Switch(config)# end
```

step 7 Validation

Display Switch1 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD   RD   TYPE ST   UP-Time   Remote-Addr     vrf
10   20   S-SD U   00:01:27   12.12.12.2      default
```

Display Switch2 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD   RD   TYPE ST   UP-Time   Remote-Addr     vrf
20   10   S-SD U   00:01:27   11.11.11.1      default
```

Configuring BFD waiting to report interval

Topology

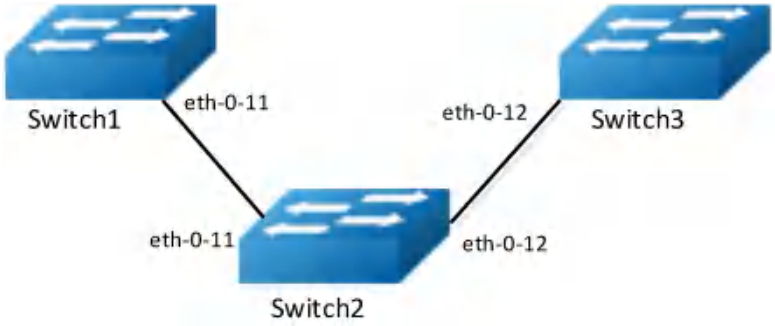


Figure 17-55 BFD WTR

This topology and configuration is for one BFD session.

Configuration Steps

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.1/24
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-11
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 11.11.11.2/24
Switch(config-if)# exit
Switch(config)#interface eth-0-12
Switch(config-if)#no switchport
Switch(config-if)#no shutdown
Switch(config-if)#ip address 12.12.12.1/24
Switch(config-if)#exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-12
Switch(config-if)# no switchport
Switch(config-if)# no shutdown
Switch(config-if)# ip address 12.12.12.2/24
Switch(config-if)# exit
```

step 3 Configuring static route

Configuring Switch1:

```
Switch1(config)# ip route 12.12.12.2/32 11.11.11.2
Switch1(config)# bfd test peer-ip 12.12.12.2 source-ip 11.11.11.1 auto
Switch1(config-ip-bfd)# wtr 120
```

Configuring Switch3:

```
Switch3(config)# ip route 11.11.11.1/32 12.12.12.1
Switch3(config)# bfd test peer-ip 11.11.11.1 source-ip 12.12.12.2 auto
```

step 4 Exit configure mode

```
Switch(config)# end
```

Validation

Display Switch1 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD  RD  Remote-Addr  TYPE ST  UP-Time  vrf
8192 8193  12.12.12.2
               M-SD U   00:01:27 default
Switch# show bfd session detail
=====================================================

Session Interface : eth-0-11          Session Name : test
Lower Layer : IPv4                Version : 1
Session Type : Multi Hop IP BFD      Session State : UP
Discriminator Type : Dynamic         vrf : default
Local Discriminator : 8192          Local Address : 11.11.11.1
Remote Discriminator: 0             Remote Address : 12.12.12.2
Local Port : 49152                Remote Port : 4784
DSCP : 56                      TTL : 255
Sbfd Type : None                  Mbfd Type : None

Bind FEC : -

Diagnostics : None

WTR Interval in seconds : 120

Timers in Milliseconds :
Min Tx: 20          Min Rx: 20          Multiplier: 3
Neg Tx: 20          Neg Rx: 20          Neg detect mult: 3

Sess up time : 00:01:27
Sess down time : 00:00:00
Bind Application : -
-----------------------------------------------------------
Number of Sessions:   1
```

Display Switch2 results:

```
Switch# show bfd session
abbreviation:
LD: local Discriminator.   RD: Discriminator
S: single hop session.   M: multi hop session.
SD: Static Discriminator.  DD: Dynamic Discriminator
```

```
A: Admin down.   D:down.   I:init.   U:up.
=====================================================
LD  RD  Remote-Addr  TYPE ST  UP-Time  vrf
8193 8192  11.11.11.1
              M-SD U   00:01:27 default
```

17.13 Configuring VARP

17.13.1 Overview

Function Introduction

Virtual ARP (VARP) allows multiple switches to simultaneously route packets with the same destination MAC address. Each switch is configured with the same virtual MAC address for the the L3 interfaces configured with a virtual IP address. In MLAG configurations, VARP is preferred over VRRP because VARP working on active-active mode without traffic traverse peer link.

For ARP and GARP requests to virtual IP address, VARP will use the virtual MAC address to reply. The virtual MAC address is only used in the destination field of inbound packets and never used in the source field of outbound packets.Topology

Principle Description

N/A

17.13.2 Configuration

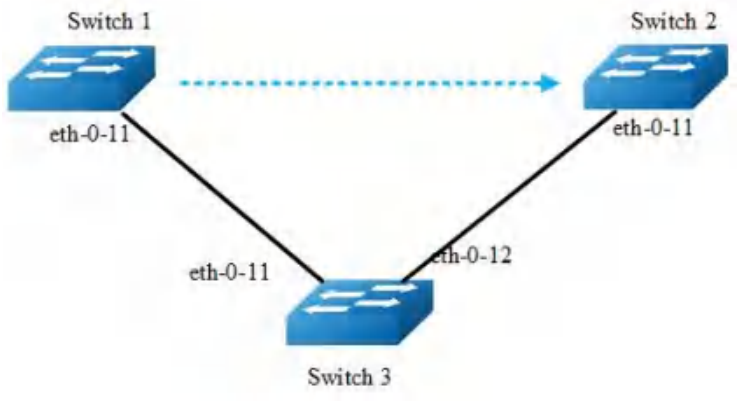


Figure 17-56 VARP with MALG

The following configuration should be operated on all devices if the device ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Set the virtual-router mac address

```
Switch(config)# ip virtual-router mac a.a.a
```

step 3 Enter the vlan configure mode and create the vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 2
Switch(config-vlan)# exit
```

step 4 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-11
Switch(config-if)# switchport access vlan 2
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

step 5 Create the vlan interface and set ip and virtual router ip

Configuring Switch1:

```
Switch1(config)# interface vlan 2
Switch1(config-if)# ip address 10.10.10.1/24
Switch1(config-if)# ip virtual-router address 10.10.10.254
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config-if)# interface vlan 2
Switch2(config-if)# ip address 10.10.10.2/24
Switch2(config-if)# ip virtual-router address 10.10.10.254
Switch2(config-if)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1.

```
Switch1# show ip arp
Protocol   Address        Age (min)  Hardware Addr   Interface
Internet   10.10.10.1          -  cef0.12da.8100  vlan2
Internet   10.10.10.254        -  000a.000a.000a  vlan2
```

Display the result on Switch2.

```
Switch2#  show ip arp
Protocol   Address        Age (min)  Hardware Addr   Interface
Internet   10.10.10.2          -  66d1.4c26.e100  vlan2
Internet   10.10.10.254        -  000a.000a.000a  vlan2
```

17.13.3 Application cases

N/A

17.14 Configuring UDP Helper

17.14.1 Overview

Brief Introduction

The main function of UDP helper is to relay and forward the specified UDP message in IP broadcast packet, convert the specified UDP message in IP broadcast packet into unicast packet and send it to the specified server, it plays a role of relay.

After enabling the UDP helper function, the device will make a judgement on the destination port number of the received broadcast UDP packet. If the packet whose destination port number matches the port number configured by the UDP helper, it will copy it and modify the the destination IP address of packet header and sent to the designated server.

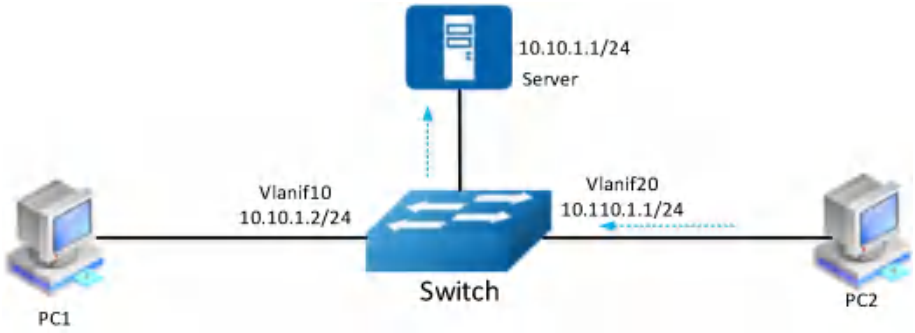


Figure 17-57 UDP-Helper configuration

The default 6 UDP destination port：

Protocol	UDP destination port
DNS (Domain Name System)	53
NetBIOS-DS (NetBIOS Datagram Service)	138
NetBIOS-NS (NetBIOS Name Service)	137
TACACS (Terminal Access Controller Access Control System)	49
TFTP (Trivial File Transfer Protocol)	69
Time Service	37

17.14.2 Configuration Examples

Step 2 Enter the configure mode

```
Switch# configure terminal
```

Step 3 Enable UDP Helper

```
Switch(config)# ip udp-helper enable
```

Step 4 Configure the IP address and UDP Helper Server IP address on interface

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10,20
Switch(config-vlan)# exit
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20
Switch(config-if)# exit
Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# exit
Switch(config)# interface vlan 20
Switch(config-if)# ip address 10.110.1.1/24
Switch(config-if)# ip udp-helper server 10.10.1.1
Switch(config-if)# exit
Switch(config)# interface vlan 10
Switch(config-if)# ip address 10.10.1.2/24
```

Step 5 configure the ARP

```
Switch(config)# arp 10.10.1.1 0.0.1
```

Step 6 Exit the configure mode

```
Switch(config)# end
```

Step 7 Validation

To display the UDP Helper configuration, use following privileged EXEC commands.

```
Switch# show  ip udp-helper server
Interface     Server IP       Packet Received Packet Dropped
--------------+---------------+---------------+---------------
vlan20        10.10.1.1       0               0
```

18 Network Virtualization Configuration Guide

18.1 Configuring VXLAN

18.1.1 Overview

Function Introduction

Virtual Extensible LAN (VXLAN) is a networking technology that encapsulates MAC-based Layer 2 Ethernet frames within Layer 3 UDP packets to aggregate and tunnel multiple layer 2 networks across a Layer 3 infrastructure. VXLAN scales up to 16 million logical networks and supports layer 2 adjacency across IP networks. Multicast transmission architecture is used for broadcast/multicast/unknown packets.

Background

Nowadays, Server virtualization gets more and more recognitions and deployments since it reduces the cost of IT, improves the flexibilities of business deployment and reduces the cost of maintenance and other advantages. One server can virtual into multiple virtual machines; one virtual machine is equal to a host, the number of hosts has a big change on its amount.

When virtual machine runs, it needs to move to new server because the resources of server and other problems (such as CPU is too high, the storage is not enough and so on). To make sure that business is not interrupted during the migration, it needs to make sure that the IP address, MAC address and other parameters of virtual machine are not changing. This is asking the business network as a Layer 2 network, and requires that the network itself has multiple paths of redundancy and reliabilities.

The above requirements make traditional Layer 2 domain becomes larger and larger. The traditional network becomes powerless to cope with larger Layer 2 requirements, it mainly reflected in the following aspects:

- Under the environment of larger Layer 2 networks, packets forwarding by looking up MAC address table; the volume of switch's MAC address table limits the number of virtual machines
- Network isolates abilities limitation, the most popular isolation techniques are VLAN or VPN (Virtual Private Network). It has the following limitation that large size of virtual networks deploy: since IEEE 802.1Q defines that VLAN Tag domain only has 12 bit, so it only show 4096 VLAN which cannot satisfy users group's requirement in large Layer 2 network.
- VLAN/VPN in traditional Layer 2 network cannot satisfy the requirement of network dynamic adjustment. The network structure limits the range that virtual machines migrate.

Direct at the above problems in large Layer 2 network, the raise of VXLAN (Virtual eXtensible Local Area Network) solves the problems well.

- Aim to virtual machines suffer from network size limitation, VXLAN encapsulates the data packages that virtual machines send into UDP, and uses IP and MAC address from physical network as external header to encapsulate, then express the parameters after encapsulating to network. Hence, it rapidly reduces the requirements of MAC addresses' size that large Layer 2 network requires.
- Direct at the network isolates abilities limitation, VXLAN imports users id which is similar as VLAN ID, that is consisted of 24 bit; it supports up to 16 million users and is satisfied by large amount of user id
- Aim to the range of virtual machines migrate is limited by network structure, Using VXLAN to construct large Layer 2 network makes sure the IP and MAC address are keeping the same during virtual machines migrate.

Principle Description

1. Basic principle

Abbreviation and Terminology

Abbreviation/Terminology	Description
VTEP(VXLAN Tunnel Endpoint)	VETP is the terminated point of VXLAN tunnel, is the device that is used for encapsulating and decapsulating VXLAN packets

Basic principle

VXLAN is a kind of technique of network virtualization. Figure 2-1 shows the packet format of VXLAN, VXLAN packet adds a VXLAN header and UDP/IP header to the packet, original packets which are sent by VM or physical server, and uses added MAC/IP to forward on

network devices that it passes through, then it restores the decapsulated packets on tunnel endpoint and sends the original packet to VM or physical server.

Packet encapsulation

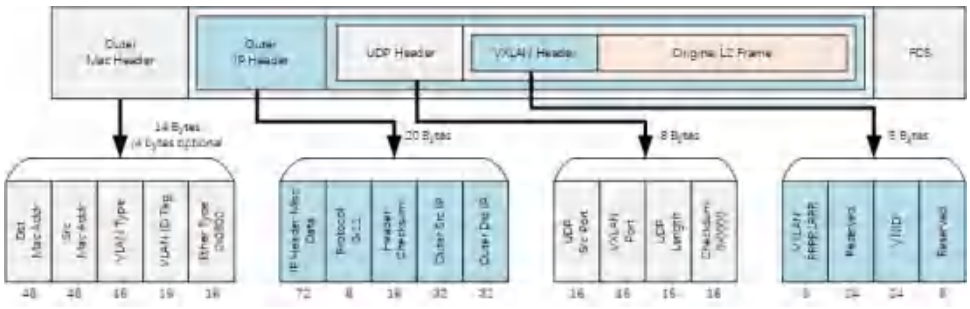


Figure 18-1 The packet format of VXLAN

VXLAN header encapsulates:

- Flag: 8 bit, current protocol value is 00001000, the fifth bit indicates whether the VXLAN packet is valid or not
- VNID: VXLAN Network Identifier, VXLAN network id, 24 bit, is used for distinguish VXLAN sections.
- Reserved: 24 bit and 8 bit, current protocol regulates that should be 0 for all.

UCP header encapsulates:

- destination port number is 4789, source port number is depends the calculation of Hash value according to header contexts of original packets

External IP header:

- The source ip is the IP of VTEP where the transmit server or virtual machine belongs to packet; the destination IP is the IP of VTEP server or virtual machine belongs to.

External Ethernet header:

- Src MAC: the MAC of VTEP's physical network interface that sending packets
- Dst MAC: MAC of the next hop to destination VTEP IP.
- VLAN: if the under layer physical network uses VLAN interface, then it can take the corresponding VLAN TAG optionally.

Forwarding Theory - Broadcast in Layer 2

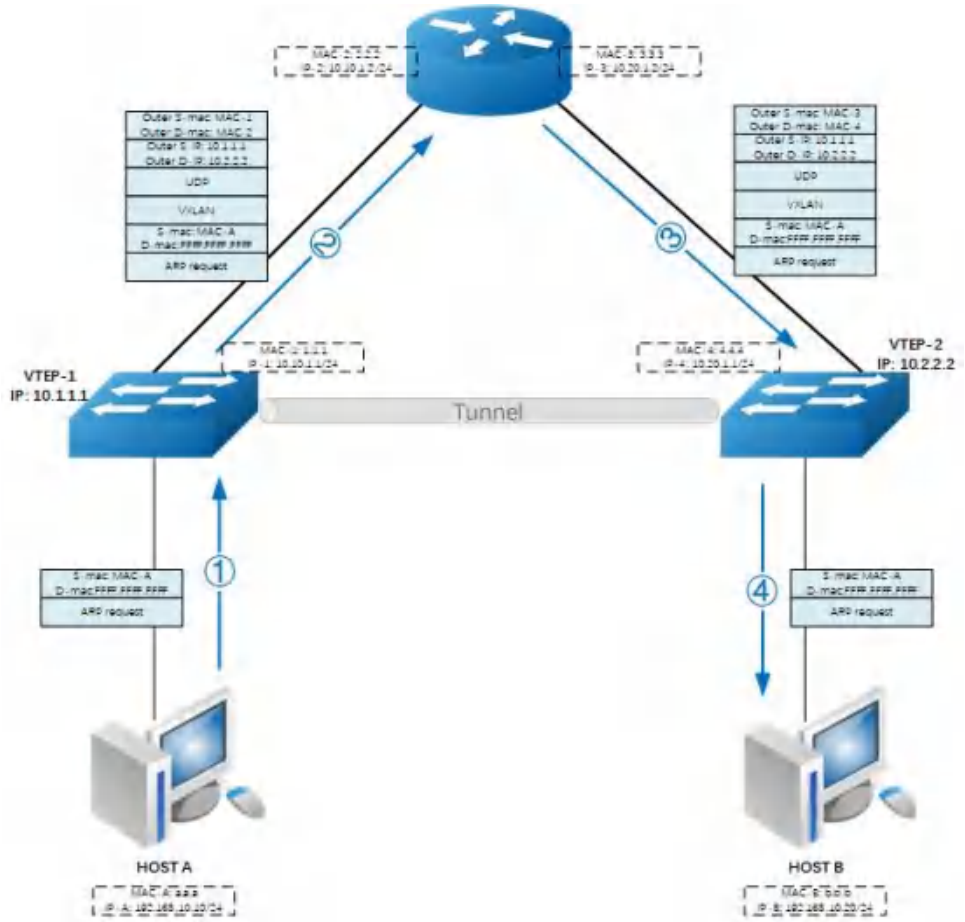


Figure 18-2 The process of broadcast packets forwarding

As shown in the figure above, HOST A and HOST B are in the same network segment but distributes in different VTEP, the detail networking process for HOST A and HOST B is the following:

- HOST A and HOST B are in the same network segment, HOST A sends ARP to request the MAC address of HOST B
- When ARP requested packet reaches to VTEP-1, switch finds that it is a broadcast packet and need to flood in VLAN, and one copy of this packet is sent to VXLAN tunnel (broadcast copies on the head VTEP, if there are several tunnels, it will send one packet to each tunnel). VLAN maps to VNI with VXLAN capsulate, it will add the external UDP IP encapsulation, according the external IP searching routing table to make sure the next hop of this packet. Switch will learn address MAC-A to interface eth-0-1 in corresponding VLAN at the same time
- Packet is forwarding depends on the external IP; it will reach to VTEP-2 at last.
- After VTEP-2 receives this packet, it will find that the outer D-mac is the local address, meanwhile, it will check whether local has configured corresponding tunnel by packet's Outer S-IP, Outer D-IP and VNI. If it has the tunnel, then it will take decapsulated operation, cut the outer header and map VNI into corresponding VLAN; it will broadcast in VLAN when it finds it is broadcast packet. At the same time, it will learn Mac-A to relevant tunnel interface in VLAN.

Forwarding Theory - Unicast in Layer 2

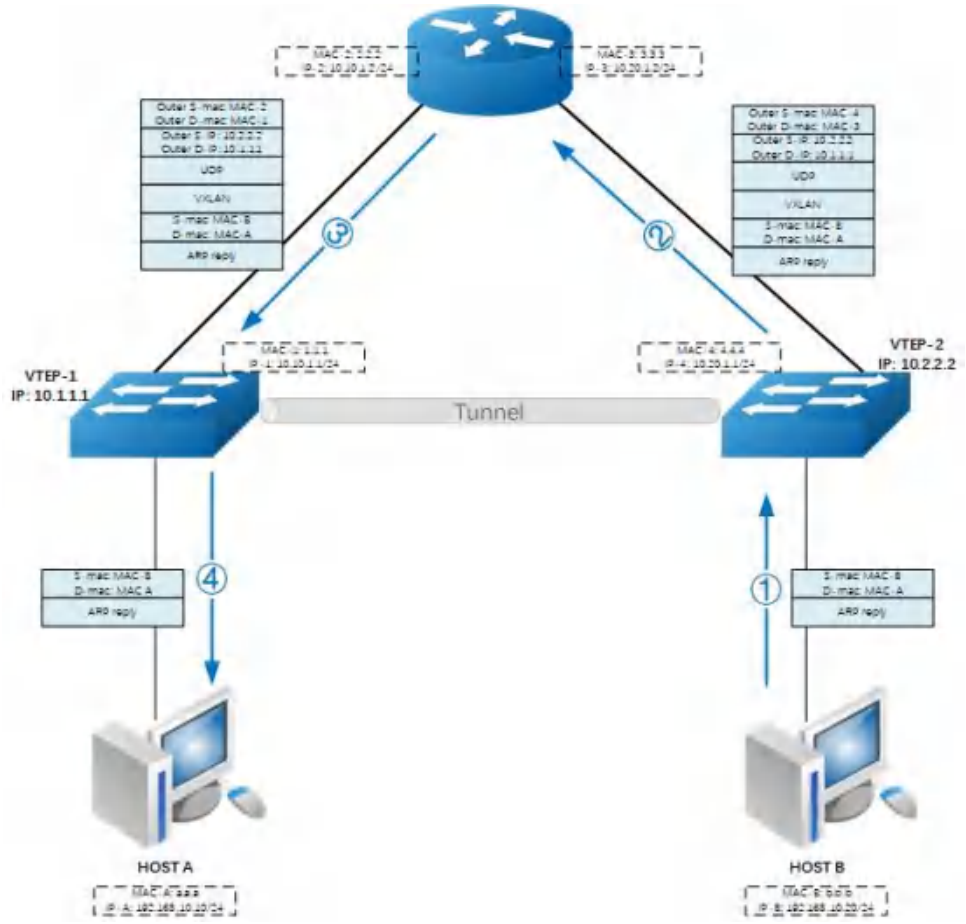


Figure 18-3 The process of unicast packets forwarding

As shown in the figure above, HOST B replies the ARP request sent from HOST A; this packet is unicast packet, the following example shows the process of unicast packet forwarding:

- HOST B sends ARP reply; it will reach to switch VTEP-2 by eth-0-1.
- On VTEP-2, it will find that the outbound interface of MAC-B is VXLAN tunnel in this VLAN, and remote terminal VTEP is 10.1.1.1 by searching mac address table using Dst-mac; then map the VLAN where the original packet is in to VNI and add the header of VXLAN, and add outer MAC/IP header, Outer S-IP as local VTEP IP, Outer D-IP ad remote terminal VTEP IP, Outer S-mac as local outbound interface's mac, Outer D-mac as the next hop's interface mac for Outer D-IP.
- Packet is forwarding according to outer IP in the network, it will reach to VTEP-1 finally.
- After VTEP-1 receives packet and finds that the Outer D-mac is the local address, it will examine whether it configures the relevant tunnel by packet's Outer S-IP, Outer D-IP and VNI. If it has configured, then it will take decapsulation action to cut the outer header and map VNI to corresponding VLAN, and search the mac address table through original packet's Dst-mac (MAC-A); then it will find that the outbound interface of Mac-A is eth-0-1, then it will send the packet after decapsulating through eth-0-1. Meanwhile, it learns Mac-B to relevant tunnel interface in VLAN.



By default, VTEP has tunnels that connect with each other; if it finds that the exit of Dst-mac is tunnel after decapsulating, then the packet will be discard.

Forwarding Theory - Concentrated VXLAN network gateway

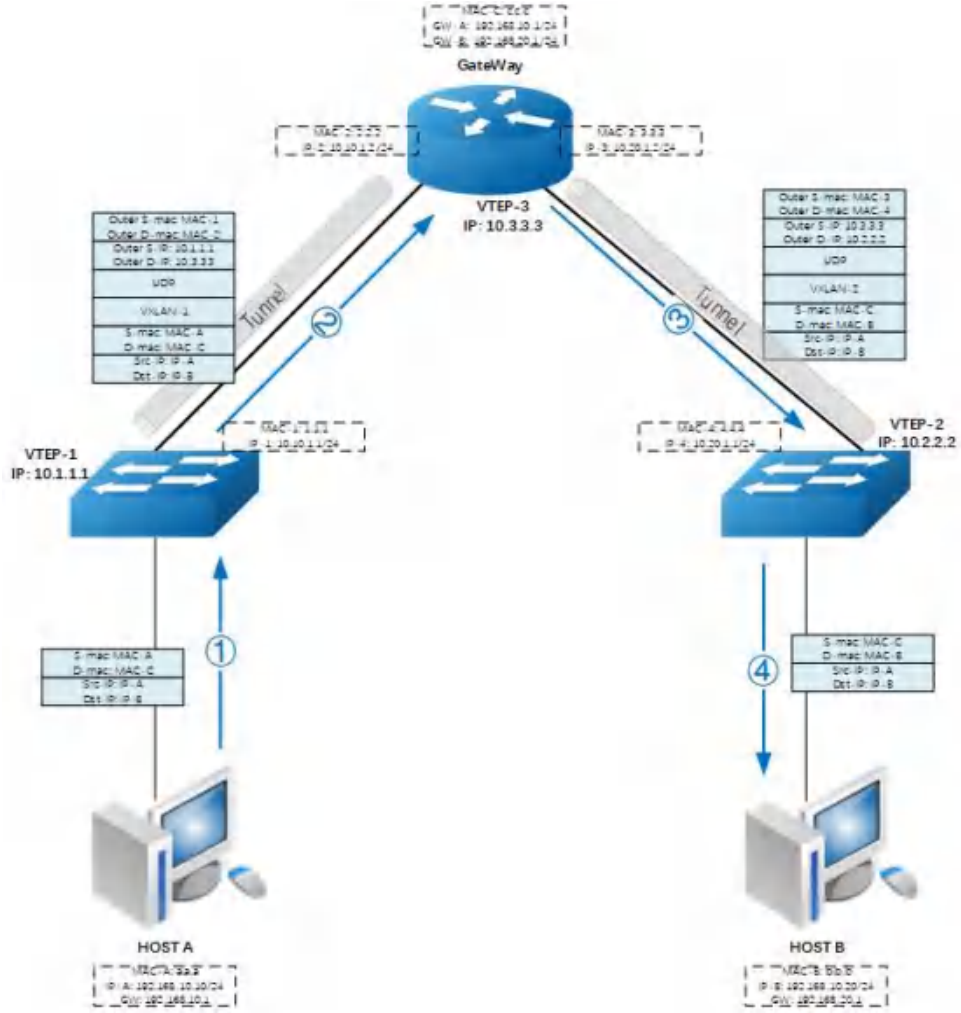


Figure 18-4 Diagram of concentrated gateway

As shown in the figure above, HOST A and B are in different network section, network between them need the help from network gateway. In this situation, it can configure the network gateway onto VTEP-3 intensively, VTEP-1 and VTEP-2 will create VXLAN tunnels to VTEP-3 separately, and then the administrators can make managements intensively.

- HOST A sends arp-request to request the mac address of gateway, VTEP-1 will do vxlan encapsulation to this arp-requestand then forward to VTEP-3 as unknown unicast traffic.
- VTEP-3 will determine whether itself has the gateway address that the packet requested after it receives the arp-request,if it does then it replies arp-reply with vxlan encapsulation to HOST A and learn arp entry of HOST A, its outbound interface is vxlan tunnel.
- VTEP-1 will cut the header of vxlan after it receives this arp-reply; it will forward the original arp-reply to HOST A and learn the mac address of gateway and its outbound interface is vxlan tunnel.
- When HOST A receives the arp-reply, it will learn the mac address of gateway and starts sending data packet. The format of data packet is shown in Figure 2-4.
- Gateway will determine if it needs to do routing forward or not after it receives this packet; however, there is no HOST B's arp table items at this time. Therefore, it will send arp-request to request the arp of HOST B, this arp-request will add the vxlan header to VTEP-2.
- After VTEP-2 receiving the arp-request, it decapsultes and forwards to HOST B and learns the gateway's mac address, its outbound interface is vxlan tunnel.

- HOST B replys arp-reply, its destination mac is the mac of gateway; it will search and forward on VTEP-2, and finds out the outbound interface is vxlan tunnel, hence VTEP-2 will do vxlan encapsulation of this packet and send it to gateway.
- After gateway receives this packet, it learns the arp of HOST B, and then forwards the data packet to HOST B through vxlan tunnel.

Forwarding Theory - Distributed VXLAN network gateway

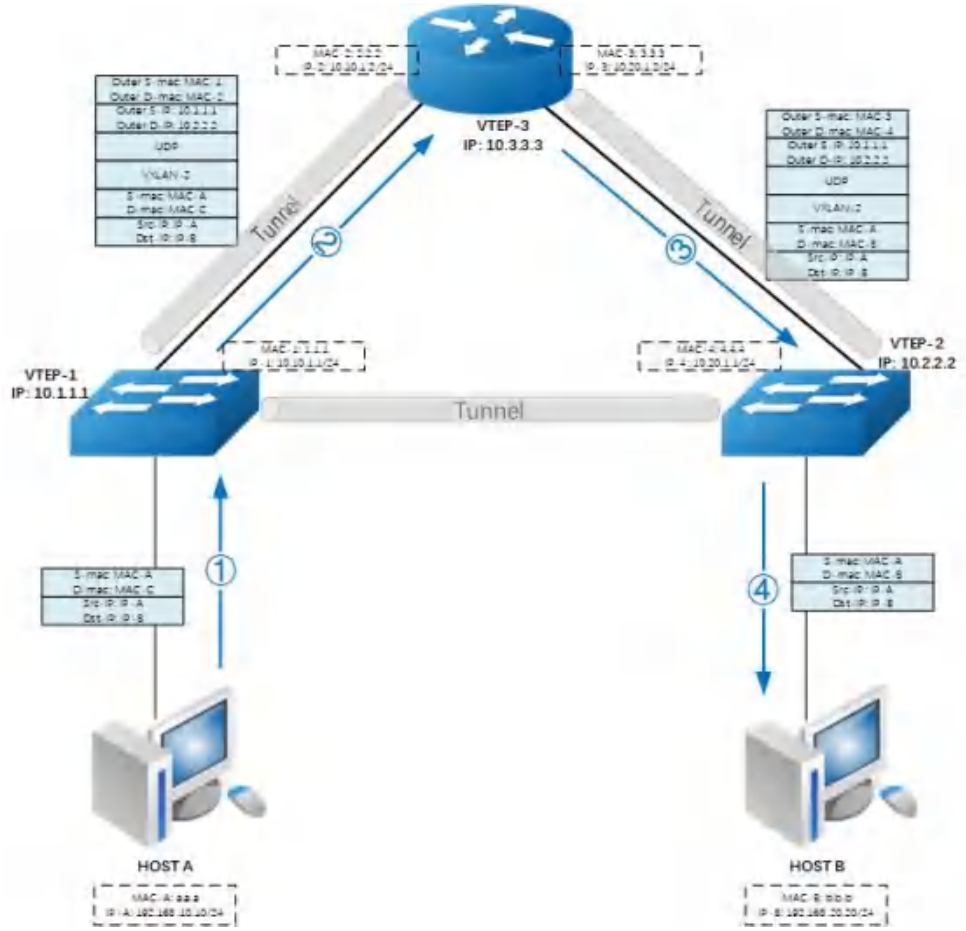


Figure 18-5 Diagram of distributed gateway

Rather than the routing function of concentrated gateway is done by a certain device, distributed gateway can spread the routing function into each VTEP it connects to, which will reduce the pressure of concentrated network gateway.

As shown in the figure above, network knows the IP-B is the host connects to VTEP-2, and the VXLAN ID is VXLAN-2. According to the information above, we can configure routing table on inbound VTEP (which is VTEP-1 in Figure 5), directly encapsulate packets and sent to IP-B , to reduce the pressure of gateway and reduce the delay time of forwarding at the same time. The detailed forwarding process is:

- HOST A sends packet Dst-ip as IP-B, since IP-B and this ip (IP-A) are not in the same network sections, hence this Dst-mac is gateway's mac (Mac-C)
- When the packet gets to VTEP-1, VTEP-1 finds that the Dst-mac is the gateway mac in the network of VLAN, and then it will search whether it has the routing information of IP-B in routing table. If it exsits: then forwarding by the routing table, this route entry includes the VTEP and VXLAN ID information where IP-B is and the corresponding mac (Mac-B) information of IP-B. It will change the original packet's Dst-mac to Mac-B on VTEP-1 and add the relevant encapsulation depends on outer packet information and then precede forwarding; if it does not exsit: then discard this packet
- Packet will forward according to outer packet in the network.
- Packet decapsulates when it reaches to VETP-2, the process of decapsulation is the same as Layer 2 forwarding.

Distributed gateway needs the help from BGP EVPN to synchronize the arp table items on distributed gateway, or static configures DVR routing.

2. Surrounding Features

keep-vlan-tag

By default, it will cut the VLAN tag which the original packet carries when encapsulation and map to outer VXLAN-VNI, and then encapsulated inner packet will not have VLAN tag anymore. If the packet enters the VXLAN network with multiple VLAN tags, it needs to keep the VLAN tag for original packet, the keep-vlan-tag properties should be enabled. For configuration in details, please refer to command line reference guide.

tunnel-aware

Packets that are encapsulated by VXLAN, traditional network devices can only recognize the outer header information of packet s and it will forward by outer header information. If congestions or other situations happen in the network, since it cannot recognize the inner information of packet s, it will not locate which virtual machine or server generates the congestion accurately. After enabling tunnel-aware function, it will allow switches to analyze the inner packet information of original packet, and then use this information on ACL, Flow tracing and so on, which will analyze network traffic accurately.

Distributed-Gatway

In distributed network gateway network, users can configure the gateway of server or virtual machine on to the VTEP which is directly connected. Networking between each virtual machine in different network sections will not use concentrated gateway anymore in the same VTEP. For example, cooperation with DVR routing will not deploy concentrated gateway, the detailed configuration method can be referred to the cases in chapter "Configuration Examples" of this document.

split-horizon

By default, we suggest that configure vxlan tunnels between vtep's full mesh, the traffic from one tunnel cannot enter into another tunnel; if it has requirements of forwarding traffic with each other between tunnels, then disable the split-horizon on tunnels.

Source address of tunnel

By specifying sec-ip address when create tunnels, it can use different source IP address for different tunnels, a switch can accept up to four different source addresses.

DSCP strategy

When encapsulate, users can configure generate strategy of encapsulating packets' DSCP, includes following options： fixed value, copy from original packet, map by priority of packet. The default option is copy from original packet.

18.1.2 Configuration

Vxlan Configuration

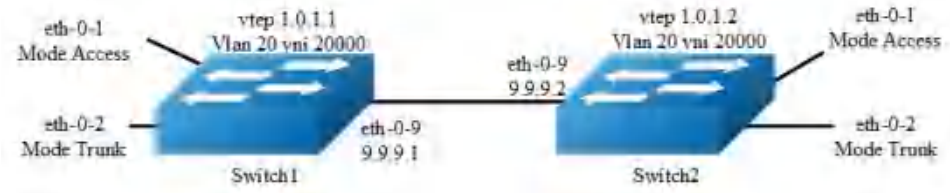


Figure 18-6 Vxlan

In the following example, switch1 and swith2 are connected via layer 3 route. The traffic of vlan 20 are encapsulated in vni 20000, in order to pass through the layer 3 networks.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.1/32
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.2/32
Switch(config-if)# exit
```

Step 4 Create a static route

Configuring Switch1:

```
Switch(config)# ip route 1.0.1.2/32 9.9.9.2
```

Configuring Switch2:

```
Switch(config)# ip route 1.0.1.1/32 9.9.9.1
```

Step 5 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.2 type vxlan
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.1 type vxlan
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.0.1.1
------------------------------------------------------------------------------
VLAN ID                  : 20
VNI                      : 20000
EVPN Tunnel Data-fdb Learning : Enable
Remote VTEP NUM          : 1
      Index: 1, Ip address: 1.0.1.2, Source ip: 1.0.1.1, Type: VxLAN, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------------------
```

Display the result on Switch2:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.0.1.2
------------------------------------------------------------------------------
VLAN ID                  : 20
VNI                      : 20000
EVPN Tunnel Data-fdb Learning : Eanble
Remote VTEP NUM          : 1
      Index: 1, Ip address: 1.0.1.1, Source ip: 1.0.1.2, Type: VxLAN, Protocol: Static
```

```
DVR Gateway NUM: 0
--------------------------------------------------------------------------------
```

Configuring VXLAN Routing

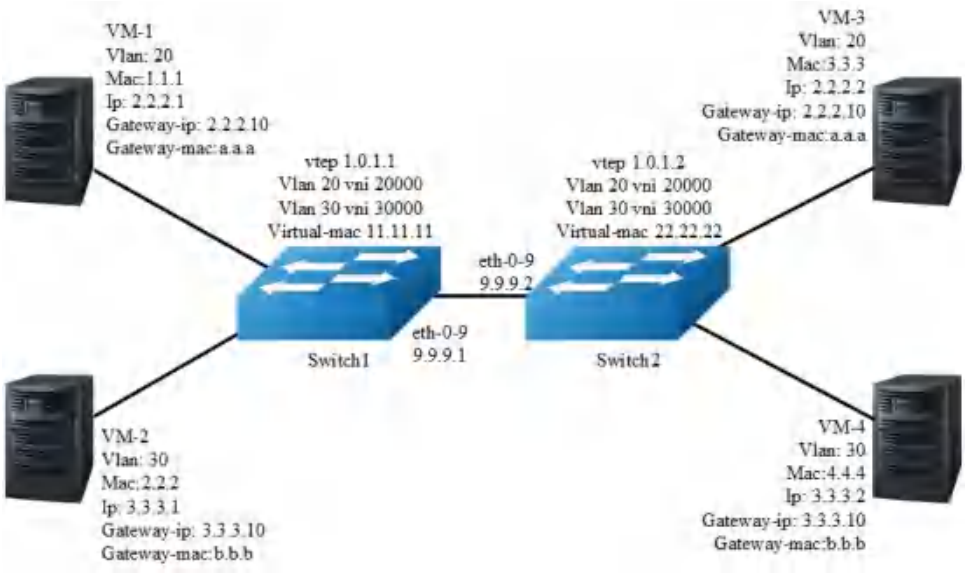


Figure 18-7 Vxlan Routing

In the following example, VM-1 & VM-3 are encapsulated in same vni to make up the distributed route via vxlan; VM-2 & VM-4 are encapsulated in another vni to make up the distributed route via vxlan.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20,30
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# vlan 30 overlay enable
Switch(config-vlan)# exit
```

step 3 Create a vrf instance

```
Switch(config)# ip vrf tenant
Switch(config-vrf)# exit
```

step 4 Create the layer 3 interface and set the ip address

Configuring Switch1:

```
Switch(config)# interface vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.111/24
Switch(config-if)# exit

Switch(config)# interface vlan 30
Switch(config-if)# ip vrf forwarding tenant
```

```
Switch(config-if)# ip address 3.3.3.111/24
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface  vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.222/24
Switch(config-if)# exit

Switch(config)# interface  vlan 30
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 3.3.3.222/24
Switch(config-if)# exit
```

step 5 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add  30
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Configuring Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.1/32
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.2/32
Switch(config-if)# exit
```

Step 6 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.2 type vxlan
Switch(config-overlay)# remote-vtep 1 virtual-mac 22.22.22
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 30 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.1 type vxlan
Switch(config-overlay)# remote-vtep 1 virtual-mac 11.11.11
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 30 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

step 7 Create a static routes and vxlan routes

Configuring Switch1:

```
Switch(config)# ip route 1.0.1.2/32 9.9.9.2
Switch(config)# ip route vrf tenant 2.2.2.2/32 remote-vtep 1 vni 20000 inner-macda 3.3.3
Switch(config)# ip route vrf tenant 3.3.3.2/32 remote-vtep 1 vni 30000 inner-macda 4.4.4
```

Configuring Switch2:

```
Switch(config)# ip route 1.0.1.1/32 9.9.9.1
Switch(config)# ip route vrf tenant 2.2.2.1/32 remote-vtep 1 vni 20000 inner-macda 1.1.1
Switch(config)# ip route vrf tenant 3.3.3.1/32 remote-vtep 1 vni 30000 inner-macda 2.2.2
```

step 8 Exit the configure mode

```
Switch(config)# end
```

step 9 Validation

Display the result on Switch1:

```
Switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S     2.2.2.2/32 is in overlay remote vxlan vtep:1.0.1.1->1.0.1.2, vni:20000
S     3.3.3.2/32 is in overlay remote vxlan vtep:1.0.1.1->1.0.1.2, vni:30000
```

Display the result on Switch2:

```
Switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S     2.2.2.1/32 is in overlay remote vxlan vtep:1.0.1.2->1.0.1.1, vni:20000
S     3.3.3.1/32 is in overlay remote vxlan vtep:1.0.1.2->1.0.1.1, vni:30000
```

Configuring VXLAN Distributed Routing by EBGP EVPN

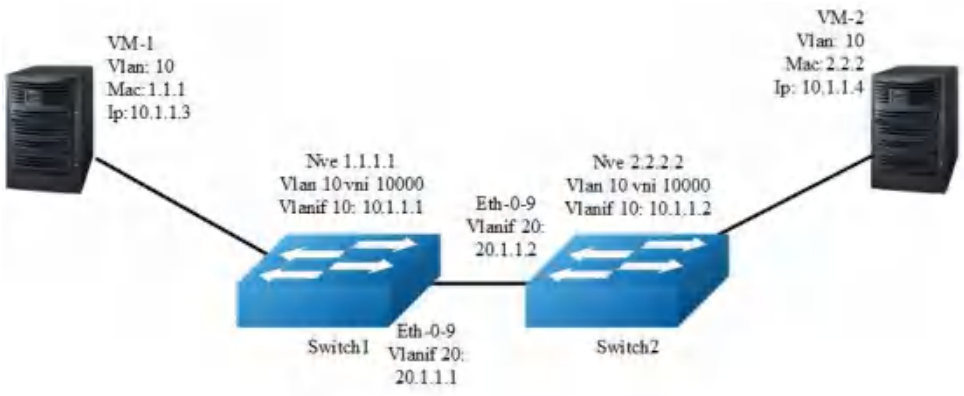


Figure 18-8 EBGP_EVPN

In the following example, VM-1 & VM-2 are encapsulated in same vni to make up the distributed route via vxlan by EBGP EVPN for sending vxlan tunnel and host information;

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10, 20
Switch(config-vlan)# vlan 10 overlay enable
Switch(config-vlan)# exit
```

option: enable arp broadcast suppress for vlan

```
Switch(config-vlan)# vlan 10 arp-broadcast-suppress enable
```

step 3 Create vlan mapping vni for vxlan

```
Switch(config)# overlay
Switch(config-overlay)# vlan 10 vni 10000
Switch(config-vlan)# exit
```

option: Disable inner fdb learning for overlay

```
Switch(config-overlay)# vlan 10 mac-address-tunnel learning-disable
```

step 4 Create evpn instance

```
Switch(config)# evpn
Switch(config-evpn)# vni 10000
Switch(config-evi)# rd auto
Switch(config-evi)# route-target both 1:10000
Switch(config-evi)# exit
```

step 5 Create a vrf instance, and enable EVPN

Configuring Switch1:

```
Switch1(config)# ip vrf tenant
Switch1(config-vrf)# rd 1:20000
Switch1(config-vrf)# route-target both 1:10000 evpn
Switch1(config-vrf)# vxlan vni 20000
Switch1(config-vrf)# exit
```

Configuring Switch2:

```
Switch2(config)# ip vrf tenant
Switch2(config-vrf)# rd 2:20000
Switch2(config-vrf)# route-target both 1:10000 evpn
Switch2(config-vrf)# vxlan vni 20000
Switch2(config-vrf)# exit
```

step 6 Create the layer 3 interface， set the ip address and enable distributed gateway

Configuring Switch1:

```
Switch1(config)# interface  vlan 10
Switch1(config-if)# ip vrf forwarding tenant
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# overlay host-collect enable
Switch1(config-if)# ip address 10.1.1.1/24
Switch1(config-if)# exit
Switch1(config)# interface  vlan 20
Switch1(config-if)# ip address 20.1.1.1/24
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config)# interface  vlan 10
Switch2(config-if)# ip vrf forwarding tenant
Switch2(config-if)# overlay distributed-gateway enable
Switch2(config-if)# overlay host-collect enable
Switch2(config-if)# ip address 10.1.1.2/24
Switch2(config-if)# exit
Switch2(config)# interface  vlan 20
Switch2(config-if)# ip address 20.1.1.2/24
Switch2(config-if)# exit
```

step 7 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# interface eth-0-9
Switch(config-if)# switchport mode trunk
```

```
Switch(config-if)# switchport trunk allowed vlan add  20
Switch(config-if)# vxlan uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Step 8 Create NVE

Configuring Switch1:

```
Switch1(config)# interface loopback 1
Switch1(config-if)# ip address 1.1.1.1/32
Switch1(config-if)# exit
Switch1(config)# interface nve 1
Switch1(config-if)# source loopback 1
Switch1(config-if)# member vni 10000
Switch1(config-if)# member vni 20000 associate-vrf
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config)# interface loopback 1
Switch2(config-if)# ip address 2.2.2.2/32
Switch2(config-if)# exit
Switch2(config)# interface nve 1
Switch2(config-if)# source loopback 1
Switch2(config-if)# member vni 10000
Switch2(config-if)# member vni 20000 associate-vrf
Switch2(config-if)# exit
```

option: configure the attribute of EVPN tunnel

```
Switch(config-if)# keep-vlan-tag enable
Switch(config-if)# split-horizon disable
Switch(config-if)# encapsulation-dscp-strategy custom-assign 63
Switch(config-if)# virtual-mac a.a.a
```

Step 9 Create BGP EVPN

Configuring Switch1:

```
Switch1(config)# router bgp 100
Switch1(config-router)# neighbor 20.1.1.2 remote-as 200
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 20.1.1.2 activate
Switch1(config-router-af)# neighbor 20.1.1.2 send-community extended
Switch1(config-router-af)# neighbor 20.1.1.2 attribute-unchanged next-hop
Switch1(config-router-af)# exit
Switch1(config-router)# exit
```

Configuring Switch2:

```
Switch2(config)# router bgp 200
Switch2(config-router)# neighbor 20.1.1.1 remote-as 100
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 20.1.1.1 activate
Switch2(config-router-af)# neighbor 20.1.1.1 send-community extended
Switch2(config-router-af)# neighbor 20.1.1.1 attribute-unchanged next-hop
Switch2(config-router-af)# exit
Switch2(config-router)# exit
```

step 10 Create a static routes

Configuring Switch1:

```
Switch1(config)# ip route 2.2.2.2/32 20.1.1.2
```

Configuring Switch2:

```
Switch(config2)# ip route 1.1.1.1/32 20.1.1.1
```

step 11 Exit the configure mode

```
Switch(config)# end
```

step 12 Validation

Display the result on Switch1:

```
Switch1# show bgp evpn all
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:10000 (L2VNI 10000)
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[0.0.0.0]/136
                 1.1.1.1                        32768 i
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[10.1.1.3]/136
                 1.1.1.1                        32768 i
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[0.0.0.0]/136
                 2.2.2.2                        0 200 i
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[10.1.1.4]/136
                 2.2.2.2                        0 200 i
*> [3]:[0]:[32]:[1.1.1.1]/80
                 1.1.1.1                        32768 i
*> [3]:[0]:[32]:[10.20.30.40]/80
                 2.2.2.2                        0 200 i

Route Distinguisher: 1:10000
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[0.0.0.0]/136
                 2.2.2.2                        0 200 i
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[10.1.1.4]/136
                 2.2.2.2                        0 200 i
*> [3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                        0 200 i

Route Distinguisher: 1:20000 (L3VNI 20000)
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[10.1.1.4]/136
                 2.2.2.2

Switch1# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni     Type   Remote-vtep IP-Address     Src-Address     Head-end-flooding Protocol
10   10000   VxLAN  0          2.2.2.2         1.1.1.1         Enable          Evpn
```

Display the result on Switch2:

```
Head-end-floodingSwitch2# show bgp evpn all
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
          S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:10000 (L2VNI 10000)
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[0.0.0.0]/136
                 1.1.1.1                        0 100 i
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[10.1.1.3]/136
                 1.1.1.1                        0 100 i
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[0.0.0.0]/136
                 2.2.2.2                    32768 i
*> [2]:[0]:[48]:[ac7f.1cc5.fe00]:[32]:[10.1.1.4]/136
                 2.2.2.2                    32768 i
*> [3]:[0]:[32]:[1.1.1.1]/80
                 1.1.1.1                        0 100 i
*> [3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                    32768 i

Route Distinguisher: 1:10000
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[0.0.0.0]/136
                 1.1.1.1                        0 100 i
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[10.1.1.3]/136
                 1.1.1.1                        0 100 i
*> [3]:[0]:[32]:[1.1.1.1]/80
                 1.1.1.1                        0 100 i

Route Distinguisher: 2:20000 (L3VNI 20000)
*> [2]:[0]:[48]:[4623.28ef.da00]:[32]:[10.1.1.3]/136
                 1.1.1.1                        0 100 i

Switch2# show overlay tunnel
-------------------------------------------------------------------------------
Vlan Vni     Type   Remote-vtep IP-Address     Src-Address     Head-end-flooding Protocol
10   10000   VxLAN  0          1.1.1.1         2.2.2.2         Enable          Evpn
```

Configuring VXLAN Distributed Routing by IBGP EVPN

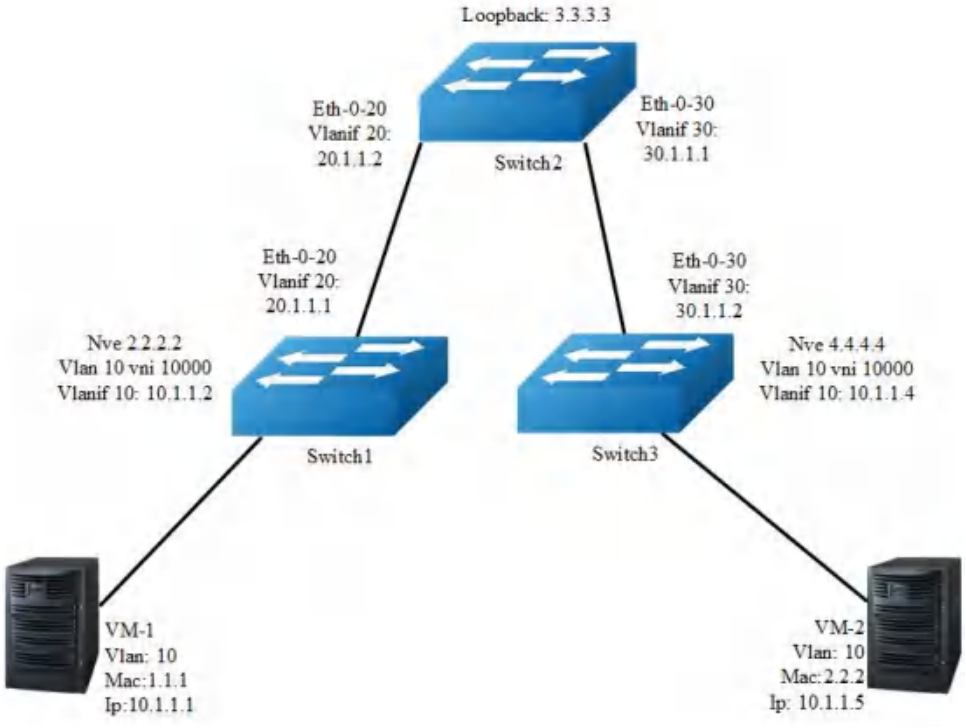


Figure 18-9 IBGP_EVPN

In the following example, VM-1 & VM-2 are encapsulated in same vni to make up the distributed route via vxlan by IBGP EVPN for sending vxlan tunnel and host information;EVPN route is exchanged by bgp route reflector.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

Configuring Switch1:

```
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10, 20
Switch1(config-vlan)# vlan 10 overlay enable
Switch1(config-vlan)# exit
```

Configuring Switch2:

```
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 20, 30
Switch2(config-vlan)# exit
```

Configuring Switch3:

```
Switch3(config)# vlan database
Switch3(config-vlan)# vlan 10, 30
Switch3(config-vlan)# vlan 10 overlay enable
Switch3(config-vlan)# exit
```

option: enable arp broadcast suppress for vlan

```
Switch(config-vlan)# vlan 10 arp-broadcast-suppress enable
```

step 3 Create vlan mapping vni for vxlan

Configuring Switch1:

```
Switch1(config)# overlay
Switch1(config-overlay)# vlan 10 vni 10000
Switch1(config-vlan)# exit
```

option: Disable inner fdb learning for overlay

```
Switch1(config-overlay)# vlan 10 mac-address-tunnel learning-disable
```

Configuring Switch3:

```
Switch3(config)# overlay
Switch3(config-overlay)# vlan 10 vni 10000
Switch3(config-vlan)# exit
```

option: Disable inner fdb learning for overlay

```
Switch3(config-overlay)# vlan 10 mac-address-tunnel learning-disable
```

step 4 Create evpn instance

Configuring Switch1:

```
Switch1(config)# evpn
Switch1(config-evpn)# vni 10000
Switch1(config-evi)# rd 2:2
Switch1(config-evi)# route-target both 20:20
Switch1(config-evi)# exit
```

Configuring Switch2:

```
Switch2(config)# evpn
```

Configuring Switch3:

```
Switch3(config)# evpn
Switch3(config-evpn)# vni 10000
Switch3(config-evi)# rd 4:4
Switch3(config-evi)# route-target both 20:20
Switch3(config-evi)# exit
```

step 5 Create a vrf instance, and enable EVPN

Configuring Switch1:

```
Switch1(config)# ip vrf tenant
Switch1(config-vrf)# rd 22:22
Switch1(config-vrf)# route-target both 20:20 evpn
Switch1(config-vrf)# vxlan vni 20000
Switch1(config-vrf)# exit
```

在 switch3 配置：

Configuring Switch3:

```
Switch3(config)# ip vrf tenant
Switch3(config-vrf)# rd 44:44
Switch3(config-vrf)# route-target both 20:20 evpn
Switch3(config-vrf)# vxlan vni 20000
Switch3(config-vrf)# exit
```

step 6 Create the layer 3 interface， set the ip address and enable distributed gateway

Configuring Switch1:

```
Switch1(config)# interface  vlan 10
Switch1(config-if)# ip vrf forwarding tenant
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# overlay host-collect enable
Switch1(config-if)# ip address 10.1.1.2/24
Switch1(config-if)# exit
Switch1(config)# interface  vlan 20
Switch1(config-if)# ip address 20.1.1.1/24
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config)# interface  vlan 20
Switch2(config-if)# ip address 20.1.1.2/24
Switch2(config-if)# exit
Switch2(config)# interface  vlan 30
Switch2(config-if)# ip address 30.1.1.1/24
Switch2(config-if)# exit
```

Configuring Switch3:

```
Switch3(config)# interface  vlan 10
Switch3(config-if)# ip vrf forwarding tenant
Switch3(config-if)# overlay distributed-gateway enable
Switch3(config-if)# overlay host-collect enable
Switch3(config-if)# ip address 10.1.1.3/24
Switch3(config-if)# exit
Switch3(config)# interface  vlan 30
Switch3(config-if)# ip address 30.1.1.2/24
Switch3(config-if)# exit
```

step 7 Enter the interface configure mode and set the attributes of the interface

Configuring Switch1:

```
Switch1(config)# interface eth-0-10
Switch1(config-if)# switchport access vlan 10
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
Switch1(config)# interface eth-0-20
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add  20
Switch1(config-if)# vxlan uplink enable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config)# interface eth-0-20
Switch2(config-if)# switchport mode trunk
Switch2(config-if)# switchport trunk allowed vlan add  20
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface eth-0-30
Switch2(config-if)# switchport mode trunk
```

```
Switch2(config-if)# switchport trunk allowed vlan add  30
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
```

Configuring Switch3:

```
Switch3(config)# interface eth-0-10
Switch3(config-if)# switchport access vlan 10
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# interface eth-0-30
Switch3(config-if)# switchport mode trunk
Switch3(config-if)# switchport trunk allowed vlan add  30
Switch3(config-if)# vxlan uplink enable
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
```

Step 8 Create NVE

Configuring Switch1:

```
Switch1(config)# interface loopback 2
Switch1(config-if)# ip address 2.2.2.2/32
Switch1(config-if)# exit
Switch1(config)# interface nve 1
Switch1(config-if)# source 2.2.2.2
Switch1(config-if)# member vni 10000
Switch1(config-if)# member vni 20000 associate-vrf
Switch1(config-if)# exit
```

Configuring Switch2:

```
Switch2(config)# interface loopback 3
Switch2(config-if)# ip address 3.3.3.3/32
Switch2(config-if)# exit
```

Configuring Switch3:

```
Switch3(config)# interface loopback 4
Switch3(config-if)# ip address 4.4.4.4/32
Switch3(config-if)# exit
Switch3(config)# interface nve 1
Switch3(config-if)# source 4.4.4.4
Switch3(config-if)# member vni 10000
Switch3(config-if)# member vni 20000 associate-vrf
Switch3(config-if)# exit
```

option: configure the attribute of EVPN tunnel

```
Switch(config-if)# keep-vlan-tag enable
Switch(config-if)# split-horizon disable
Switch(config-if)# encapsulation-dscp-strategy custom-assign 63
Switch(config-if)# virtual-mac a.a.a
```

Step 9 Create BGP EVPN

Configuring Switch1:

```
Switch1(config)# router bgp 100
Switch1(config-router)# neighbor 3.3.3.3 remote-as 100
```

```
Switch1(config-router)# neighbor 3.3.3.3 update-source loopback2
Switch1(config-router)# neighbor 20.1.1.2 remote-as 100
Switch1(config-router)# address-family ipv4
Switch1(config-router-af)# network 2.2.2.2 mask 255.255.255.255
Switch1(config-router-af)# neighbor 20.1.1.2 weight 32768
Switch1(config-router-af)# exit
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 3.3.3.3 activate
Switch1(config-router-af)# neighbor 3.3.3.3 send-community extended
Switch1(config-router-af)# exit
Switch1(config-router)# exit
```

Configuring Switch2:

```
Switch2(config)# router bgp 100
Switch2(config-router)# neighbor 2.2.2.2 remote-as 100
Switch2(config-router)# neighbor 2.2.2.2 update-source loopback3
Switch2(config-router)# neighbor 4.4.4.4 remote-as 100
Switch2(config-router)# neighbor 4.4.4.4 update-source loopback3
Switch2(config-router)# neighbor 20.1.1.1 remote-as 100
Switch2(config-router)# neighbor 30.1.1.2 remote-as 100
Switch2(config-router)# address-family ipv4
Switch2(config-router-af)# network 3.3.3.3 mask 255.255.255.255
Switch2(config-router-af)# network 20.1.1.0 mask 255.255.255.0
Switch2(config-router-af)# network 30.1.1.0 mask 255.255.255.0
Switch2(config-router-af)# neighbor 20.1.1.1 weight 32768
Switch2(config-router-af)# neighbor 20.1.1.1 route-reflector-client
Switch2(config-router-af)# neighbor 30.1.1.2 weight 32768
Switch2(config-router-af)# neighbor 30.1.1.2 route-reflector-client
Switch2(config-router-af)# exit
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 2.2.2.2 activate
Switch2(config-router-af)# neighbor 2.2.2.2 route-reflector-client
Switch2(config-router-af)# neighbor 2.2.2.2 send-community extended
Switch2(config-router-af)# neighbor 4.4.4.4 activate
Switch2(config-router-af)# neighbor 4.4.4.4 route-reflector-client
Switch2(config-router-af)# neighbor 4.4.4.4 send-community extended
Switch2(config-router-af)# exit
Switch2(config-router)# exit
```

Configuring Switch3:

```
Switch3(config)# router bgp 100
Switch3(config-router)# neighbor 3.3.3.3 remote-as 100
Switch3(config-router)# neighbor 3.3.3.3 update-source loopback4
Switch3(config-router)# neighbor 30.1.1.1 remote-as 100
Switch3(config-router)# address-family ipv4
Switch3(config-router-af)# network 4.4.4.4 mask 255.255.255.255
Switch3(config-router-af)# neighbor 30.1.1.1 weight 32768
Switch3(config-router-af)# exit
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 3.3.3.3 activate
Switch3(config-router-af)# neighbor 3.3.3.3 send-community extended
Switch3(config-router-af)# exit
Switch3(config-router)# exit
```

step 10 Exit the configure mode

```
Switch(config)# end
```

step 11 Validation

Display the result on Switch1:

```
Switch1# show bgp evpn all
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 2:2 (L2VNI 10000)
*> [2]:[0]:[48]:[988b.123a.4000]:[32]:[0.0.0.0]/136
                 2.2.2.2                        32768 i
*> [2]:[0]:[48]:[988b.123a.4000]:[32]:[10.1.1.1]/136
                 2.2.2.2                        32768 i
*> [3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                        32768 i
*>i[3]:[0]:[32]:[4.4.4.4]/80
                 4.4.4.4                   100     0 i

Route Distinguisher: 4:4
*>i[3]:[0]:[32]:[4.4.4.4]/80
                 4.4.4.4                   100     0 i


Switch1# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni     Type   Remote-vtep IP-Address     Src-Address     Head-end-flooding Protocol
10   10000   VxLAN  0          4.4.4.4         2.2.2.2         Enable          Evpn
```

Display the result on Switch2:

```
Switch2# show bgp evpn all
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 2:2
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[0.0.0.0]/136
                 2.2.2.2                   100     0 i
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[10.1.1.1]/136
                 2.2.2.2                   100     0 i
*>i[3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                   100     0 i

Route Distinguisher: 4:4
*>i[3]:[0]:[32]:[4.4.4.4]/80
                 4.4.4.4                   100     0 i
```

Display the result on Switch3:

```
Switch3# show bgp evpn all
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
          S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 2:2
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[0.0.0.0]/136
                 2.2.2.2                  100     0 i
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[10.1.1.1]/136
                 2.2.2.2                  100     0 i
*>i[3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                  100     0 i

Route Distinguisher: 4:4 (L2VNI 10000)
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[0.0.0.0]/136
                 2.2.2.2                  100     0 i
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[10.1.1.1]/136
                 2.2.2.2                  100     0 i
*>i[3]:[0]:[32]:[2.2.2.2]/80
                 2.2.2.2                  100     0 i
*> [3]:[0]:[32]:[4.4.4.4]/80
                 4.4.4.4                    32768 i

Route Distinguisher: 44:44 (L3VNI 20000)
*>i[2]:[0]:[48]:[988b.123a.4000]:[32]:[10.1.1.1]/136
                 2.2.2.2                  100     0 i

Switch3# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni     Type   Remote-vtep IP-Address     Src-Address     Head-end-flooding Protocol
10   10000   VxLAN  0          2.2.2.2         4.4.4.4         Enable          Evpn

Switch3# show mac address-table
       Mac Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
Vlan   Mac Address      Type       Ports
----   -----------      --------   -----
30     fcc0.9318.0a00   dynamic    eth-0-9
10     988b.123a.4000   dynamic    VxLAN: 4.4.4.4->2.2.2.2(EI)

Switch3# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C       10.1.1.0/24 is directly connected, vlan10
C       10.1.1.3/32 is in local loopback, vlan10
B       10.1.1.1/32 is in overlay remote vxlan vtep:4.4.4.4->2.2.2.2, vni:20000
```

Users In the same network segment communicate through VXLAN Network

1. Topology

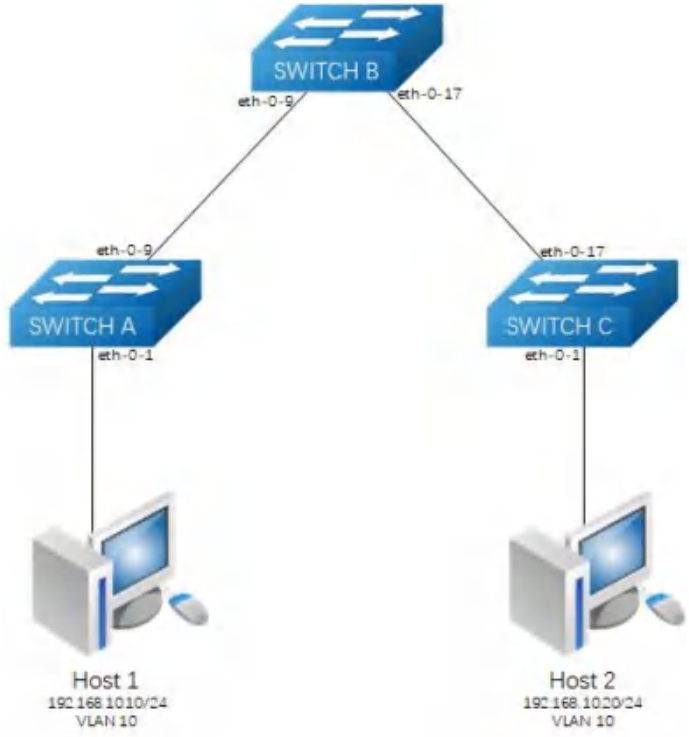


Figure 18-10 VXLAN network communication

2. Requirement

Host 1 and Host 2 are in the same segment, they need to realize interflow by VXLAN tunnels.

3. Configuration Steps

4. step 1 Configure routing on SWITCH A/SWITCH B/SWITCH C, make sure that these three switches can interflow with each other in Layer 3 network

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# interface loopback 0
SWITCH_A(config-if)# ip address 10.1.1.1/32
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface eth-0-9
SWITCH_A(config-if)# no switchport
SWITCH_A(config-if)# ip address 192.168.1.1/24
SWITCH_A(config-if)# vxlan uplink enable
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# exit
SWITCH_A(config)# ip route 10.3.3.3/32 192.168.1.2
SWITCH_A(config)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# interface eth-0-9
```

```
SWITCH_B(config-if)# no switchport
SWITCH_B(config-if)# ip address 192.168.1.2/24
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-17
SWITCH_B(config-if)# no switchport
SWITCH_B(config-if)# ip address 192.168.2.1/24
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# ip route 10.1.1.1/32 192.168.1.1
SWITCH_B(config)# ip route 10.3.3.3/32 192.168.2.2
SWITCH_B(config)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# interface loopback 0
SWITCH_C(config-if)# ip address 10.3.3.3/32
SWITCH_C(config-if)# exit
SWITCH_C(config)# interface eth-0-17
SWITCH_C(config-if)# no switchport
SWITCH_C(config-if)# ip address 192.168.2.2/24
SWITCH_C(config-if)# vxlan uplink enable
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# exit
SWITCH_C(config)# ip route 10.1.1.1/32 192.168.2.1
SWITCH_C(config)# end
```

step 2 Configure VLAN on SWITCH A/SWITCH C

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# vlan database
SWITCH_A(config-vlan)# vlan 10
SWITCH_A(config-vlan)# vlan 10 overlay enable
SWITCH_A(config-vlan)# exit
SWITCH_A(config)# interface eth-0-1
SWITCH_A(config-if)# switchport mode access
SWITCH_A(config-if)# switchport access vlan 10
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# vlan database
SWITCH_C(config-vlan)# vlan 10
SWITCH_C(config-vlan)# vlan 10 overlay enable
SWITCH_C(config-vlan)# exit
SWITCH_C(config)# interface eth-0-1
SWITCH_C(config-if)# switchport mode access
SWITCH_C(config-if)# switchport access vlan 10
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# end
```

step 3 Configure VXLAN tunnels on SWITCH A/SWITCH C

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# overlay
SWITCH_A(config-overlay)# source 10.1.1.1
SWITCH_A(config-overlay)# remote-vtep 1 ip-address 10.3.3.3 type vxlan
SWITCH_A(config-overlay)# vlan 10 vni 10000
SWITCH_A(config-overlay)# vlan 10 remote-vtep 1
SWITCH_A(config-overlay)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# overlay
SWITCH_C(config-overlay)# source 10.3.3.3
SWITCH_C(config-overlay)# remote-vtep 1 ip-address 10.1.1.1 type vxlan
SWITCH_C(config-overlay)# vlan 10 vni 10000
SWITCH_C(config-overlay)# vlan 10 remote-vtep 1
SWITCH_C(config-overlay)# end
```

step 4 Validation

Check whether it can communicate with the networks

```
SWITCH_A# ping
Protocol [ip]:
Target IP address: 10.3.3.3
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Data pattern [0xABCD]:
PATTERN: 0xabcd
PING 10.3.3.3 (10.3.3.3) from 10.1.1.1 : 100(128) bytes of data.
108 bytes from 10.3.3.3: icmp_seq=0 ttl=63 time=775 ms
108 bytes from 10.3.3.3: icmp_seq=1 ttl=63 time=904 ms
108 bytes from 10.3.3.3: icmp_seq=2 ttl=63 time=768 ms
108 bytes from 10.3.3.3: icmp_seq=3 ttl=63 time=668 ms
108 bytes from 10.3.3.3: icmp_seq=4 ttl=63 time=723 ms

--- 10.3.3.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 668.636/768.201/904.300/78.078 ms, pipe 2
```

Check whether the configurations of the tunnels are correct

```
SWITCH_A# show overlay vlan 10
------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 10.1.1.1
------------------------------------------------------------
VLAN ID       : 10
```

```
VNI        : 10000
Remote VTEP NUM: 1
      Index: 1, Ip address: 10.3.3.3,  Type: VxLAN
DVR Gateway NUM: 0
------------------------------------------------------------------
SWITCH_A#
SWITCH_A# show overlay uplink
------------------------------------------------------------------
Uplink port:
  eth-0-9
------------------------------------------------------------------
```

Concentrated VXLAN Gateway

1. Topology

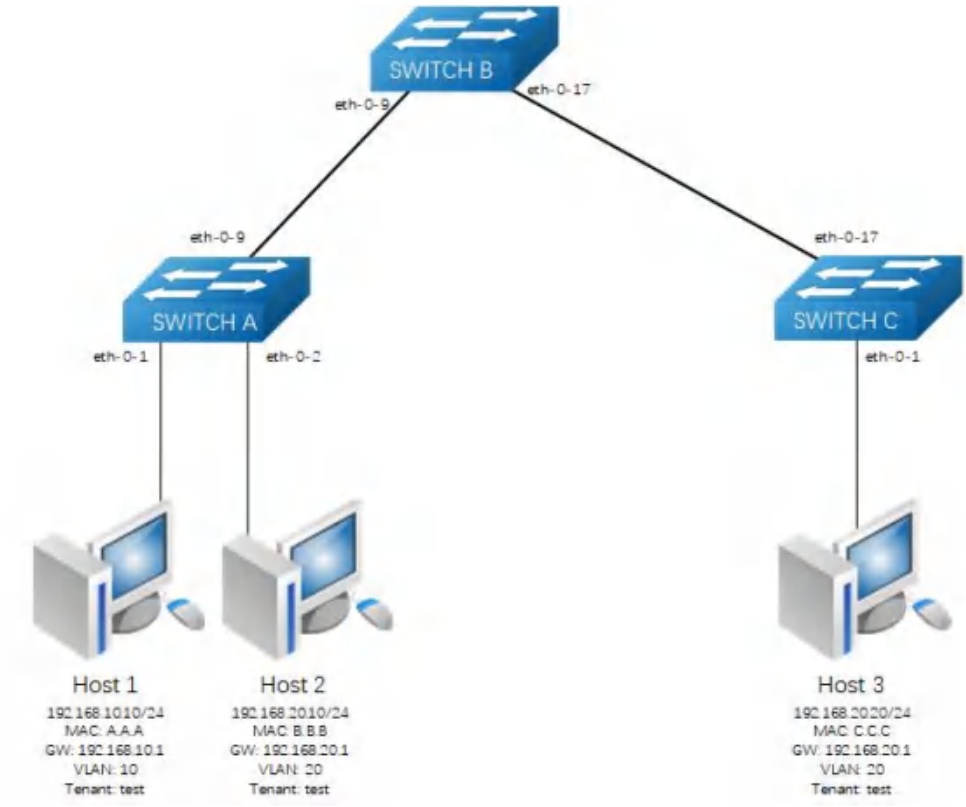


Figure 18-11 Concentrated VXLAN gateway networking diagram

2. Requirement

As shown in the figure, HOST1/2/3 are all belong to tenement test, it has the same segments and different segments, and they all need interflow with each other. Between Switch A Switch B and Switch C is Layer 3 network, so it needs to create tunnels by configuring VXLAN between HOST, and realizes networking between HOST by concentrated network gateway.


It can select SWITCH B as the concentrated network gateway.

3. Configuration Steps

4. step 1 Configure routing on SWITCH A/SWITCH B/SWITCH C, make sure that these three switches can interflow with each other in Layer 3 network

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# interface loopback 0
SWITCH_A(config-if)# ip address 10.1.1.1/32
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface eth-0-9
SWITCH_A(config-if)# no switchport
SWITCH_A(config-if)# ip address 192.168.9.1/24
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# exit
SWITCH_A(config)# ip route 10.2.2.2/32 192.168.9.2
SWITCH_A(config)# ip route 10.3.3.3/32 192.168.9.2
SWITCH_A(config)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# interface loopback 0
SWITCH_B(config-if)# ip address 10.2.2.2/32
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-9
SWITCH_B(config-if)# no switchport
SWITCH_B(config-if)# ip address 192.168.9.2/24
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-17
SWITCH_B(config-if)# no switchport
SWITCH_B(config-if)# ip address 192.168.17.2/24
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# ip route 10.3.3.3/32 192.168.17.1
SWITCH_B(config)# ip route 10.1.1.1/32 192.168.9.1
SWITCH_B(config)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# interface loopback 0
SWITCH_C(config-if)# ip address 10.3.3.3/32
SWITCH_C(config-if)# exit
SWITCH_C(config)# interface eth-0-17
SWITCH_C(config-if)# no switchport
SWITCH_C(config-if)# ip address 192.168.17.1/24
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# exit
SWITCH_C(config)# ip route 10.1.1.1/32 192.168.17.2
SWITCH_C(config)# ip route 10.2.2.2/32 192.168.17.2
SWITCH_C(config)# end
```

step 2 Configure VLAN on SWITCH A SWITCH B and SWITCH C

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
SWITCH_A(config)# vlan database
SWITCH_A(config-vlan)# vlan 10,20
SWITCH_A(config-vlan)# vlan 10 overlay enable
SWITCH_A(config-vlan)# vlan 20 overlay enable
SWITCH_A(config-vlan)# exit
SWITCH_A(config)# interface eth-0-1
SWITCH_A(config-if)# switchport mode access
SWITCH_A(config-if)# switchport access vlan 10
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface eth-0-2
SWITCH_A(config-if)# switchport mode access
SWITCH_A(config-if)# switchport access vlan 20
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# vlan database
SWITCH_B(config-vlan)# vlan 10,20
SWITCH_B(config-vlan)# vlan 10 overlay enable
SWITCH_B(config-vlan)# vlan 20 overlay enable
SWITCH_B(config-vlan)# exit
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# vlan database
SWITCH_C(config-vlan)# vlan 20
SWITCH_C(config-vlan)# vlan 20 overlay enable
SWITCH_C(config-vlan)# exit
SWITCH_C(config)# interface eth-0-1
SWITCH_C(config-if)# switchport mode access
SWITCH_C(config-if)# switchport access vlan 20
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# exit
```

step 3 Create VXLAN tunnels between SWITCH A SWITCH B and SWITCH C

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# interface eth-0-9
SWITCH_A(config-if)# vxlan uplink enable
SWITCH_A(config-if)# exit
SWITCH_A(config)# overlay
SWITCH_A(config-overlay)# source 10.1.1.1
SWITCH_A(config-overlay)# remote-vtep 1 ip-address 10.2.2.2 type vxlan
SWITCH_A(config-overlay)# remote-vtep 2 ip-address 10.3.3.3 type vxlan
SWITCH_A(config-overlay)# vlan 10 vni 10000
SWITCH_A(config-overlay)# vlan 10 remote-vtep 1
SWITCH_A(config-overlay)# vlan 20 vni 20000
SWITCH_A(config-overlay)# vlan 20 remote-vtep 1
```

```
SWITCH_A(config-overlay)# vlan 20 remote-vtep 2
SWITCH_A(config-overlay)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# interface eth-0-9
SWITCH_B(config-if)# vxlan uplink enable
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-17
SWITCH_B(config-if)# vxlan uplink enable
SWITCH_B(config-if)# exit
SWITCH_B(config)# overlay
SWITCH_B(config-overlay)# source 10.2.2.2
SWITCH_B(config-overlay)# remote-vtep 1 ip-address 10.1.1.1 type vxlan
SWITCH_B(config-overlay)# remote-vtep 2 ip-address 10.3.3.3 type vxlan
SWITCH_B(config-overlay)# vlan 10 vni 10000
SWITCH_B(config-overlay)# vlan 10 remote-vtep 1
SWITCH_B(config-overlay)# vlan 20 vni 20000
SWITCH_B(config-overlay)# vlan 20 remote-vtep 1
SWITCH_B(config-overlay)# vlan 20 remote-vtep 2
SWITCH_B(config-overlay)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# interface eth-0-17
SWITCH_C(config-if)# vxlan uplink enable
SWITCH_C(config-if)# exit
SWITCH_C(config)# overlay
SWITCH_C(config-overlay)# source 10.3.3.3
SWITCH_C(config-overlay)# remote-vtep 1 ip-address 10.1.1.1 type vxlan SWITCH_C(config-overlay)# remote-vtep 2 ip-address
10.2.2.2 type vxlan
SWITCH_C(config-overlay)# vlan 20 vni 20000
SWITCH_C(config-overlay)# vlan 20 remote-vtep 1
SWITCH_C(config-overlay)# vlan 20 remote-vtep 2
SWITCH_C(config-overlay)# end
```

step 4 Configure corresponding HOST gateway on SWITCH B, and bind to relevant VRF

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# ip vrf test
SWITCH_B(config-vrf)# exit
SWITCH_B(config)# interface vlan 10
SWITCH_B(config-if)# ip vrf forwarding test
SWITCH_B(config-if)# ip address 192.168.10.1/24
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface vlan 20
SWITCH_B(config-if)# ip vrf forwarding test
SWITCH_B(config-if)# ip address 192.168.20./24
SWITCH_B(config-if)# end
```

The configuration is done at this point; HOST 1/2/3/4 can interflow with each other by ping.

 In this case VRF is used to separate different tenants. The vrf configuration can be removed if it is unnecessary.

step 5 Validation

Check whether it can communicate with the networks

```
SWITCH_A# ping -a 10.1.1.1 10.2.2.2
PING 10.2.2.2 (10.2.2.2) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.2.2.2: icmp_seq=1 ttl=64 time=2.57 ms
64 bytes from 10.2.2.2: icmp_seq=2 ttl=64 time=2.05 ms
64 bytes from 10.2.2.2: icmp_seq=3 ttl=64 time=2.29 ms
64 bytes from 10.2.2.2: icmp_seq=4 ttl=64 time=2.07 ms
64 bytes from 10.2.2.2: icmp_seq=5 ttl=64 time=2.44 ms

--- 10.2.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 2.059/2.287/2.573/0.210 ms

SWITCH_A# ping -a 10.1.1.1 10.3.3.3
PING 10.3.3.3 (10.3.3.3) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.3.3.3: icmp_seq=1 ttl=63 time=3.32 ms
64 bytes from 10.3.3.3: icmp_seq=2 ttl=63 time=2.45 ms
64 bytes from 10.3.3.3: icmp_seq=3 ttl=63 time=2.50 ms
64 bytes from 10.3.3.3: icmp_seq=4 ttl=63 time=2.42 ms
64 bytes from 10.3.3.3: icmp_seq=5 ttl=63 time=3.49 ms

--- 10.3.3.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 2.421/2.839/3.491/0.467 ms
```

Check whether the configurations of the tunnels are correct

```
SWITCH_B# show overlay
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 10.1.1.1

Vlan Vni     Type   Remote-vtep IP-Address     Src-Address     Head-end-flooding Protocol
------------------------------------------------------------------------------
10   10000     VxLAN  1         10.2.2.2    10.1.1.1    Enable         Static
20   20000     VxLAN  1         10.2.2.2    10.1.1.1    Enable         Static
20   20000     VxLAN  2         10.3.3.3    10.1.1.1    Enable         Static
SWITCH_A# show overlay uplink
------------------------------------------------------------------------------
Uplink port:
  eth-0-9
------------------------------------------------------------------------------S
```

Check the ARP tables on SWITCH B (Gateway)

```
SWITCH_B # show ip arp
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   192.168.9.2           -  001e.081b.bce0  eth-0-9
 Internet   192.168.9.1           0  001e.080a.a7fb  eth-0-9
 Internet   192.168.17.2          -  001e.081b.bce0  eth-0-17
 Internet   192.168.17.1          0  001e.081f.13bc  eth-0-17
 Internet   192.168.10.10         0  001e.080c.46ce  vlan10(tunnel)
```

```
Internet   192.168.20.10         0  001e.0811.05f9  vlan20(tunnel)
Internet   192.168.10.1          -  001e.081b.bce0  vlan10
Internet   192.168.20.1          -  001e.081b.bce0  vlan20
Internet   192.168.20.20         0  001e.080c.755e  vlan20(tunnel)
```

18.1.3 Deployment Suggestion

Active-Active Connect to VXLAN Network and Distributed Gateway

1. Topology

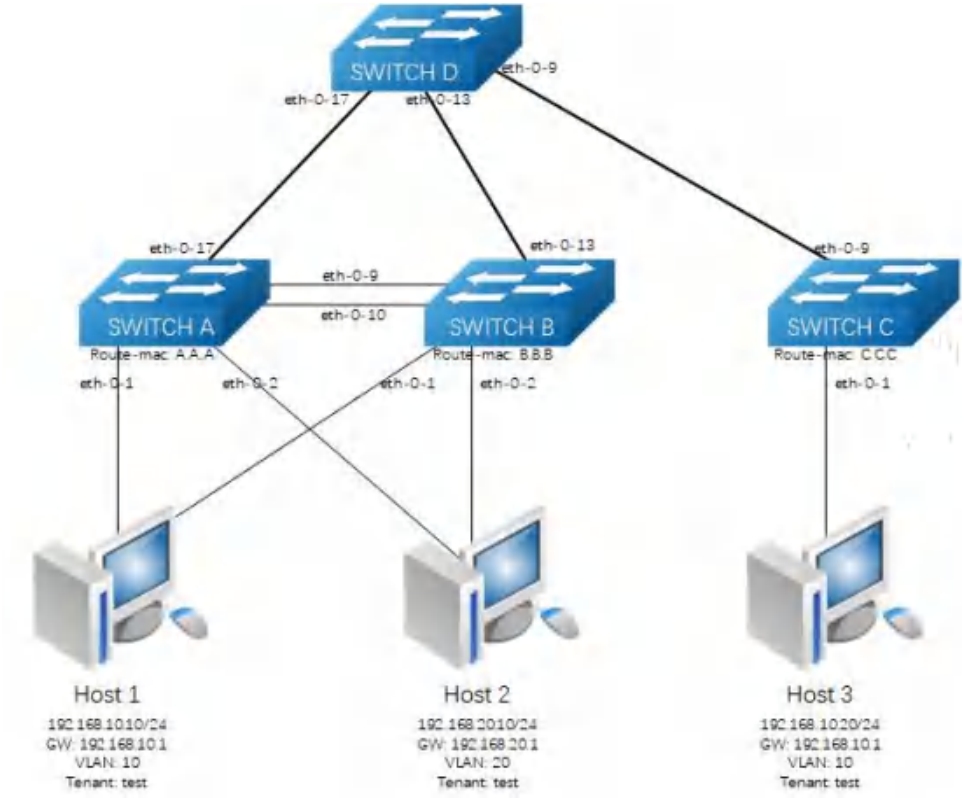


Figure 18-12 Active-active connects to VXLAN and distributed gateway

2. Requirement

From the diagram, Switch A/B/C are TOR switches, between them is Layer 3 network, the downlink HOST are tenement test, it requires to isolate with other tenements. To make sure the reliabilities, part of the servers need active-active connection, so it needs to configure MLAG and configure virtual ip as server's gateway on SWITCH A/B. At the same time, making sure the gateway address is not changing after virtual machine migration, and then deploy distributed gateway on SWITCH A/B/C.

3. Configuration Steps

4. step 1 Configure MLAG on SWITCH A and B separately, is used for server active-active connection, SWITCH C uses single downlink interface

configure peer-link

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A (config)# interface range eth-0-9 – 10
SWITCH_A(config-if-range)# no shutdown
SWITCH_A (config-if-range)# channel-group 55 mode active
```

```
SWITCH_A (config-if-range)# exit
SWITCH_A (config)# interface agg 55
SWITCH_A (config-if)# switchport mode trunk
SWITCH_A (config-if)# switchport trunk allowed vlan all
SWITCH_A (config-if)# spanning-tree port disable
SWITCH_A (config-if)# exit
SWITCH_A (config)# no ip igmp snooping
SWITCH_A (config)# mlag configuration
SWITCH_A (config-mlag)# peer-link agg 55
SWITCH_A (config-mlag)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# interface range eth-0-9 – 10
SWITCH_B(config-if-range)# no shutdown
SWITCH_B(config-if-range)# channel-group 55 mode active
SWITCH_B(config-if-range)# exit
SWITCH_B(config)# interface agg 55
SWITCH_B(config-if)# switchport mode trunk
SWITCH_B(config-if)# switchport trunk allowed vlan all
SWITCH_B(config-if)# spanning-tree port disable
SWITCH_B(config-if)# exit
SWITCH_B(config)# mlag configuration
SWITCH_B(config-mlag)# peer-link agg 55
SWITCH_B(config-mlag)# end
```

configure peer-address

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# vlan database
SWITCH_A(config-vlan)# vlan 4094
SWITCH_A(config-vlan)# exit
SWITCH_A(config)# interface vlan 4094
SWITCH_A(config-if)# ip address 40.94.0.1/24
SWITCH_A(config-if)# exit
SWITCH_A(config)# mlag configuration
SWITCH_A(config-mlag)# peer-address 40.94.0.2
SWITCH_A(config-mlag)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# vlan database
vSWITCH_B(config-vlan)# vlan 4094
SWITCH_B(config-vlan)# exit
SWITCH_B(config)# interface vlan 4094
SWITCH_B(config-if)# ip address 40.94.0.2/24
SWITCH_B(config-if)# exit
SWITCH_B(config)# no ip igmp snooping
SWITCH_B(config)# mlag configuration
SWITCH_B(config-mlag)# peer-address 40.94.0.1
SWITCH_B(config-mlag)# end
```

configure downlink interface, SWITCH A and B use MLAG double downlink, SWITCH C uses normal single downlink

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# vlan database
SWITCH_A(config-vlan)# vlan 10,20,100
SWITCH_A(config-vlan)# exit
SWITCH_A(config)# interface eth-0-1
SWITCH_A(config-if)# switchport mode trunk
SWITCH_A(config-if)# switchport trunk allowed vlan add 10
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# static-channel-group 1
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface eth-0-2
SWITCH_A(config-if)# switchport mode trunk
SWITCH_A(config-if)# switchport trunk allowed vlan add 20
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# static-channel-group 2
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface agg 1
SWITCH_A(config-if)# mlag 1
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface agg 2
SWITCH_A(config-if)# mlag 2
SWITCH_A(config-if)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# vlan database
SWITCH_B(config-vlan)# vlan 10,20,100
SWITCH_B(config-vlan)# exit
SWITCH_B(config)# interface eth-0-1
SWITCH_B(config-if)# switchport mode trunk
SWITCH_B(config-if)# switchport trunk allowed vlan add 10
SWITCH_B(config-if)# static-channel-group 1
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-2
SWITCH_B(config-if)# switchport mode trunk
SWITCH_B(config-if)# switchport trunk allowed vlan add 20
SWITCH_B(config-if)# static-channel-group 2
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface agg 1
SWITCH_B(config-if)# mlag 1
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface agg 2
SWITCH_B(config-if)# mlag 2
SWITCH_B(config-if)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
SWITCH_C(config)# vlan database
SWITCH_C(config-vlan)# vlan 10
SWITCH_C(config-vlan)# exit
SWITCH_C(config)# interface eth-0-1
SWITCH_C(config-if)# switchport mode trunk
SWITCH_C(config-if)# switchport trunk allowed vlan add 10
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# end
```

step 2 Configure the address of network gateway and enable distributed gateway, use virtual ip on MLAG device

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# ip vrf test
SWITCH_A(config-vrf)# exit
SWITCH_A(config)# interface vlan 10
SWITCH_A(config-if)# ip vrf forwarding test
SWITCH_A(config-if)# ip address 192.168.10.253/24
SWITCH_A(config-if)# ip virtual-router address 192.168.10.1
SWITCH_A(config-if)# overlay distributed-gateway enable
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface vlan 20
SWITCH_A(config-if)# ip vrf forwarding test
SWITCH_A(config-if)# ip address 192.168.20.253/24
SWITCH_A(config-if)# ip virtual-router address 192.168.20.1
SWITCH_A(config-if)# overlay distributed-gateway enable
SWITCH_A(config-if)# exit
SWITCH_A(config)# ip virtual-router mac 0.0.1
SWITCH_A(config)# interface vlan 100
SWITCH_A(config-if)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# ip vrf test
SWITCH_B(config-vrf)# exit
SWITCH_B(config)# interface vlan 10
SWITCH_B(config-if)# ip vrf forwarding test
SWITCH_B(config-if)# ip address 192.168.10.254/24
SWITCH_B(config-if)# ip virtual-router address 192.168.10.1
SWITCH_B(config-if)# overlay distributed-gateway enable
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface vlan 20
SWITCH_B(config-if)# ip vrf forwarding test
SWITCH_B(config-if)# ip address 192.168.20.254/24
SWITCH_B(config-if)# ip virtual-router address 192.168.20.1
SWITCH_B(config-if)# overlay distributed-gateway enable
SWITCH_B(config-if)# exit
SWITCH_B(config)# ip virtual-router mac 0.0.1
SWITCH_B(config)# interface vlan 100
SWITCH_B(config)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# ip vrf test
SWITCH_C(config-vrf)# exit
SWITCH_C(config)# interface vlan 10
SWITCH_C(config-if)# ip vrf forwarding test
SWITCH_C(config-if)# ip address 192.168.10.1/24
SWITCH_C(config-if)# overlay distributed-gateway enable
SWITCH_C(config-if)# end
```

step 3 Configure Layer 3 network between switches

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# interface loopback 0
SWITCH_A(config-if)# ip address 10.1.1.1/32
SWITCH_A(config-if)# exit
SWITCH_A(config)# interface eth-0-17
SWITCH_A(config-if)# no switchport
SWITCH_A(config-if)# no shutdown
SWITCH_A(config-if)# ip address 192.168.17.1/24
SWITCH_A(config-if)# exit
SWITCH_A(config)# ip route 10.3.3.3/32 192.168.17.2
SWITCH_A(config)# ip route 10.3.3.3/32 40.94.0.2 100
SWITCH_A(config)#end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# interface loopback 0
SWITCH_B(config-if)# ip address 10.1.1.1/32
SWITCH_B(config-if)# exit
SWITCH_B(config)# interface eth-0-13
SWITCH_B(config-if)# no switchport
SWITCH_B(config-if)# no shutdown
SWITCH_B(config-if)# ip address 192.168.13.1/24
SWITCH_B(config-if)# exit
SWITCH_B(config)# ip route 10.3.3.3/32 192.168.13.2
SWITCH_B(config)# ip route 10.3.3.3/32 40.94.0.1 100
SWITCH_B(config)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# interface loopback 0
SWITCH_C(config-if)# ip address 10.3.3.3/32
SWITCH_C(config-if)# exit
SWITCH_C(config)# interface eth-0-9
SWITCH_C(config-if)# no shutdown
SWITCH_C(config-if)# no switchport
SWITCH_C(config-if)# ip address 192.168.9.1/24
SWITCH_C(config-if)# exit
SWITCH_C(config)# ip route 10.1.1.1/32 192.168.9.2
SWITCH_C(config)# end
```

configure SWITCH D

```
SWITCH_D# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_D(config)# interface eth-0-17
SWITCH_D(config-if)# no shutdown
SWITCH_D(config-if)# no switchport
SWITCH_D(config-if)# ip address 192.168.17.2/24
SWITCH_D(config-if)# exit
SWITCH_D(config)# interface eth-0-13
SWITCH_D(config-if)# no shutdown
SWITCH_D(config-if)# no switchport
SWITCH_D(config-if)# ip address 192.168.13.2/24
SWITCH_D(config-if)# exit
SWITCH_D(config)# interface eth-0-9
SWITCH_D(config-if)# no shutdown
SWITCH_D(config-if)# no switchport
SWITCH_D(config-if)# ip address 192.168.9.2/24
SWITCH_D(config-if)# exit
SWITCH_D(config)# ip route 10.1.1.1/32 192.168.17.1
SWITCH_D(config)# ip route 10.1.1.1/32 192.168.13.1
SWITCH_D(config)# ip route 10.3.3.3/32 192.168.9.1
SWITCH_D(config)# end
```

step 4 Configure VXLAN tunnels

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# vlan database
SWITCH_A(config-vlan)# vlan 10 overlay enable
SWITCH_A(config-vlan)# vlan 100 overlay enable
SWITCH_A(config-vlan)# exit
SWITCH_A(config)# overlay
SWITCH_A(config-overlay)# source 10.1.1.1
SWITCH_A(config-overlay)# remote-vtep 1 ip-address 10.3.3.3 type vxlan
SWITCH_A(config-overlay)# vlan 10 vni 10000
SWITCH_A(config-overlay)# vlan 10 remote-vtep 1
SWITCH_A(config-overlay)# vlan 100 vni 100
SWITCH_A(config-overlay)# vlan 100 remote-vtep 1
SWITCH_A(config-overlay)# exit
SWITCH_A(config)# interface eth-0-17
SWITCH_A(config-if)# vxlan uplink enable
SWITCH_A(config-if)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# vlan database SWITCH_B(config-vlan)# vlan 10 overlay enable
SWITCH_B(config-vlan)# vlan 100 overlay enable
SWITCH_B(config-vlan)# exit
SWITCH_B(config)# overlay
SWITCH_B(config-overlay)# source 10.1.1.1
SWITCH_B(config-overlay)# remote-vtep 1 ip-address 10.3.3.3 type vxlan
SWITCH_B(config-overlay)# vlan 10 vni 10000
SWITCH_B(config-overlay)# vlan 10 remote-vtep 1
```

```
SWITCH_B(config-overlay)# vlan 100 vni 100
SWITCH_B(config-overlay)# vlan 100 remote-vtep 1
SWITCH_B(config-overlay)# exit
SWITCH_B(config)# interface eth-0-13
SWITCH_B(config-if)# vxlan uplink enable
SWITCH_B(config-if)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# vlan database
SWITCH_C(config-vlan)# vlan 10,100
SWITCH_C(config-vlan)# vlan 10 overlay enable
SWITCH_C(config-vlan)# vlan 100 overlay enable
SWITCH_C(config-vlan)# exit
SWITCH_C(config)# overlay
sSWITCH_C(config-overlay)# source 10.3.3.3
SWITCH_C(config-overlay)# remote-vtep 1 ip-address 10.1.1.1 type vxlan
SWITCH_C(config-overlay)# vlan 10 vni 10000
SWITCH_C(config-overlay)# vlan 10 remote-vtep 1
SWITCH_C(config-overlay)# vlan 100 vni 100
SWITCH_C(config-overlay)# vlan 100 remote-vtep 1
SWITCH_C(config-overlay)# exit
SWITCH_C(config)# interface eth-0-9
SWITCH_C(config-if)# vxlan uplink enable
SWITCH_C(config-if)# end
```

step 5 Configure DVR routing, make it interflows with different switches under different segments' hosts

configure SWITCH A

```
SWITCH_A# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_A(config)# ip route vrf test 192.168.10.20/32 remote-vtep 1 vni 100 inner-macda c.c.c
SWITCH_A(config)# end
```

configure SWITCH B

```
SWITCH_B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_B(config)# ip route vrf test 192.168.10.20/32 remote-vtep 1 vni 100 inner-macda c.c.c
SWITCH_B(config)# end
```

configure SWITCH C

```
SWITCH_C# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SWITCH_C(config)# ip route vrf test 192.168.20.10/32 remote-vtep 1 vni 100 inner-macda 0.0.1
SWITCH_C(config)# end
```

step 6 Validation

Check the status of MLAG

```
SWITCH_A# show mlag peer
MLAG neighbor is 40.94.0.2, MLAG version 1
MLAG state = Established, up for 23:41:26
Last read 00:00:42, hold time is 240, keepalive interval is 60 seconds
Received 1652 messages,Sent 1654 messages
```

```
Open    : received 1, sent 2
KAlive  : received 1646, sent 1646
Fdb sync : received 0, sent 0
Failover : received 0, sent 0
Conf    : received 2, sent 2
Syspri  : received 1, sent 1
Peer fdb : received 15, sent 15
STP Total: received 2, sent 3
 Global  : received 2, sent 3
 Packet  : received 0, sent 0
 Instance: received 0, sent 0
 State   : received 0, sent 0

Connections established 1; dropped 0
Local host: 40.94.0.1, Local port: 61000
Foreign host: 40.94.0.2, Foreign port: 42371
remote_sysid: 06a8.c402.0300
```

Check the status of downlink interface on MLAG

```
SWITCH_A# show mlag interface
mlagid  local-if  local-state   remote-state
1     agg1    up          up
2     agg2    up          up
```

Check the status of VXLAN

```
SWITCH_A# show overlay
------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 10.1.1.1

Vlan Vni       Type   Remote-vtep IP-Address
------------------------------------------------------------------
10   10000     VxLAN  1          10.3.3.3
100  100       VxLAN  1          10.3.3.3
```

Check if DVR routing is effective

```
SWITCH_A# show ip route vrf test
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     Dc - DHCP Client
     [*] - [AD/Metric]
     * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.253/32 is in local loopback, vlan10
C       192.168.10.1/32 is directly connected, vlan10
S       192.168.10.20/32 is in overlay remote vxlan vtep:10.3.3.3, vni:100
C       192.168.20.0/24 is directly connected, vlan20
C       192.168.20.253/32 is in local loopback, vlan20
C       192.168.20.1/32 is directly connected, vlan20
```

18.2 Configuring NVGRE

18.2.1 Overview

Brief Introduction

Network Virtualization using Generic Routing Encapsulation (NVGRE) is an encapsulation technique intended to allow virtual network overlays across the physical network. NVGRE uses Generic Routing Encapsulation (GRE) as the encapsulation method. It uses the lower 24 bits of the GRE header to represent the Tenant Network Identifier (TNI.) Like VXLAN this 24 bit space allows for 16 million virtual networks.

18.2.2 Configuration Examples

NVGRE Configuration

1. Topology

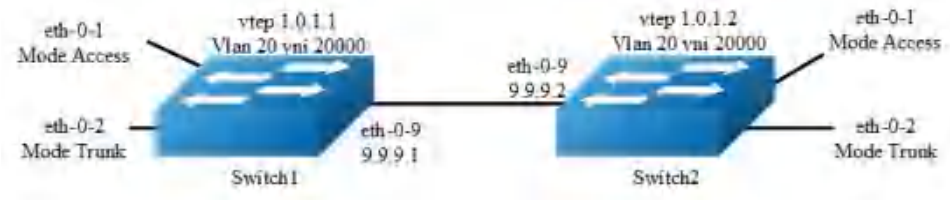


Figure 18-13 NVGRE

2. Configuration Steps

In the following example, switch1 and swith2 are connected via layer 3 route. The traffic of vlan 20 are encapsulated in vni 20000, in order to pass through the layer 3 networks.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# exit
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
```

```
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.1/32
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.2/32
Switch(config-if)# exit
```

Step 4 Create a static route

Configuring Switch1:

```
Switch(config)# ip route 1.0.1.2/32 9.9.9.2
```

Configuring Switch2:

```
Switch(config)# ip route 1.0.1.1/32 9.9.9.1
```

Step 5 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.2 type nvgre
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.1 type nvgre
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Step 6 Exit the configure mode

```
Switch(config)# end
```

Step 7 Validation

Display the result on Switch1:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.0.1.1
------------------------------------------------------------------------------
VLAN ID                  : 20
VNI                      : 20000
EVPN Tunnel Data-fdb Learning : Enable
Remote VTEP NUM          : 1
      Index: 1, Ip address: 1.0.1.2, Source ip: 1.0.1.1, Type: NvGRE, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------------------------
```

Display the result on Switch2:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.0.1.2
------------------------------------------------------------------------------
VLAN ID                  : 20
VNI                      : 20000
EVPN Tunnel Data-fdb Learning : Eanble
Remote VTEP NUM          : 1
      Index: 1, Ip address: 1.0.1.1, Source ip: 1.0.1.2, Type: NvGRE, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------------------------
```

Configuring NVGRE distributed Routing

1. Topology

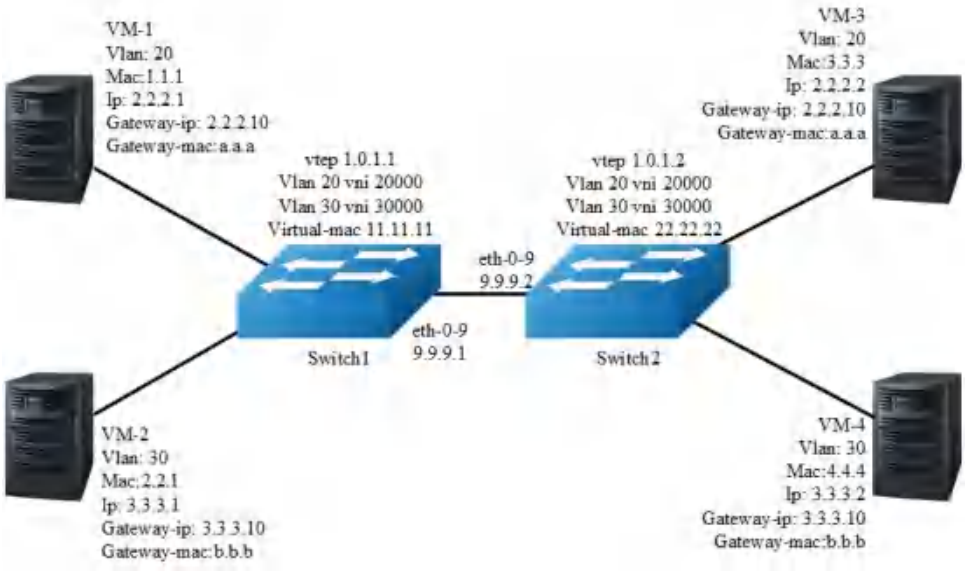


Figure 18-14 NVGRE distributed route

2. Configuration Steps

In the following example, VM-1 & VM-3 are encapsulated in same vni to make up the distributed route via NVGRE; VM-2 & VM-4 are encapsulated in another vni to make up the distributed route via NVGRE.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20,30
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# vlan 30 overlay enable
Switch(config-vlan)# exit
```

Step 3 Create a vrf instance

```
Switch(config)# ip vrf tenant
Switch(config-vrf)# exit
```

Step 4 Create the layer 3 interface and set the ip address

Configuring Switch1:

```
Switch(config)# interface  vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.111/24
Switch(config-if)# exit

Switch(config)# interface  vlan 30
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 3.3.3.111/24
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface  vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.222/24
Switch(config-if)# exit

Switch(config)# interface  vlan 30
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 3.3.3.222/24
Switch(config-if)# exit
```

Step 5 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add  30
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Configuring Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.1/32
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.0.1.2/32
Switch(config-if)# exit
```

Step 6 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.2 type nvgre
Switch(config-overlay)# remote-vtep 1 virtual-mac 22.22.22
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 30 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.0.1.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.0.1.1 type nvgre
Switch(config-overlay)# remote-vtep 1 virtual-mac 11.11.11
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 30 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

Step 7 Create a static routes and NVGRE routes

Configuring Switch1:

```
Switch(config)# ip route 1.0.1.2/32 9.9.9.2
Switch(config)# ip route vrf tenant 2.2.2.2/32 remote-vtep 1 vni 20000 inner-macda 3.3.3
Switch(config)# ip route vrf tenant 3.3.3.2/32 remote-vtep 1 vni 30000 inner-macda 4.4.4
```

Configuring Switch2:

```
Switch(config)# ip route 1.0.1.1/32 9.9.9.1
Switch(config)# ip route vrf tenant 2.2.2.1/32 remote-vtep 1 vni 20000 inner-macda 1.1.1
Switch(config)# ip route vrf tenant 3.3.3.1/32 remote-vtep 1 vni 30000 inner-macda 2.2.2
```

Step 8 Exit the configure mode

```
Switch(config)# end
```

Step 9 Validation

Display the result on Switch1:

```
Switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S     2.2.2.2/32 is in overlay remote nvgre vtep:1.0.1.1->1.0.1.2, vni:20000
S     3.3.3.2/32 is in overlay remote nvgre vtep:1.0.1.1->1.0.1.2, vni:30000
```

Display the result on Switch2:

```
Switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S     2.2.2.1/32 is in overlay remote nvgre vtep:1.0.1.2->1.0.1.1, vni:20000
S     3.3.3.1/32 is in overlay remote nvgre vtep:1.0.1.2->1.0.1.1, vni:30000
```

18.3 Configuring GENEVE

18.3.1 Overview

Function Introduction

Generic Network Virtualization Encapsulation (GENEVE) is a networking technology that encapsulates MAC-based Layer 2 Ethernet frames within Layer 3 UDP packets to aggregate and tunnel multiple layer 2 networks across a Layer 3 infrastructure. GENEVE scales up to 16 million logical networks and supports layer 2 adjacency across IP networks. Multicast transmission architecture is used for broadcast/multicast/unknown packets.

Principle Description

N/A

18.3.2 Configuration

GENEVE Configuration

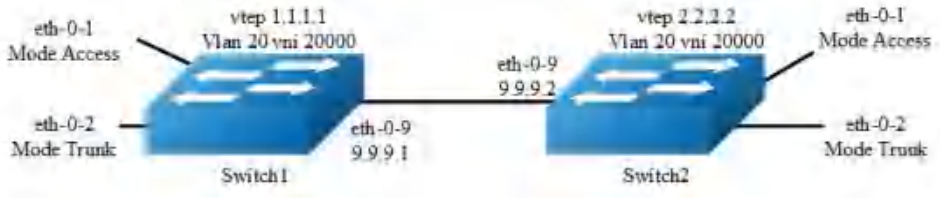


Figure 18-15 GENEVE

In the following example, switch1 and swith2 are connected via layer 3 route. The traffic of vlan 20 are encapsulated in vni 20000, in order to pass through the layer 3 networks.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
```

```
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 2.2.2.2/32
Switch(config-if)# exit
```

Step 4 Create a static route

Configuring Switch1:

```
Switch(config)# ip route 2.2.2.0/24 9.9.9.2
```

Configuring Switch2:

```
Switch(config)# ip route 1.1.1.0/24 9.9.9.1
```

step 5 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.1.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 2.2.2.2 type geneve
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 2.2.2.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.1.1.1 type geneve
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.1.1.1
------------------------------------------------------------------------------
VLAN ID                 : 20
VNI                     : 20000
EVPN Tunnel Data-fdb Learning : Eanble
Remote VTEP NUM: 1
      Index: 1, Ip address: 2.2.2.2, Source ip: 1.1.1.1, Type: GENEVE, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------------------
```

Display the result on Switch2:

```
Switch# show overlay vlan 20
------------------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 2.2.2.2
------------------------------------------------------------------------------
VLAN ID                 : 20
VNI                     : 20000
EVPN Tunnel Data-fdb Learning : Enable
Remote VTEP NUM: 1
      Index: 1, Ip address: 1.1.1.1, Source ip: 2.2.2.2, Type: GENEVE, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------------------------
```

Configuring GENEVE Routing

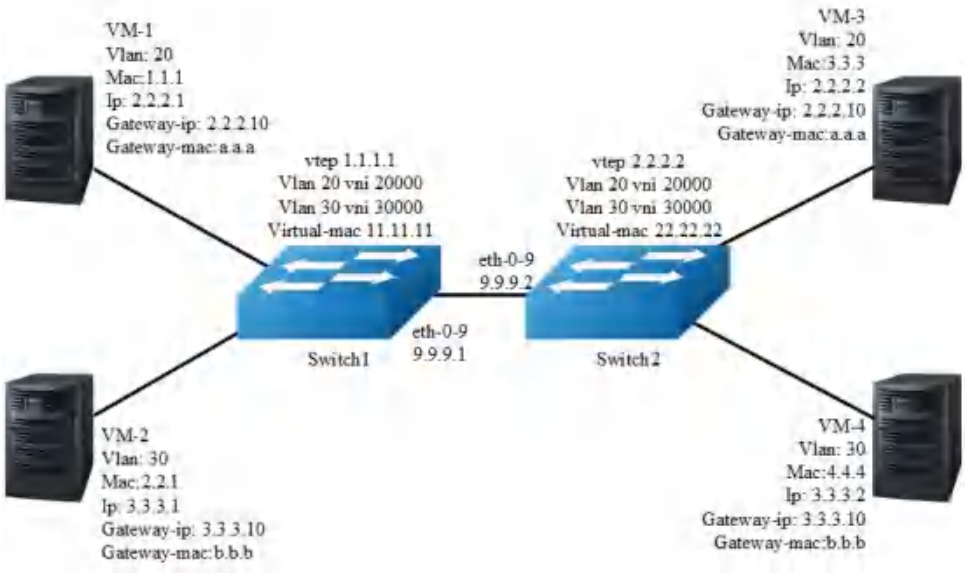


Figure 18-16 GENEVE

In the following example, VM-1 & VM-3 are encapsulated in same vni to make up the distributed route via GENEVE; VM-2 & VM-4 are encapsulated in another vni to make up the distributed route via GENEVE.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20,30
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# vlan 30 overlay enable
Switch(config-vlan)# exit
```

step 3 Create a vrf instance

```
Switch(config)# ip vrf tenant
Switch(config-vrf)# exit
```

step 4 Create the layer 3 interface and set the ip address

Configuring Switch1:

```
Switch(config)# interface  vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.111/24
Switch(config-if)# exit

Switch(config)# interface  vlan 30
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 3.3.3.111/24
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface  vlan 20
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 2.2.2.222/24
Switch(config-if)# exit

Switch(config)# interface  vlan 30
Switch(config-if)# ip vrf forwarding tenant
Switch(config-if)# ip address 3.3.3.222/24
Switch(config-if)# exit
```

step 5 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add  30
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Configuring Switch1:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# exit
```

Configuring Switch2:

```
Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface loopback0
Switch(config-if)# ip address 2.2.2.2/32
Switch(config-if)# exit
```

step 6 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.1.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 2.2.2.2 type geneve
Switch(config-overlay)# remote-vtep 1 virtual-mac 22.22.22
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 30 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 2.2.2.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.1.1.1 type geneve
Switch(config-overlay)# remote-vtep 1 virtual-mac 11.11.11
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 30 vni 30000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 20 gateway-mac a.a.a
Switch(config-overlay)# vlan 30 gateway-mac b.b.b
Switch(config-overlay)# exit
```

step 7 Create a static routes and GENEVE routes

Configuring Switch1:

```
Switch(config)# ip route 2.2.2.0/24 9.9.9.2
Switch(config)# ip route vrf tenant 2.2.2.2/32 remote-vtep 1 vni 20000 inner-macda 3.3.3
Switch(config)# ip route vrf tenant 3.3.3.2/32 remote-vtep 1 vni 30000 inner-macda 4.4.4
```

Configuring Switch2:

```
Switch(config)# ip route 1.1.1.0/24 9.9.9.1
Switch(config)# ip route vrf tenant 2.2.2.1/32 remote-vtep 1 vni 20000 inner-macda 1.1.1
Switch(config)# ip route vrf tenant 3.3.3.1/32 remote-vtep 1 vni 30000
```

step 8 Exit the configure mode

```
Switch(config)# end
```

step 9 Validation

Display the result on Switch1:

```
switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S   2.2.2.2/32 is in overlay remote geneve vtep:1.1.1.1->2.2.2.2, vni:20000
S   3.3.3.2/32 is in overlay remote geneve vtep:1.1.1.1->2.2.2.2, vni:30000
```

Display the result on Switch2:

```
switch# show ip route vrf tenant
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default
S   2.2.2.1/32 is in overlay remote geneve vtep:2.2.2.2->1.1.1.1, vni:20000
S   3.3.3.1/32 is in overlay remote geneve vtep:2.2.2.2->1.1.1.1, vni:30000
```

18.4 Configuring Overlay

18.4.1 Overview

Function Introduction

Overlay function supports multiple source ip address of vtep, it can set different source ip for different networks and improve the reliability of overlay.

Overlay function also supports tunnel without horizon split, it means that when uplink port receiving tunnel packets and decapsulate them , and then send them into another tunnel for encapsulation.

Principle Description

N/A

18.4.2 Configuration

Configuring Overlay multiple source ip

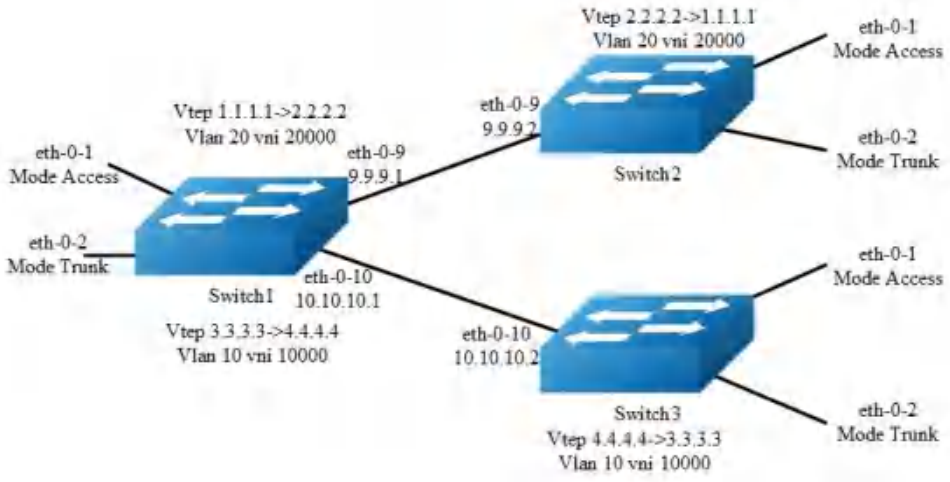


Figure 18-17 Overlay multiple source ip

The following example uses vxlan for overlay configuration. NVGRE and GENEVE configurations are similar with vxlan.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

Configuring Switch1:

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20,10
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# vlan 10 overlay enable
Switch(config-vlan)# exit
```

Configuring Switch2:

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# exit
```

Configuring Switch3:

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 10
Switch(config-vlan)# vlan 10 overlay enable
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
```

```
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config)# interface loopback1
Switch(config-if)# ip address 3.3.3.3/32
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 2.2.2.2/32
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 10
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-10
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.10.10.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 4.4.4.4/32
Switch(config-if)# exit
```

step 4 Create static routes

Configuring Switch1:

```
Switch(config)# ip route 2.2.2.0/24 9.9.9.2
Switch(config)# ip route 4.4.4.0/24 10.10.10.2
```

Configuring Switch2:

```
Switch(config)# ip route 1.1.1.0/24 9.9.9.1
```

Configuring Switch3:

```
Switch(config)# ip route 3.3.3.0/24 10.10.10.1
```

step 5 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.1.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 2.2.2.2 type vxlan
Switch(config-overlay)# remote-vtep 2 ip-address 4.4.4.4 type vxlan src-ip 3.3.3.3
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 10 vni 10000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 10 remote-vtep 2
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 2.2.2.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.1.1.1 type vxlan
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Configuring Switch3:

```
Switch(config)# overlay
Switch(config-overlay)# source 4.4.4.4
Switch(config-overlay)# remote-vtep 1 ip-address 3.3.3.3 type vxlan
Switch(config-overlay)# vlan 10 vni 10000
Switch(config-overlay)# vlan 10 remote-vtep 1
Switch(config-overlay)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch1:

```
switch# show overlay vlan 20
------------------------------------------------------------------
ECMP Mode       : Normal
Source VTEP     : 1.1.1.1
------------------------------------------------------------------
VLAN ID                  : 2
VNI                      : 20000
EVPN Tunnel Data-fdb Learning : Eanble
Remote VTEP NUM: 1
```

```
    Index: 1, Ip address: 2.2.2.2, Source ip: 1.1.1.1, Type: VxLAN, Protocol: Static
    Index: 2, Ip address: 2.2.2.2, Source ip: 3.3.3.3, Type: VxLAN, Protocol: Static
DVR Gateway NUM: 0
------------------------------------------------------------
```

Configuring OVERLAY without Horizon Split

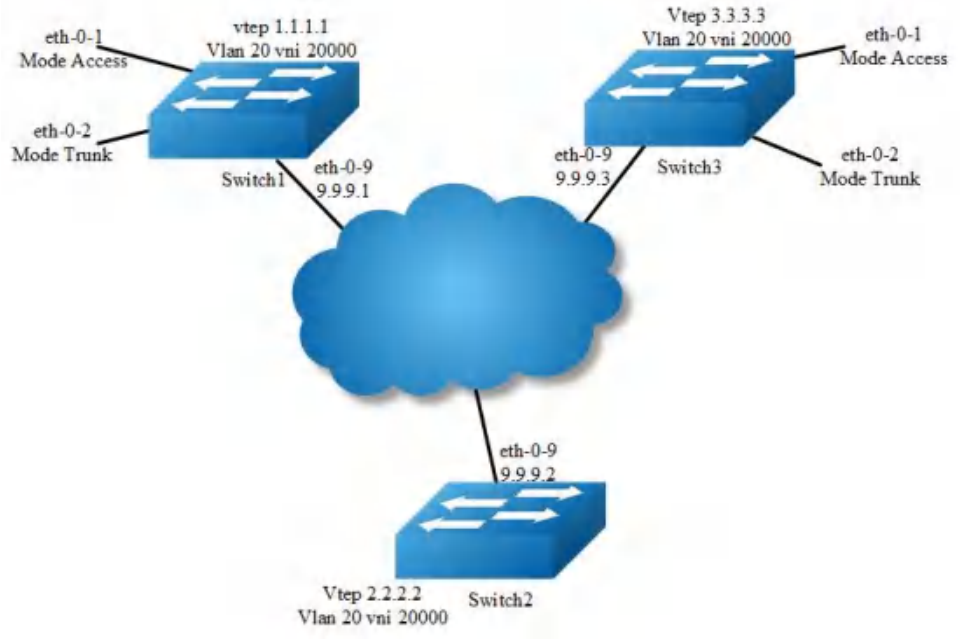


Figure 18-18 OVERLAY without Horizon Split

In the following example, there is a tunnel between switch1 and switch2, there is another tunnel between switch2 and switch3. The horizon split is disable on switch2, therefor packets from one tunnel can be forwarded to another tunnel.

The following example uses vxlan for overlay configuration. NVGRE and GENEVE configurations are similar with vxlan.

The following configuration should be operated on all switches if the switch ID is not specified.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enter the vlan configure mode and create vlan, enable overlay for each vlan

```
Switch(config)# vlan database
Switch(config-vlan)# vlan 20
Switch(config-vlan)# vlan 20 overlay enable
Switch(config-vlan)# exit
```

step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
```

```
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.1/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 1.1.1.1/32
Switch(config-if)# exit
```

Interface configuration for Switch2:

```
Switch(config-if)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.2/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 2.2.2.2/32
Switch(config-if)# exit
```

Interface configuration for Switch3:

```
Switch(config)# interface eth-0-1
Switch(config-if)# switchport access vlan 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan add 20
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface eth-0-9
Switch(config-if)# no switchport
Switch(config-if)# ip address 9.9.9.3/24
Switch(config-if)# overlay uplink enable
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface loopback0
Switch(config-if)# ip address 3.3.3.3/32
Switch(config-if)# exit
```

step 4 Create a static route

Configuring Switch1:

```
Switch(config)# ip route 2.2.2.0/24 9.9.9.2
```

Configuring Switch2:

```
Switch(config)# ip route 1.1.1.0/24 9.9.9.1
Switch(config)# ip route 3.3.3.3/24 9.9.9.3
```

Configuring Switch3:

```
Switch(config)# ip route 2.2.2.0/24 9.9.9.2
```

step 5 Set attributes for overlay

Configuring Switch1:

```
Switch(config)# overlay
Switch(config-overlay)# source 1.1.1.1
Switch(config-overlay)# remote-vtep 1 ip-address 2.2.2.2 type vxlan
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

Configuring Switch2:

```
Switch(config)# overlay
Switch(config-overlay)# source 2.2.2.2
Switch(config-overlay)# remote-vtep 1 ip-address 1.1.1.1 type vxlan horizon-split-disable
Switch(config-overlay)# remote-vtep 2 ip-address 3.3.3.3 type vxlan horizon-split-disable
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# vlan 20 remote-vtep 2
Switch(config-overlay)# exit
```

Configuring Switch3:

```
Switch(config)# overlay
Switch(config-overlay)# source 3.3.3.3
Switch(config-overlay)# remote-vtep 1 ip-address 2.2.2.2 type vxlan
Switch(config-overlay)# vlan 20 vni 20000
Switch(config-overlay)# vlan 20 remote-vtep 1
Switch(config-overlay)# exit
```

step 6 Exit the configure mode

```
Switch(config)# end
```

step 7 Validation

Display the result on Switch2:

```
switch# show overlay remote-vtep
Index Type  Virtual-Mac   IP-Address  Source-Ip  Split-Horizon Keep-vtag  Dscp-strategy
-------------------------------------------------------------------------------------
1   VxLAN     -     1.1.1.1   2.2.2.2    Disable     Disable   Dscp-copy

2   VxLAN     -     3.3.3.3   2.2.2.2    Disable     Disable   Dscp-copy
```

18.5 Configuring EVPN

18.5.1 Overview

Brief Introduction

Abbreviation

Abbreviation	Full Name
EVPN	Ethernet Virtual Private Network
VXLAN	Virtual eXtensible Local Area Network
IRB	Integrated Routing and Bridging
RR	Route Reflector
VTEP	VXLAN Tunnel End Point
VM	Virtual Machine
BUM	Broadcast, Unknown unicast, or Multicast
L2 VNI	Layer 2 Virtual Network Identifier
L3 VNI	Layer 3 Virtual Network Identifier
PMSI	Provider Multicast Service Interface
NLRI	Network Layer Reachability Information
BGP	Border Gateway Protocol
ARP	Address Resolution Protocol
MLAG	Multi-Chassis Link Aggregation Group

Function description

EVPN (Ethernet Virtual Private Network) is a kind of VPN technology that is used for network interconnection. EVPN technique uses the mechanism which is similar as BGP/MPLS IP VPN; it defines a kind of new NLRI (Network Layer Reachability Information) that is EVPN NLRI which is based on BGP protocol. EVPN NLRI defines some kinds of new routing types of BGP EVPN; they are used in MAC address learning and advertisingon different sites of Layer 2 network. It provides Layer 3 interconnection between different subnets of the same tenants, and provides Layer 3 interconnection to external network.

Original VXLAN realization scheme does not have control plane, it conducts finding VTEP and host information learning (includes IP address, MAC address, VNI and VTEP IP address of gateway) by traffic flooding on data plane; this method leads to traffic flooding in data center network. To solving this problem, VXLAN imports EVPN as control plane, realizing auto finding, host information advertising and other features by switching BGP EVPN routing between VTEPs, and it avoids unnecessary data traffic flooding.

To sum up, EVPN defines some new BGP EVPN routing by extending BGP protocol; these BGP EVPN routing can transmit VTEP address and host information, so it can make VTEP finding and host information learning transfer from data plane to control plane when apply EVPN into VXLAN network.

IN EVPN network, some common basic concepts are shown as following:

- Symmetrical routing: when different subnet devices in different VTEPs from the same tenant conducting Layer 3 networking, searching routing and forwarding its encapsulation on local VTEP; after it reaches to remote VTEP terminal, it decapsulates and then searching routing forwarding again. It is called symmetrical routing, it usually use the same VNI between symmetrical routing modes.
- Unsymmetrical routing: when different subnet devices in different VTEPs from the same tenant conducting Layer 3 networking, searching routing and forwarding its encapsulation on local VTEP; after it reaches to remote VTEP terminal, it decapsulates and then searching Layer 2 forwarding table items and forwards. It is called unsymmetrical routing, it usually use different VNIs between unsymmetrical routing modes.
- L2VNI: Layer 2 VNI, it maps 1:1 with VLAN, it has the same L2VNI for devices in the same Layer 2, networks between each other will pass through this VNI. In unsymmetrical routing mode, it belongs to the same tenant but conducts by different subnet devices, and then it will use L2VNI which belongs to destination devices to encapsulate.
- L3VNI: Layer 3 VNI, it maps 1:1 with VRF. In symmetrical routing mode, it conducts networking which belongs to the same tenant but different subnet devices unify using this VNI.
- EVPN instance: one Layer 2 VNI corresponds with one EVPN instance, each EVPN instance has independent RD, Route-Target attribute.

Background

VXLAN is a Network Virtualization over Layer 3 technology used in data center virtualization, providing multi-tenant access extension and VM migration scenarios. The existing VXLAN does not have a control plane and uses the multicast flooding mode to discover VTEPs and learn remote host addresses on the data plane. As a result, there is a lot of flooding in the data center network.

In order to overcome the flooding mode of learning, the draft ietf standard (draft-ietf-bess-evpn-overlay-02.txt) defines MP-BGP EVPN as the VXLAN control plane. As a control plane, EVPN can automatically discover VXLAN tunnels and notify the control plane of host information (host IP address, host MAC address, and host ARP). Because the host information is notified through the EVPN control surface, traffic flooding is avoided.

Principle Description

Packet format

NOTE The description of VXLAN packet reference to [VXLAN] chapter.

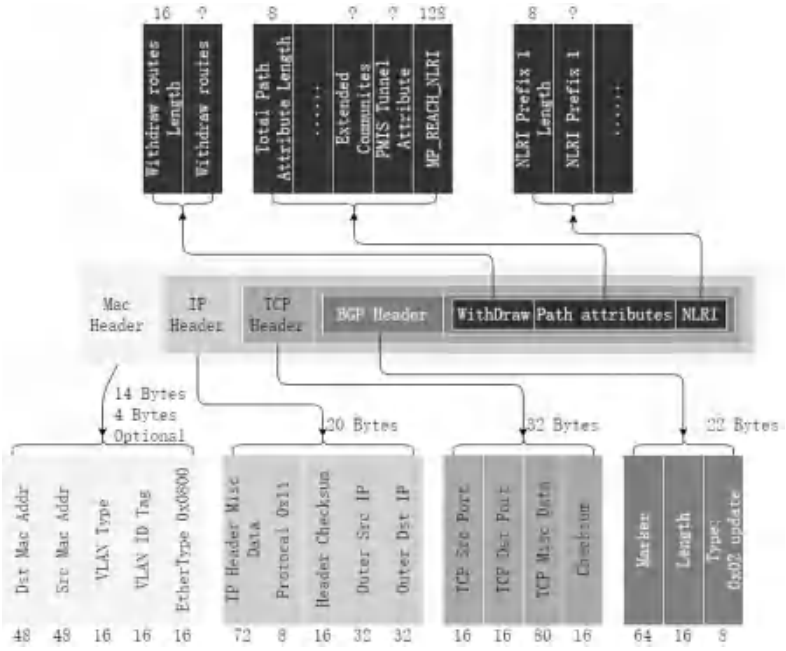


Figure 18-19 BGP Update message format

EVPN protocol packet is encapsulate in the Path attribute of BGP packet, carried with NLRI format. The route types and packet format defined in EVPN NLRI will be described in following chapters.

Withdraw Routes

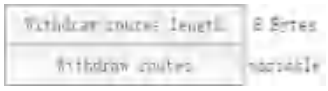


Figure 18-20 Withdraw Routes message format

Withdraw Route includes the routes to withdraw, in the withdraw list each unit includes a fixed length field and a variable prefix field.

Table 18-1 Each fields of Withdraw message

section	Explanation
Withdraw routes length	The length of route to withdraw. 0 indicates no route to withdraw
Withdraw routes	The prefix of route to withdraw

BGP EXTEMDED_COMMUNITIES

BGP EXTEMDED_COMMUNITIES is a type in Path Attribute of BGP packets, wich is used to carry the BGP extended communication attribute. EVPN route target message can be carried by this attribute.

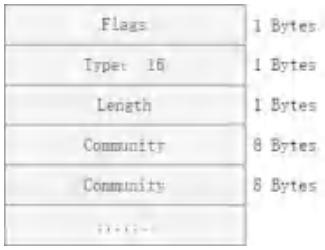


Figure 18-21 EXTEMDED_COMMUNITIES message format

Carry AS/AN (route target) with the communication attribute Community Transitive Two-Octet AS Route target.

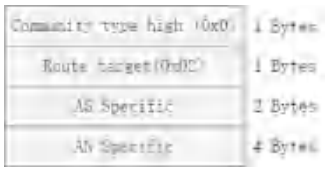


Figure 18-22 Community Transitive Two-Octet AS Route target message format

Community Transitive Opaque Encapsulation attribute specified the encapsulation type.

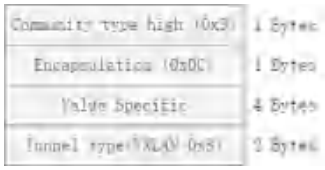


Figure 18-23 Community Transitive Opaque Encapsulation message format

For notifying VTEP IP, VNI and downlink host address and other information between VTEPs. EVPN NLRI defines the following BGP EVPN routing types that apply to VXLAN control plane:

- Type 2 routing (MAC/IP routing)
- Type 3 routing (Inclusive Multicast routing)
- Type 5 routing (IP prefix routing)

The 3 types of routing are described as following:

Type 2 routing (MAC/IP routing)

Type 2 routing is used for notifying MAC/IP of hosts on local VTEP to remote terminal VTEP. The following is the message format of this type of routing.



Figure 18-24 Type 2 routing's message format

Table 18-2 Each fields of Type2 message

section	Explanation
Route Distinguisher	this section is the configured value of RD (Route Distinguisher) of EVPN instance
Ethernet Segment Identifier	this section is the unique identification that connects the definition between current device and opposite terminal (it is all "0" currently)
Ethernet Tag ID	this section id the VLAN ID that is practical configured on current device (it is all "0" currently)
MAC Address Length	this section is the length of host MAC address that this route takes.
MAC Address	this section is the MAC address of host which this route takes
IP Address Length	this section is the mask length of host IP address that this route carries.
IP Address	this is the host IP address that this route carries
MPLS Label1	this section is Layer 2 VNI that this route carries
MPLS Label2	this section is Layer 3 VNI that this route carries.

This type's main functions include:

- Host MAC notification: To realize Layer 2 interconnect between subnet's hosts, VTEPs from both terminals need to learn host MAC with each other. As exchanging Type 2 routing between VTEPs of BGP EVPN neighbors, they can notify with each other about host MACs that are gained by VTEPS. MAC Address Length and MAC Address section are host's MAC address. When only need to notify MAC, Type 2 routing does not need to carry Layer 3 VNI and IP Address; meanwhile, IP Address Length should be 0.
- Host IP notification: In distributed network gateway, to realize interconnection between across subnet hosts, two terminals of VTEP (as Layer 3 gateway) need to learn host's IP routings with each other. By notifying with each other about gained host's IP routing, it can exchanging Type 2 routing between VTEPs which are neighbors of BGP EVPN. Here, IP Address Length and IP Address section are destination address of host IP routing; at the same time, MPLS Label2 section must carry Layer 3 VNI. The MAC/IP routing is also called IRB (Integrated Routing and Bridge) type routing.

Type 3 routing (Inclusive Multicast routing)

Type 3 routing contains prefix and PMSI attribute; VTEP IP address will store in prefix's Originating Router's IP Address section, VNI will store in MPLS Label section of PMSI attribute. This type routing is used for VTEP's auto finding and dynamic establishing of VXLAN tunnel. VTEPs as the neighbors of BGP EVPN, transmitting Layer 2 VNI and VTEP IP address information with each other by Inclusive Multicast route. When received message carries the VNI which has the same configuration as local, then it will create a head-end replication table, which will be used for forwarding BUM messages later. The following is the message format of this type of routing.

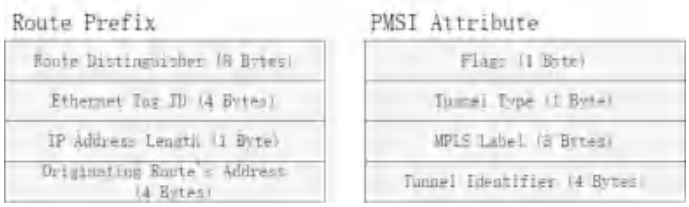


Figure 18-25 Type 3 routing's prefix and PMSI attributes

Table 18-3 Each fields of Type3 Route prefix and PMSI attribute

Section	Explanation
Route Distinguisher	this section is the configured value of RD (Route Distinguisher) of EVPN instance
Ethernet Tag ID	this section id the VLAN ID that is practical configured on current device (it is all "0" currently)
IP Address Length	this section is the mask length of host IP address that this route carries.

Originating Router's IP Address	this section is the local VTEP IP address that is carried by this route.
Flags	this section is flag spot, it notes whether the current tunnel needs information about leaf node or not In VXLAN, this section does not have practical meaning.
Tunnel Type	this section is the type of tunnel that is carried by this routing. Currently in VXLAN scenario, types that it supports is only "6：Ingress Replication", that is head-end replication, which is used for forwarding BUM message.
MPLS Label	this section id Layer 2 VNI that carried by this route.
Tunnel Identifier	this is the tunnel information that this route carries. Currently in VXLAN, this section is the IP address of local VTEP.

Type 5 routing (IP prefix routing)

This type of routing is notifying network sections address to realize the host from VXLAN network to access external network. The message format for this type of routing is shown as the following figure.



Figure 18-26 Type 5 routing's message format

Table 18-4 Each fields of Type5 message

section	Explanation
Route Distinguisher	this section is the configured value of RD (Route Distinguisher) of EVPN instance
Ethernet Segment Identifier	this section is the unique identification that connects the definition between current device and opposite terminal (it is all "0" currently)
Ethernet Tag ID	this section id the VLAN ID that is practical configured on current device (it is all "0" currently)
IP Prefix Length	this section is IP prefix mask length that this routing carries, normally it is not 32.
IP Prefix	this section is the IP prefix that this routing carries
GW IP Address	this section is the gateway address by default. It has no practical meaning in VXLAN setting.
MPLS Label	this section is Layer 3 VNI that this routing carries.

Establish VXLAN Tunnel in Layer 2

If EVPN is not used, the VXLAN tunnel is created by manually configuring the VNI to the peer list corresponding to the VNI. This solution requires a large amount of configuration and is very inconvenient.

After EVPN is introduced as the control plane, the inclusive route (Type3) in EVPN can be used to transmit L2 VNI and VTEP address information. In this way, L2 VXLAN tunnels can be dynamically created, reducing a large number of configurations and improving deployment efficiency.

EVPN uses Type3 routes, that is, Inclusive multicast routes, to deliver local L2 VNI+ L2VTEP information for automatic VXLAN tunnel discovery.

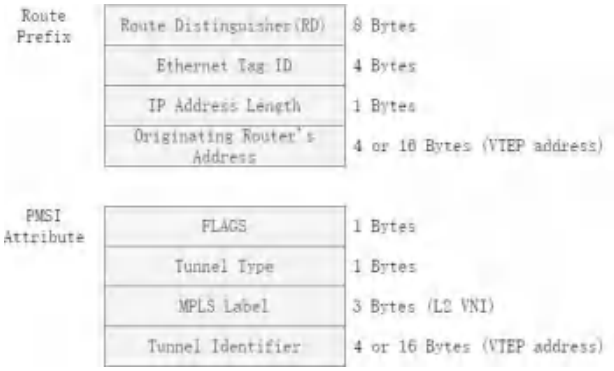


Figure 18-27 Application of Type 3 routes to automatic discovery of VXLAN tunnels

VXLAN tunnel is confirmed by a pair of VTEP IP address, create VXLAN tunnel is the process that gaining VTEP IP address of both terminated sides' VTEPs; if remote VTEP IP can be reached by Layer 3 route, then VXLAN tunnel can be created successfully. Creating VXLAN tunnels by BGP EVPN method dynamically, that is creating BGP EVPN neighbors between two VTEP, and then using BGP EVPN Type3 routing to transmit VNI and VTEP IP address between neighbors, thus realizing dynamic creating VXLAN tunnels.

In The figure below, Vtep-1 and Vtep-2 deploys a Host separately; it only need to create one VXLAN tunnel that use for Layer 2 forwarding when they are in the same network segment.

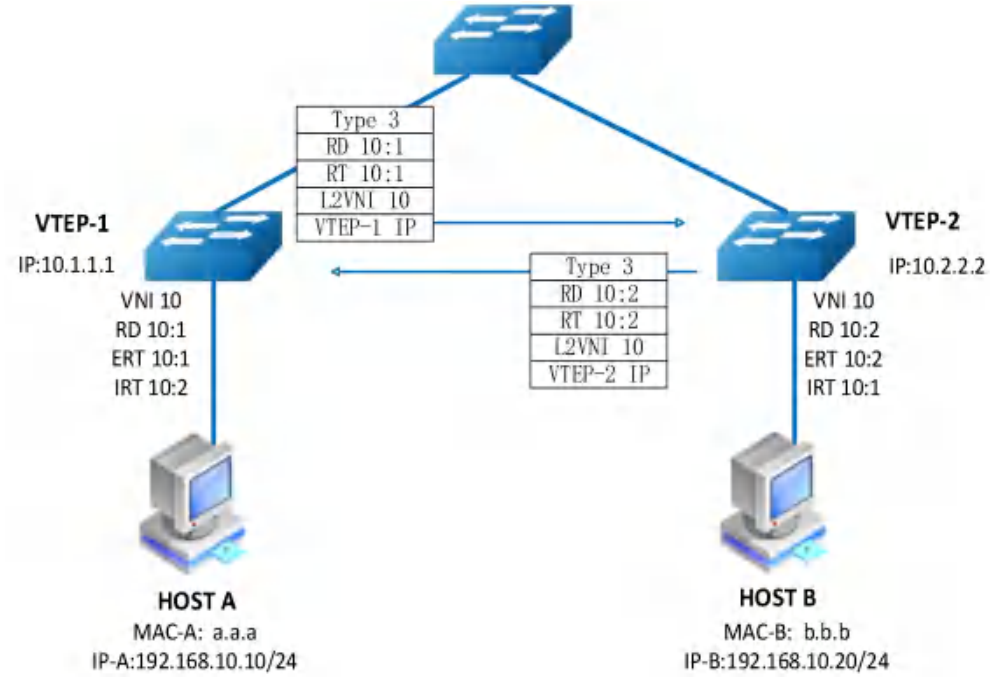


Figure 18-28 VXLAN Tunnel

Firstly, create BGP EVPN neighbor between Vtep-1 and Vtep-2. Then, create EVPN instance, configure RD of local EVPN instance, exportable Route-Target (ERT), importable Route-Target (IRT). After configuring local VTEP IP address, Vtep-1 and Vtep-2 will generate BGP EVPN Type 3 route and send to opposite end, this route carries local EVPN instance's exportable Route-Target, VTEP-1 and VTEP-2 will examine the Route-Target that this route carries after receiving BGP EVPN Type 3 routing from remote VTEP; if it is equal to importable Route-Target of local EVPN instance, then accepts this route, otherwise discard this route. After accepting this route, VTEP-1 and VTEP-2 will gain remote VTEP ip address and VNI; if VNI of remote VTEP is the same as local VNI, then it will create a VXLAN tunnel to remote VTEP ip, and it can do BUM message forwarding later.

MAC Table Synchroization

When hosts on the same network segment communicate with each other, they can use Layer 2 informed unicast forwarding after learning each other's MAC addresses. When the EVPN control plane is not introduced, hosts on the VXLAN network learn MAC addresses based on data traffic flooding. The host MAC address is passed through the EVPN Type2 route to reduce the traffic flooding of the VXLAN network.

Route Distinguisher(RD)	8 Bytes
Ethernet Segment ID	10 Bytes
Ethernet Tag ID	4 Bytes
MAC Address Length	1 Bytes
MAC Address	6 Bytes
IP Address Length	1 Bytes
IP Address	0 Byte
MPLS Label 1	3 Bytes (L2 VNI)
MPLS Label 2	0 Byte

Figure 18-29 Application of Type 2 routes to host MAC address notification

Like the figure from above section, HOST A and HOST B are in the same network segment, since it does not have detailed MAC forwarding table after creating the tunnel, traffic between them will forward by broadcasting, which causes unnecessary waste of bandwidths. It will reduce the waste of bandwidths when MAC finishing synchronization.

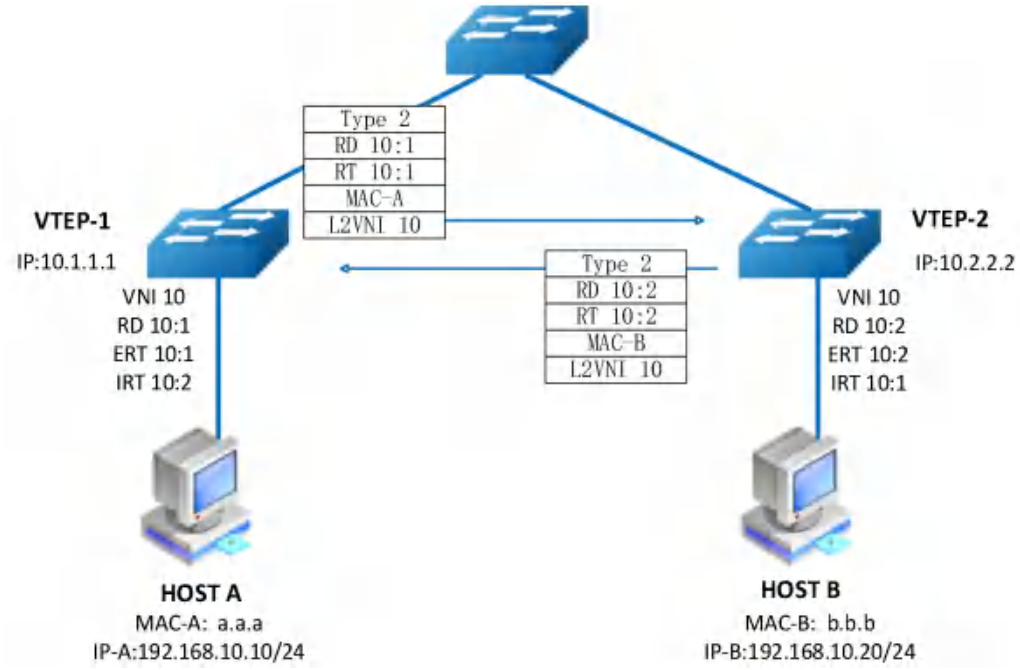


Figure 18-30 MAC table synchronization

When HostA networks with VTEP-1 at the first time, VTEP-1 learns the relationship of HostA's MAC address, its VLAN and inbound interface (eth-0-1). Meanwhile, VTEP-1 will generate BGP EVPN route according to MAC entry of HostA and send to VTEP-2, this route carries local EVPN instance's exportable Route-Target, route next hop's attribute and BGP EVPN protocol new defined Type2 routing that is MAC/IP routing. The next hop's attribute that the route carries is local VTEP IP address; HOSTA's MAC address stores in MAC Address Length and MAC Address section, Layer 2 VNI (map the relevant vlan of MAC which is learnt to corresponding VNI) stores on MPLS Label1. Since it is Layer 2 networking which only needs to synchronize MAC, so Type 2 routing doesn't need to carry host IP and Layer 3 VNI information.

After VTEP-2 receives the BGP EVPN route from VTEP-1, it will examine the Route-Target that this route carries first, if it is equal to the importable Route-Target of local EVPN instance, and then continuously compare with L2VNI whether it is equal to the configured L2VNI of this EVPN instance, if it equals then accepts this route, otherwise it discards this route. After accepting this route, VTEP-2 gains the

relationship of MAC address of HostA, L2VNI and VTEP IP address (next hop's attribute) on VTEP-1; and it will generate HOSTA's MAC entry on local MAC table, its outbound interface is pointing to VXLAN tunnel of VTEP-1, the relevant VNI is corresponding Layer 2 VNI.

Host Route Synchroization

Type2 routes can carry host MAC and host IP at the same time, so they can be used to transmit host ARP. After learning the: ARP of the host, you can use the above route to send host ARP to the neighbor. After receiving ARP, the remote neighbor saves the ARP entry to the local ARP cache list.

Route Distinguisher(RD)	8 Bytes
Ethernet Segment ID	10 Bytes
Ethernet Tag ID	4 Bytes
MAC Address Length	1 Bytes
MAC Address	6 Bytes (Host MAC Address)
IP Address Length	1 Bytes
IP Address	4 or 16 Bytes (Host IP address)
MPLS Label 1	3 Bytes (L2 VNI)
MPLS Label 2	0 or 3 Bytes

Figure 18-31 Application of Type 2 routes to host ARP notification

When hosts of VTEPs are in different segments but need to communicate with each other, it needs to use Layer 3 forwarding to come true. Then conducts MAC synchronization cannot provide communication; it needs to notify the host's IP address and next hop's corresponding information by Type 2 message to remote VTEP to make sure that they can interflow by routing between each other.

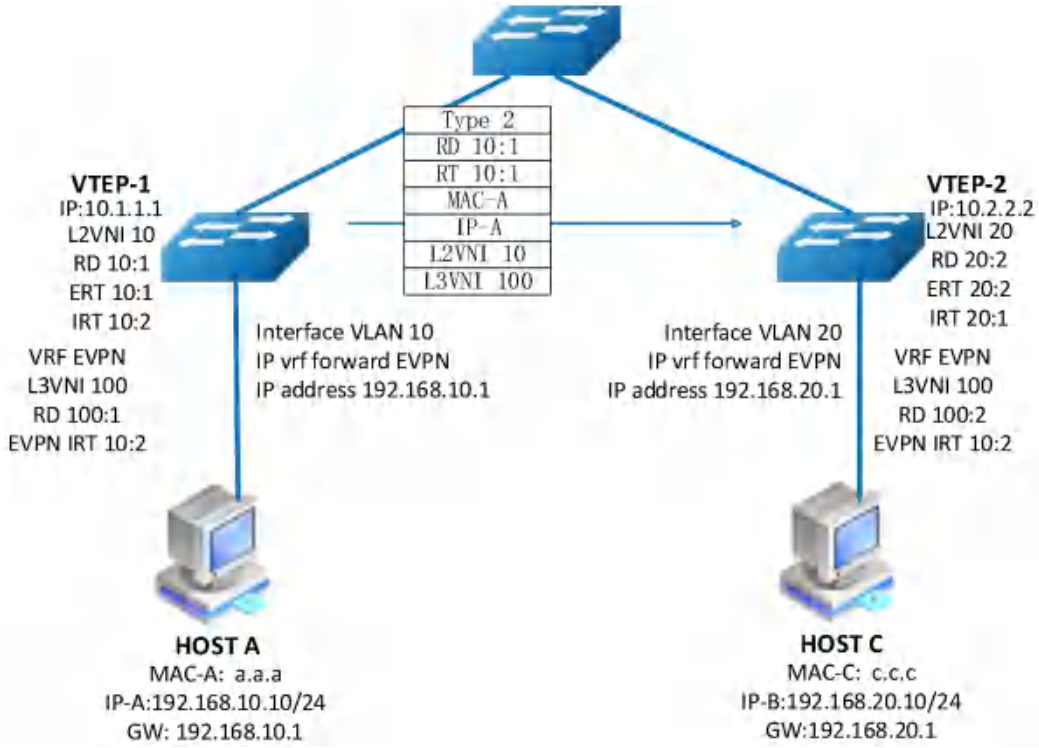


Figure 18-32 Host Route synchronization

In the figure above, HostA and HostC are in different segments, they need to notify route to each other to complete the communication; by dynamic ARP messages, VTEP-1 can learn ARP table of HostA, gain MAC and IP address of HostA. VTEP-1 gets the corresponding L2VNI by the vlan where HostA's MAC is, and learn the Layer 3 interface (interface vlan 10) on this ARP binds with vrf instance (vrf EVPN), vrf instance relevance Layer 3 VNI. VTEP-1 can get the Layer 3 VNI which belongs to HostA; when it has information of MAC/IP/Layer 2 VNI/Layer 3 VNI, EVPN instance on VTEP-1 can generate Type2 (IRB) route by above information. Host's MAC information and Layer 2 VNI

information is shown in section 2.3; host's IP address stores in IP Address Length and IP Address, Layer 3 VNI stores in MPLS Label2, RD and Route-Target are the RD and exportable Route-Target of EVPN instance.

After VTEP-2 receives the Type 2 routing from VTEP-1, it will examine the Route-Target that this route carries whether it is equal to the importable EVTP Route-Target of local VRF instance(In this example, the VRF name is "EVPN", which is not a forced name. Users can choose other string for the VRF name.); if it equals then accepts this route, otherwise it will discard. After it accepting this route, VTEP-2 gains the relevant relationship of HostA's MAC address, IP address, L2VNI, L3VNI and VTEP IP address on VTEP-1 (next hop's attribute), and generate a host route to HostA on local vrf routing table, its outbound interface is pointing to the VXLAN tunnel of VTEP-1, VNI is the relevant Layer 3 VNI.

Network Segment Route Synchroization

EVPN network is a private network, which can also communicate with the external network by accessing the external network. BGP imports external routes using Type5 IP prefix routes. Usually, in the Spine-Leaf architecture of EVPN, there are one or more devices specially connected to the external network, which is called Border Leaf. Border Leaf learns routes from the external network through common interfaces and introduces these routes into the EVPN address family to form EVPN-5 routes. This information is then notified to the EVPN network so that other VTEPs can also learn these external routes.

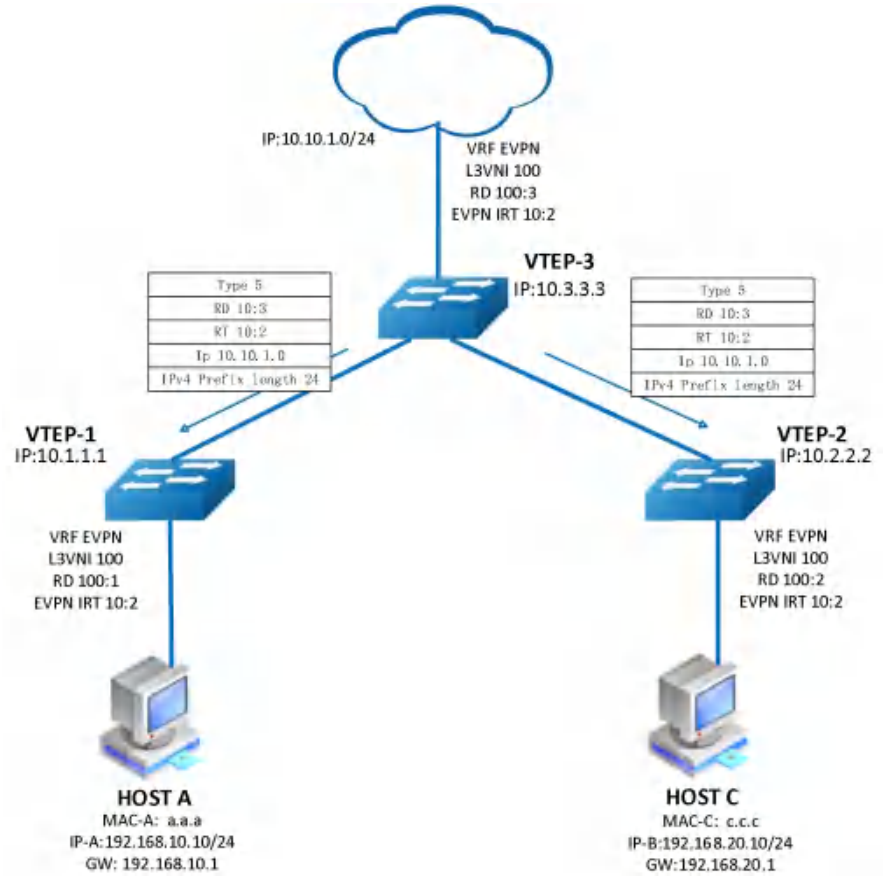


Figure 18-33 Application of Type 5 routes in host IP address advertisement

In the section above, Type 2 routing solves the synchronized problem of 32 bit host routing; however, when communicates with external networks, it cannot be satisfied by networking requirements that only synchronizes 32 bit host routing. Hence, EVPN will solve network segments routing synchronization by Type 5 routing.

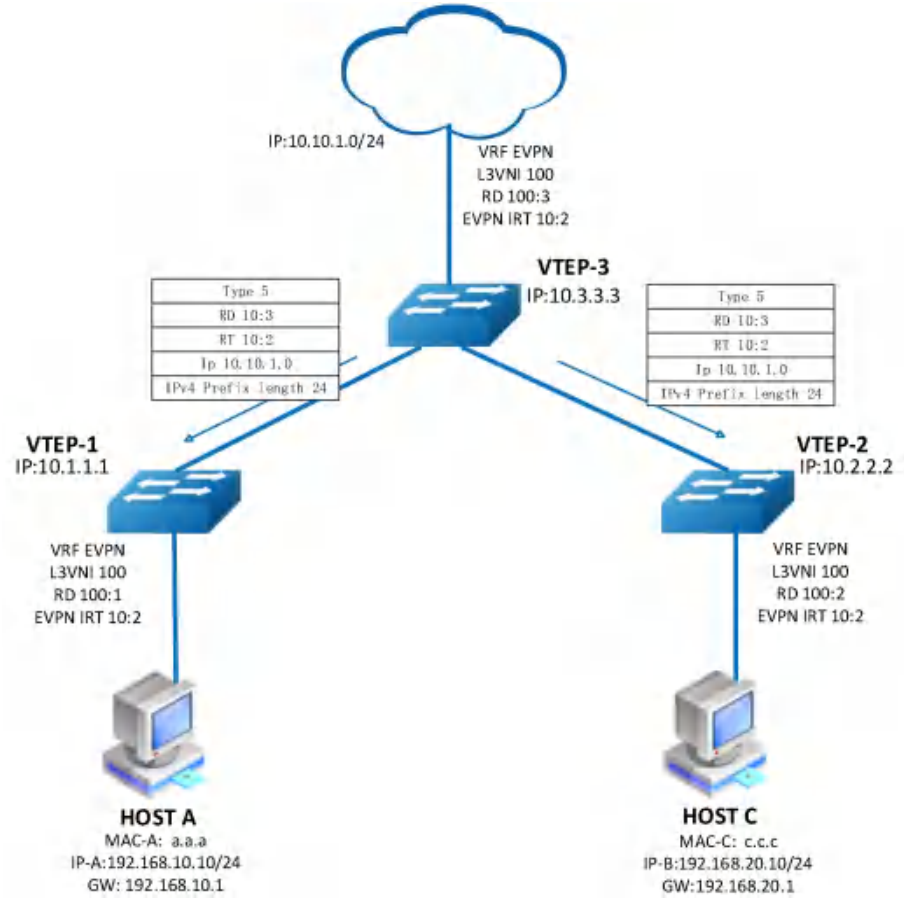


Figure 18-34 Segment route synchronization

As shown in the figure above, vrf EVPN on VTEP-3 has an external route which needs to notify VTEP-1/VTEP-2 to make HostA/HostC can access the address within this network segment. VTEP-3 will send the route information with Layer 3 VNI, RD and EVPN Export Route-Target information to EVPN neighbor (VTEP-1 and VTEP-2). When EVPN neighbor receives this message, it will determine whether the Route-Target from this message is the same as the Import Route-Target of local vrf configuration; if it is same as Import Route-Target, then it accepts this route, take the IPv4 address and mask length as the route's prefix, and its outbound interface is the VXLAN tunnel which is pointing to VTEP-3. VNI is the corresponding Layer 3 VNI.

18.5.2 Application Scenarios

Route-reflector-client used in BGP EVPN network

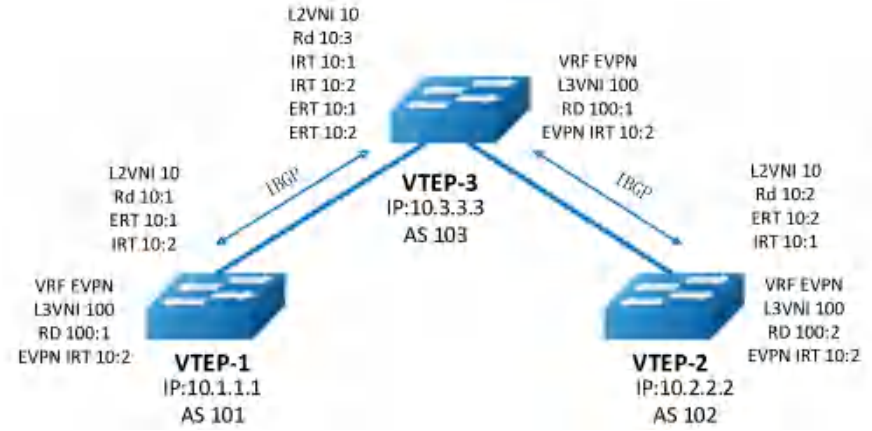


Figure 18-35 Route-reflector-client

As the instance shown in the figure above, VTEP-1/VTEP-2 both established ibgp's evpn neighbors with VTEP-3; by default, routes that are received from ibgp will not send to other ibgp neighbors. At this moment, VTEP-1 and VTEP-2 cannot learn each other's routes. To make VTEP-1 and VTEP-2 can learn each other's route under this situation, we can configure neighbor 10.1.1.1 route-reflector-client/neighbor 10.2.2.2 route-reflector-client under VTEP-3's bgp evpn address family and enable route reflector function. After configuration, VTEP-3 will reflect the route which is received from ibgp neighbor to other ibgp neighbor. At this time, VTEP-1 and VTEP-2 can learn their routes from each other.

Retain route target

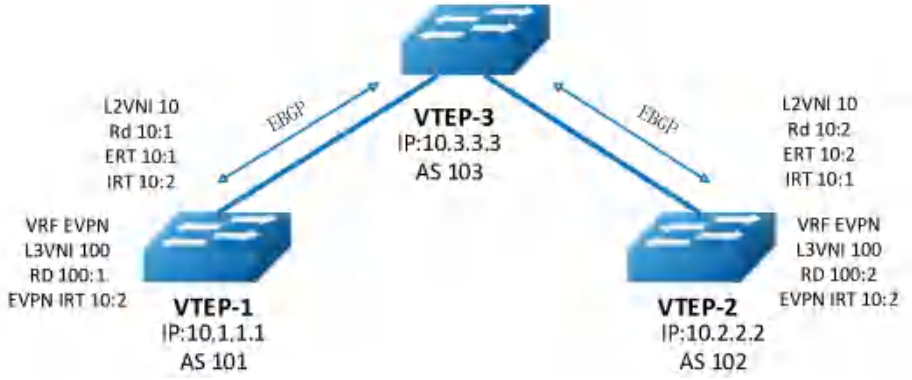


Figure 18-36 Retain route target

As the instance shown in the figure above, VTEP-1/VTEP-2 established ebgp's evpn neighbors with VTEP-3. VTEP-1 and VTEP-2 are both have EVPN instances while VTEP-3 does not deploy EVPN instance; at this time, after VTEP-3 receiving routes from VTEP-1 and VTEP-2, since it has no EVPN instances, it will discard these routes, which leads that VTEP-1 and VTEP-2 cannot learn EVPN routes with each other. To let VTEP-1 and VTEP-2 can learn routes between each other, it can configure retain route-target all under VTEP-3's bgp evpn address family; after configuration, VTEP-3 will save the received bgp evpn routes temporary, and then advertise to its neighbors, so VTEP-1 and VTEP-2 can learn routes between each other.

allowas-in

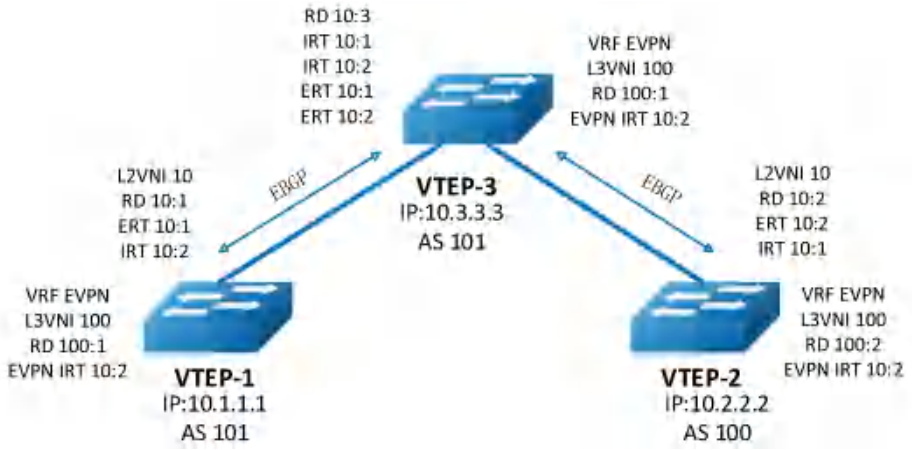


Figure 18-37 allowas-in

As the instance shown above, VTEP-1/VTEP-2 established ebgp's evpn neighbors with VTEP-3; VTEP-1 advertise route to VTEP-2 will pass through VTEP-3; in the BGP EVPN route that received by VTEP-2, AS PATH will contain the local AS number(100) of VTEP-1. Since the AS of VTEP-1 and VTEP-2 are same, BGP does not accept the route notification which contains local AS within AS PATH by default. To let VTEP-1 and VTEP-2 can learn routes between each other, it can configure neighbor 10.3.3.3 allowas-in under VTEP-2's evpn address family; after configuration, BGP will accept AS PATH which contains local AS's route notification.

attribute-unchanged

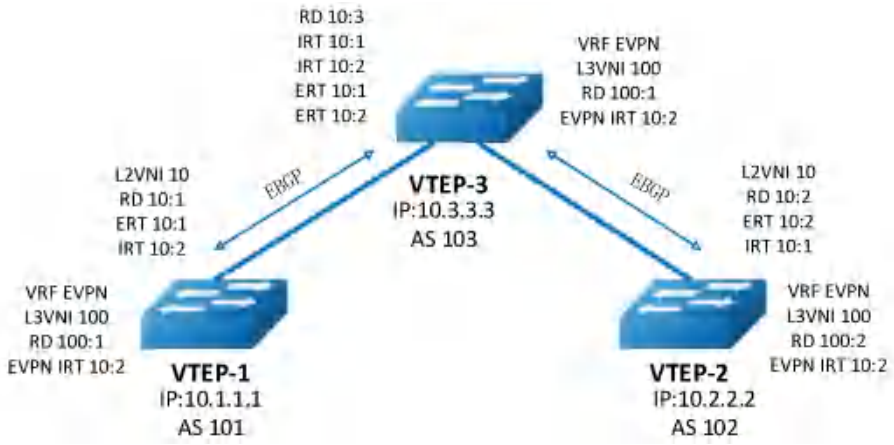


Figure 18-38 attribute-unchanged

As the instance shown above, VTEP-1/VTEP-2 established ebgp's evpn neighbors with VTEP-3; VTEP-1 advertising route to VTEP-2 will pass through VTEP-3. By default, VTEP-3 will replace next hop and other information in the route by VTEP-3's local address when sending the route from VTEP-1 to VTEP-2; at this situation, the tunnel that is created on VTEP-2 will point to VTEP-3. If you want to create tunnel between VTEP-1 and VTEP-2, you can configure neighbor 10.1.1.1 attribute-unchanged/neighbor 10.2.2.2 attribute-unchanged under VTEP-3's bgp evpn address family. At this time, VTEP-3 will not change route's nexthop and other information.

evpn virtual-mac enable

When needs to access into VXLAN network by active-active, we suggest using MLAG metho; the two devices form MLAG should using the same VTEP IP/MAC so other device will treat them as one VTEP， VTEP IP will be configured on NVE interface. After configuring evpn virtual-mac enable, VTEP MAC will use VARP virtual MAC instead of system mac, virtual MAC of two devices in MLAG need to be the same.

18.5.3 Configuration Examples

BGP EVPN Concentrated Gateway

Topology

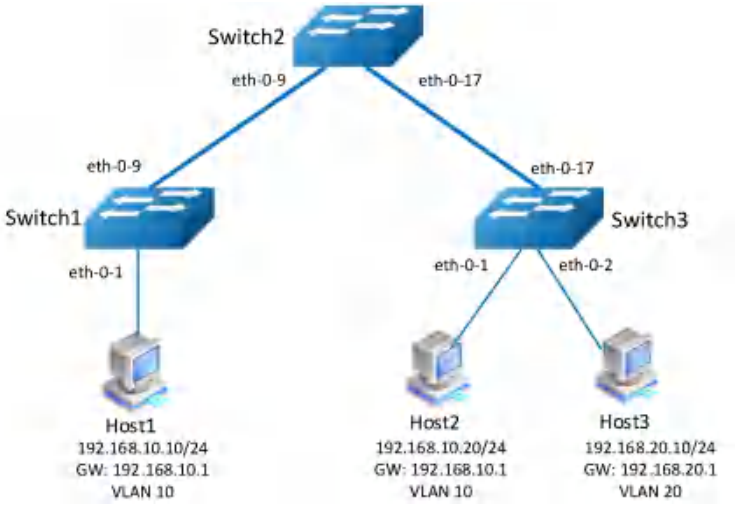


Figure 18-39 BGP EVPN concentrated gateway

Networking requirement

Host 1 and Host 2 are in the same network segment, Host 3 is in another segment; their bottom layer network is Layer 3 network, they can interflow with each other by VXLAN tunnels.

Configuration idea

Switch 1 and Switch 3 create IBGP neighbors with Switch 2 separately, and enable route reflector on Switch 2 at the same time, to make Switch 1 and Switch 3 can learn BGP EVPN route between each other and Host 1 interflows with Host 2. At the same time, deploys concentrated gateway on Switch 2, it needs the traffic interflows across subnets and forward through Switch 2 route.

Configuration Steps

Step 1 Configure routes to make sure the layer 3 network available

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# interface loopback 0
Switch1(config-if)# ip address 10.1.1.1/32
Switch1(config-if)# exit
Switch1(config)# interface eth-0-9
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.1.1/24
Switch1(config-if)# vxlan uplink enable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
Switch1(config)# ip route 10.2.2.2/32 192.168.1.2
Switch1(config)# ip route 10.3.3.3/32 192.168.1.2
Switch1(config)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# interface loopback 0
Switch2(config-if)# ip address 10.2.2.2/32
Switch2(config-if)# exit
Switch2(config)# interface eth-0-9
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.1.2/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.2.1/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# ip route 10.1.1.1/32 192.168.1.1
Switch2(config)# ip route 10.3.3.3/32 192.168.2.2
Switch2(config)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface loopback 0
Switch3(config-if)# ip address 10.3.3.3/32
Switch3(config-if)# exit
Switch3(config)# interface eth-0-17
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 192.168.2.2/24
Switch3(config-if)# vxlan uplink enable
```

```
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# ip route 10.1.1.1/32 192.168.2.1
Switch3(config)# ip route 10.2.2.2/32 192.168.2.1
Switch3(config)# end
```

Step 2 Configure VLAN

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10
Switch1(config-vlan)# vlan 10 overlay enable
Switch1(config-vlan)# exit
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 10
Switch1(config-if)# no shutdown
Switch1(config-if)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 10
Switch2(config-vlan)# vlan 10 overlay enable
Switch2(config-vlan)# vlan 20
Switch2(config-vlan)# vlan 20 overlay enable
Switch2(config-vlan)# exit
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# quit
Switch2(config)# interface vlan 10
Switch2(config-if)# ip address 192.168.10.1/24
Switch2(config-if)# interface vlan 20
Switch2(config-if)# ip address 192.168.20.1/24
Switch2(config-if)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# vlan database
Switch3(config-vlan)# vlan 10
Switch3(config-vlan)# vlan 10 overlay enable
Switch3(config-vlan)# vlan 20
Switch3(config-vlan)# vlan 20 overlay enable
Switch3(config-vlan)# exit
Switch3(config)# interface eth-0-1
Switch3(config-if)# switchport mode trunk
Switch3(config-if)# switchport trunk allowed vlan add 10
Switch3(config-if)# no shutdown
Switch3(config)# interface eth-0-2
Switch3(config-if)# switchport mode trunk
Switch3(config-if)# switchport trunk allowed vlan add 20
Switch3(config-if)# no shutdown
Switch3(config-if)# end
```

Step 3 Configure EVPN instance

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# overlay
Switch1(config-overlay)# vlan 10 vni 10
Switch1(config-overlay)# quit
Switch1(config)# interface nve 0
Switch1(config-if)# source loopback 0
Switch1(config-if)# member vni 10
Switch1(config-if)# quit
Switch1(config)# evpn
Switch1(config-evpn)# vni 10
Switch1(config-evi)# rd 10:1
Switch1(config-evi)# route-target both 10:10
Switch1(config-evpn)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# overlay
Switch2(config-overlay)# vlan 10 vni 10
Switch2(config-overlay)# vlan 20 vni 20
Switch2(config-overlay)# quit
Switch2(config)# interface nve 0
Switch2(config-if)# member vni 10
Switch2(config-if)# member vni 20
Switch2(config-if)# source loopback 0
Switch2(config-if)# quit
Switch2(config)# evpn
Switch2(config-evpn)# vni 10
Switch2(config-evi)# rd 10:2
Switch2(config-evi)# route-target both 10:10
Switch2(config-evi)# vni 20
Switch2(config-evi)# rd 20:2
Switch2(config-evi)# route-target both 20:20
Switch2(config-evi)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# overlay
Switch3(config-overlay)# vlan 10 vni 10
Switch3(config-overlay)# vlan 20 vni 20
Switch3(config-overlay)# exit
Switch3(config)# interface nve 0
Switch3(config-if)# source loopback 0
Switch3(config-if)# member vni 10
Switch3(config-if)# member vni 20
Switch3(config-if)# quit
Switch3(config)# evpn
Switch3(config-evpn)# vni 10
Switch3(config-evi)# rd 10:3
Switch3(config-evi)# route-target both 10:10
```

```
Switch3(config-evi)# vni 20
Switch3(config-evi)# rd 20:3
Switch3(config-evi)# route-target both 20:20
Switch3(config-evi)# end
```

Step 4 Configure BGP

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# router bgp 100
Switch1(config-router)# bgp router-id 10.1.1.1
Switch1(config-router)# neighbor 10.2.2.2 remote-as 100
Switch1(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 10.2.2.2 activate
Switch1(config-router-af)# neighbor 10.2.2.2 send-community both
Switch1(config-router-af)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# router bgp 100
Switch2(config-router)# bgp router-id 10.2.2.2
Switch2(config-router)# neighbor 10.1.1.1 remote-as 100
Switch2(config-router)# neighbor 10.1.1.1 update-source loopback 0
Switch2(config-router)# neighbor 10.3.3.3 remote-as 100
Switch2(config-router)# neighbor 10.3.3.3 update-source loopback 0
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 10.1.1.1 activate
Switch2(config-router-af)# neighbor 10.1.1.1 send-community both
Switch2(config-router-af)# neighbor 10.1.1.1 route-reflector-client
Switch2(config-router-af)# neighbor 10.3.3.3 activate
Switch2(config-router-af)# neighbor 10.3.3.3 send-community both
Switch2(config-router-af)# neighbor 10.3.3.3 route-reflector-client
Switch2(config-router-af)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# router bgp 100
Switch3(config-router)# bgp router-id 10.3.3.3
Switch3(config-router)# neighbor 10.2.2.2 remote-as 100
Switch3(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 10.2.2.2 activate
Switch3(config-router-af)# neighbor 10.2.2.2 send-community both
Switch3(config-router-af)# end
```

Step 5 Validation

Check the network connectivity

```
Switch1# ping -a 10.1.1.1 10.2.2.2
PING 10.2.2.2 (10.2.2.2) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.2.2.2: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 10.2.2.2: icmp_seq=2 ttl=64 time=1.91 ms
```

```
64 bytes from 10.2.2.2: icmp_seq=3 ttl=64 time=2.02 ms
64 bytes from 10.2.2.2: icmp_seq=4 ttl=64 time=1.93 ms
64 bytes from 10.2.2.2: icmp_seq=5 ttl=64 time=1.94 ms

--- 10.2.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.911/1.958/2.022/0.062 ms

Switch1# ping -a 10.1.1.1 10.3.3.3
PING 10.3.3.3 (10.3.3.3) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.3.3.3: icmp_seq=1 ttl=63 time=2.14 ms
64 bytes from 10.3.3.3: icmp_seq=2 ttl=63 time=1.90 ms
64 bytes from 10.3.3.3: icmp_seq=3 ttl=63 time=1.82 ms
64 bytes from 10.3.3.3: icmp_seq=4 ttl=63 time=1.92 ms
64 bytes from 10.3.3.3: icmp_seq=5 ttl=63 time=1.93 ms

--- 10.3.3.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.827/1.946/2.140/0.104 ms
```

Verify the tunnel status

```
Switch1# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni    Type  Remote-vtep IP-Address    Src-Address    Head-end-flooding Protocol
10   10     VxLAN -       10.2.2.2      10.1.1.1       Enable        Evpn
10   10     VxLAN -       10.3.3.3      10.1.1.1       Enable        Evpn

Switch2# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni    Type  Remote-vtep IP-Address    Src-Address    Head-end-flooding Protocol
10   10     VxLAN -       10.1.1.1      10.2.2.2       Enable        Evpn
10   10     VxLAN -       10.3.3.3      10.2.2.2       Enable        Evpn
20   20     VxLAN -       10.3.3.3      10.2.2.2       Enable        Evpn
Switch3# show overlay tunnel
--------------------------------------------------------------------------------
Vlan Vni    Type  Remote-vtep IP-Address    Src-Address    Head-end-flooding Protocol
10   10     VxLAN -       10.1.1.1      10.3.3.3       Enable        Evpn
10   10     VxLAN -       10.2.2.2      10.3.3.3       Enable        Evpn
20   20     VxLAN -       10.2.2.2      10.3.3.3       Enable        Evpn
```

Check the connectivity between the hosts

Host1 ping Host2:

```
PING 192.168.10.20 from 192.168.10.10: 56 data bytes
64 bytes from 192.168.10.20: icmp_seq=1 ttl=64 time=0.947 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.947/0.947/0.947/0.000 ms
```

Host1 ping Host3:

```
PING 192.168.20.10 from 192.168.10.10: 56 data bytes
64 bytes from 192.168.20.10: icmp_seq=1 ttl=63 time=0.613 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.613/0.613/0.613/0.000 ms
```

Host2 ping Host3:

```
PING 192.168.20.10 from 192.168.10.20: 56 data bytes
64 bytes from 192.168.20.10: icmp_seq=1 ttl=63 time=1.131 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 1.131/1.131/1.131/0.000 ms
```

BGP EVPN Distrubuted Gateway

Topology

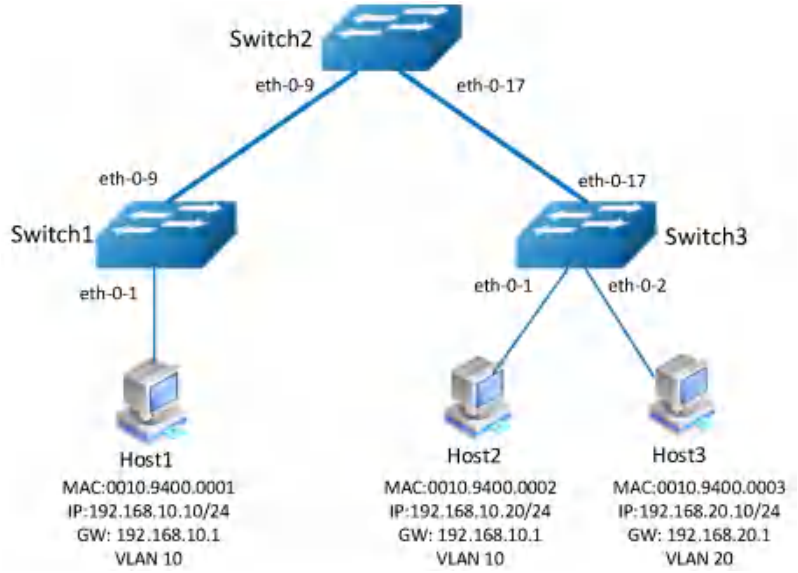


Figure 18-40 BGP EVPN concentrated gateway

Networking requirement

As shown in the figure above, HOST 1/2/3 are belongs to tenement tests, some of then are in the same network segments andsome are not, and they need network with each other. Between Switch 1 and Switch 3 is Layer 3 network, to reduce the pressure of gateway, Switch 1 and Switch 3 need to be distributed gateways for hosts.

Configuration idea

Switch 1 and Switch 3 create IBGP neighbors with Switch 2 separately, and enable route reflector on Switch 2 at the same time, to make Switch 1 and Switch 3 can learn BGP EVPN route between each other. Configure distributed network gateway, make host's gateway deploys on TOR; after enable distributed gateway, vxlan encapsulation or decapsulation will be finished on TOR. Hence, it does not need to configure evpn instance on Switch 2

Configuration Steps

Step 1 Configure routes to make sure the layer 3 network available

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# interface loopback 0
Switch1(config-if)# ip address 10.1.1.1/32
Switch1(config-if)# exit
Switch1(config)# interface eth-0-9
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.1.1/24
Switch1(config-if)# vxlan uplink enable
```

```
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
Switch1(config)# ip route 10.2.2.2/32 192.168.1.2
Switch1(config)# ip route 10.3.3.3/32 192.168.1.2
Switch1(config)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# interface loopback 0
Switch2(config-if)# ip address 10.2.2.2/32
Switch2(config-if)# exit
Switch2(config)# interface eth-0-9
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.1.2/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.2.1/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# ip route 10.1.1.1/32 192.168.1.1
Switch2(config)# ip route 10.3.3.3/32 192.168.2.2
Switch2(config)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface loopback 0
Switch3(config-if)# ip address 10.3.3.3/32
Switch3(config-if)# exit
Switch3(config)# interface eth-0-17
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 192.168.2.2/24
Switch3(config-if)# vxlan uplink enable
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# ip route 10.1.1.1/32 192.168.2.1
Switch3(config)# ip route 10.2.2.2/32 192.168.2.1
Switch3(config)# end
```

Step 2 Configure VLAN

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10
Switch1(config-vlan)# vlan 10 overlay enable
Switch1(config-vlan)# exit
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 10
```

```
Switch1(config-if)# no shutdown
Switch1(config-if)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# vlan database
Switch3(config-vlan)# vlan 10,20
Switch3(config-vlan)# vlan 10 overlay enable
Switch3(config-vlan)# vlan 20 overlay enable
Switch3(config-vlan)# exit
Switch3(config)# interface eth-0-1
Switch3(config-if)# switchport mode trunk
Switch3(config-if)# switchport trunk allowed vlan add 10
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# interface eth-0-2
Switch3(config-if)# switchport mode trunk
Switch3(config-if)# switchport trunk allowed vlan add 20
Switch3(config-if)# no shutdown
Switch3(config-if)# end
```

Step 3 Configure distributed gateway

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# ip vrf evpn
Switch1(config-vrf)# rd 1000:1
Switch1(config-vrf)# route-target import 1000:1000 evpn
Switch1(config-vrf)# vxlan vni 1000
Switch1(config-vrf)# exit
Switch1(config)# interface vlan 10
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# ip address 192.168.10.1/24
Switch1(config-if)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# ip vrf evpn
Switch3(config-vrf)# rd 1000:3
Switch3(config-vrf)# route-target import 1000:1000 evpn
Switch3(config-vrf)# vxlan vni 1000
Switch3(config-vrf)# exit
Switch3(config)# interface vlan 10
Switch3(config-if)# ip vrf forwarding evpn
Switch3(config-if)# overlay distributed-gateway enable
Switch3(config-if)# ip address 192.168.10.1/24
Switch3(config-if)# exit
Switch3(config)# interface vlan 20
Switch3(config-if)# ip vrf forwarding evpn
Switch3(config-if)# overlay distributed-gateway enable
Switch3(config-if)# ip address 192.168.20.1/24
Switch3(config-if)# end
```

Step 4 Configure EVPN instance, enable and distribute the irb route

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# overlay
Switch1(config-overlay)# vlan 10 vni 10
Switch1(config-overlay)# quit
Switch1(config)# interface nve 0
Switch1(config-if)# source loopback 0
Switch1(config-if)# member vni 10
Switch1(config-if)# member vni 1000 associate-vrf
Switch1(config-if)# quit
Switch1(config)# evpn
Switch1(config-evpn)# vni 10
Switch1(config-evi)# rd 10:1
Switch1(config-evi)# route-target both 10:10
Switch1(config-evi)# route-target export 1000:1000
Switch1(config-evi)# quit
Switch1(config-evpn)# quit
Switch1(config)# interface vlan 10
Switch1(config-if)# overlay host-collect enable
Switch1(config-evpn)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# overlay
Switch3(config-overlay)# vlan 10 vni 10
Switch3(config-overlay)# vlan 20 vni 20
Switch3(config-overlay)# exit
Switch3(config)# interface nve 0
Switch3(config-if)# source loopback 0
Switch3(config-if)# member vni 10
Switch3(config-if)# member vni 20
Switch3(config-if)# member vni 1000 associate-vrf
Switch3(config-if)# quit
Switch3(config)# evpn
Switch3(config-evpn)# vni 10
Switch3(config-evi)# rd 10:3
Switch3(config-evi)# route-target both 10:10
Switch3(config-evi)# route-target export 1000:1000
Switch3(config-evi)# vni 20
Switch3(config-evi)# rd 20:3
Switch3(config-evi)# route-target both 20:20
Switch3(config-evi)# route-target export 1000:1000
Switch3(config-evi)# end
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface vlan 10
Switch3(config-if)# overlay host-collect enable
Switch3(config-if)# interface vlan 20
Switch3(config-if)# overlay host-collect enable
Switch3(config-if)# end
```

Step 5 Configure BGP

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# router bgp 100
Switch1(config-router)# bgp router-id 10.1.1.1
Switch1(config-router)# neighbor 10.2.2.2 remote-as 100
Switch1(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 10.2.2.2 activate
Switch1(config-router-af)# neighbor 10.2.2.2 send-community both
Switch1(config-router-af)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# evpn
Switch2(config-evpn)# exit
Switch2(config)# router bgp 100
Switch2(config-router)# bgp router-id 10.2.2.2
Switch2(config-router)# neighbor 10.1.1.1 remote-as 100
Switch2(config-router)# neighbor 10.1.1.1 update-source loopback 0
Switch2(config-router)# neighbor 10.3.3.3 remote-as 100
Switch2(config-router)# neighbor 10.3.3.3 update-source loopback 0
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 10.1.1.1 activate
Switch2(config-router-af)# neighbor 10.1.1.1 send-community both
Switch2(config-router-af)# neighbor 10.1.1.1 route-reflector-client
Switch2(config-router-af)# retain route-target all
Switch2(config-router-af)# neighbor 10.3.3.3 activate
Switch2(config-router-af)# neighbor 10.3.3.3 send-community both
Switch2(config-router-af)# neighbor 10.3.3.3 route-reflector-client
Switch2(config-router-af)# retain route-target all
Switch2(config-router-af)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# router bgp 100
Switch3(config-router)# bgp router-id 10.3.3.3
Switch3(config-router)# neighbor 10.2.2.2 remote-as 100
Switch3(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 10.2.2.2 activate
Switch3(config-router-af)# neighbor 10.2.2.2 send-community both
Switch3(config-router-af)# end
```

Step 6 Validation

Check the network connectivity

```
Switch1# ping -a 10.1.1.1 10.2.2.2
PING 10.2.2.2 (10.2.2.2) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.2.2.2: icmp_seq=1 ttl=64 time=1.97 ms
64 bytes from 10.2.2.2: icmp_seq=2 ttl=64 time=1.90 ms
64 bytes from 10.2.2.2: icmp_seq=3 ttl=64 time=1.90 ms
64 bytes from 10.2.2.2: icmp_seq=4 ttl=64 time=4.70 ms
```

```
64 bytes from 10.2.2.2: icmp_seq=5 ttl=64 time=1.92 ms

--- 10.2.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.901/2.482/4.702/1.110 ms

Switch1# ping -a 10.1.1.1 10.3.3.3
PING 10.3.3.3 (10.3.3.3) from 10.1.1.1 : 56(84) bytes of data.
64 bytes from 10.3.3.3: icmp_seq=1 ttl=63 time=2.05 ms
64 bytes from 10.3.3.3: icmp_seq=2 ttl=63 time=1.91 ms
64 bytes from 10.3.3.3: icmp_seq=3 ttl=63 time=1.93 ms
64 bytes from 10.3.3.3: icmp_seq=4 ttl=63 time=1.89 ms
64 bytes from 10.3.3.3: icmp_seq=5 ttl=63 time=1.93 ms

--- 10.3.3.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.897/1.947/2.051/0.072 ms
```

Verify the MAC/Route

Switch1:

```
Switch1# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   MAC Address     Type      Ports
----   -----------     --------  -----
10     0010.9400.0002   dynamic    VxLAN: 10.1.1.1->10.3.3.3(EI)
10     0010.9400.0001   dynamic    eth-0-1(EO)

Switch1# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     Dc - DHCP Client
     [*] - [AD/Metric]
     * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.1/32 is in local loopback, vlan10
B       192.168.10.20/32 [200/0] is in overlay remote vxlan vtep:10.1.1.1->10.3.3.3, vni:1000
B       192.168.20.10/32 [200/0] is in overlay remote vxlan vtep:10.1.1.1->10.3.3.3, vni:1000
```

Switch3:

```
Switch3# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
```

```
Vlan   MAC Address     Type      Ports
----   -----------     --------  -----
20     0010.9400.0003  dynamic   eth-0-2(EO)
10     0010.9400.0001  dynamic   VxLAN: 10.3.3.3->10.1.1.1(EI)
10     0010.9400.0002  dynamic   eth-0-1(EO)

Switch3# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       Dc - DHCP Client
       [*] - [AD/Metric]
       * - candidate default

C      192.168.10.0/24 is directly connected, vlan10
C      192.168.10.1/32 is in local loopback, vlan10
B      192.168.10.10/32 [200/0] is in overlay remote vxlan vtep:10.3.3.3->10.1.1.1, vni:1000
C      192.168.20.0/24 is directly connected, vlan20
C      192.168.20.1/32 is in local loopback, vlan20
```

Check the connectivity between the hosts

Host1 ping Host2:

```
PING 192.168.10.20 from 192.168.10.10: 56 data bytes
64 bytes from 192.168.10.20: icmp_seq=1 ttl=64 time=0.489 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.489/0.489/0.489/0.000 ms
```

Host1 ping Host3:

```
PING 192.168.20.10 from 192.168.10.10: 56 data bytes
64 bytes from 192.168.20.10: icmp_seq=1 ttl=62 time=0.530 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.530/0.530/0.530/0.000 ms
```

Host2 ping Host3:

```
PING 192.168.20.10 from 192.168.10.20: 56 data bytes
64 bytes from 192.168.20.10: icmp_seq=1 ttl=63 time=0.484 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.484/0.484/0.484/0.000 ms
```

BGP EVPN Import External Route

Topology

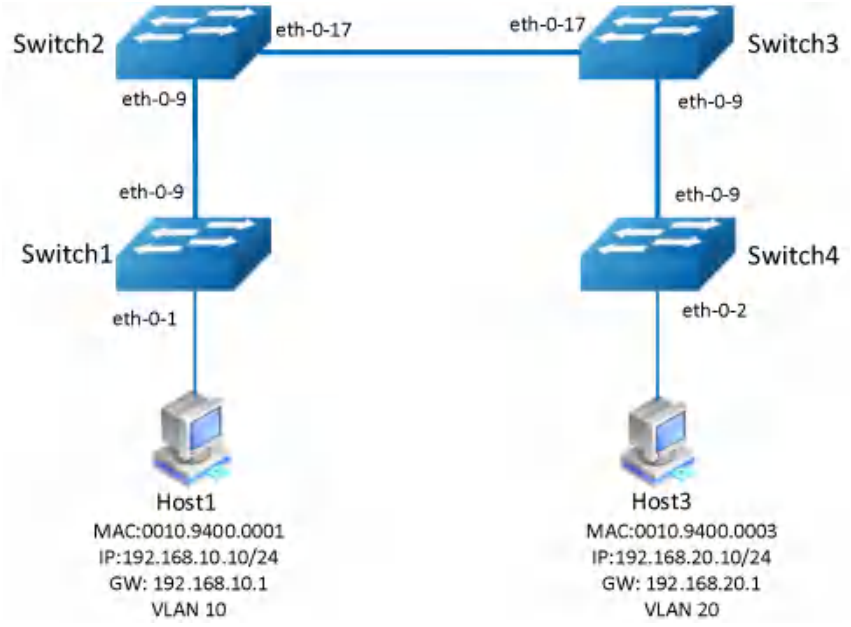


Figure 18-41 BGP EVPN Import External Route

Networking requirement

As shown in the figure above, Switch 2 and Switch 3 belong to public network; Switch 1 and Switch 4 belong to user private network and connect with hosts of two network segments separately and be the gateways of this segment host; two segments are both belongs to the same user who needs route interflow, so think about creating VXLAN tunnel between Switch 2 and Switch 3 to transmit private network route by this vxlan tunnel.

Configuration idea

Switch 1 and Switch 2, Switch 3 and Switch 4 use IBGP to create neighbors under vrf separately and release private routes. Create EBGP neighbors between Switch 2 and Switch 3 which is used to create VXLAN tunnel; and will advertise IBGP private route that is learnt and introduce to BGP EVPN, it will make Switch 1 and Switch 4 can learn network segments route between each other.

Configuration Steps

Step 1 Configure private network, create the private routes

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# ip vrf evpn
Switch1(config-vrf)# rd 1000:1
Switch1(config-vrf)# exit
Switch1(config)# interface eth-0-9
Switch1(config-if)# no switchport
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# ip address 192.168.1.1/24
Switch1(config-if)# no shutdown
Switch1(config-if)# quit
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10
Switch1(config-vlan)# quit
Switch1(config)# interface vlan 10
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# ip address 192.168.10.1/24
Switch1(config-if)# quit
```

```
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 10
Switch1(config-if)# no shutdown
Switch1(config-if)# quit
Switch1(config)# router bgp 100
Switch1(config-router)# address-family ipv4 vrf evpn
Switch1(config-router-af)# network 192.168.10.0 mask 255.255.255.0
Switch1(config-router-af)# neighbor 192.168.1.2 remote-as 100
Switch1(config-router-af)# neighbor 192.168.1.2 activate
Switch1(config-router)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# rd 1000:2
Switch2(config-vrf)# vxlan vni 1000
Switch2(config-vrf)# qui
Switch2(config)# interface eth-0-9
Switch2(config-if)# no switchport
Switch2(config-if)# ip vrf forwarding evpn
Switch2(config-if)# ip address 192.168.1.2/24
Switch2(config-if)# no shutdown
Switch2(config-if)# quit
Switch2(config)# router bgp 100
Switch2(config-router)# address-family IPv4 vrf evpn
Switch2(config-router-af)# advertise l2vpn evpn
Switch2(config-router-af)# neighbor 192.168.1.1 remote-as 100
Switch2(config-router-af)# neighbor 192.168.1.1 activate
Switch2(config-router)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# ip vrf evpn
Switch3(config-vrf)# rd 1000:3
Switch3(config-vrf)# vxlan vni 1000
Switch3(config-vrf)# quit
Switch3(config)# interface eth-0-9
Switch3(config-if)# no switchport
Switch3(config-if)# ip vrf forwarding evpn
Switch3(config-if)# ip address 192.168.2.2/24
Switch3(config-if)# no shutdown
Switch3(config-if)# quit
Switch3(config)# router bgp 200
Switch3(config-router)# address-family ipv4 vrf evpn
Switch3(config-router-af)# advertise l2vpn evpn
Switch3(config-router-af)# neighbor 192.168.2.1 remote-as 200
Switch3(config-router-af)# neighbor 192.168.2.1 activate
Switch3(config-router)# end
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Switch4(config)# ip vrf evpn
Switch4(config-vrf)# rd 1000:4
Switch4(config-vrf)# qui
Switch4(config)# interface eth-0-9
Switch4(config-if)# no switchport
Switch4(config-if)# ip vrf forwarding evpn
Switch4(config-if)# ip address 192.168.2.1/24
Switch4(config-if)# no shutdown
Switch4(config-if)# exit
Switch4(config)# vlan database
Switch4(config-vlan)# vlan 20
Switch4(config-vlan)# quit
Switch4(config)# interface vlan 20
Switch4(config-if)# ip vrf forwarding evpn
Switch4(config-if)# ip address 192.168.20.1/24
Switch4(config-if)# quit
Switch4(config)# interface eth-0-1
Switch4(config-if)# switchport mode trunk
Switch4(config-if)# switchport trunk allowed vlan add 20
Switch4(config-if)# no shutdown
Switch4(config-if)# quit
Switch4(config)# router bgp 200
Switch4(config-router)# address-family ipv4 vrf evpn
Switch4(config-router-af)# network 192.168.20.0 mask 255.255.255.0
Switch4(config-router-af)# neighbor 192.168.2.2 remote-as 200
Switch4(config-router-af)# neighbor 192.168.2.2 activate
Switch4(config-router)# end
```

Step 2 Configure BGP, create BGP evpn neighbors

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# route-target both 1000:1000 evpn
Switch2(config-vrf)# quit
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.17.1/24
Switch2(config-if)# no shutdown
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# quit
Switch2(config)# interface loopback 0
Switch2(config-if)# ip address 10.2.2.2/32
Switch2(config-if)# quit
Switch2(config)# ip route 10.3.3.3/32 192.168.17.2
Switch2(config)# evpn
Switch2(config-evpn)# exit
Switch2(config)# interface nve 0
Switch2(config-if)# source loopback 0
Switch2(config-if)# member vni 1000 associate-vrf
Switch2(config-if)# quit
Switch2(config)# router bgp 100
Switch2(config-router)# bgp router-id 10.2.2.2
Switch2(config-router)# neighbor 10.3.3.3 remote-as 200
Switch2(config-router)# neighbor 10.3.3.3 update-source loopback 0
```

```
Switch2(config-router)# neighbor 10.3.3.3 ebgp-multihop 255
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 10.3.3.3 activate
Switch2(config-router-af)# neighbor 10.3.3.3 send-community both
Switch2(config-router-af)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# ip vrf evpn
Switch3(config-vrf)# route-target both 1000:1000 evpn
Switch3(config-vrf)# quit
Switch3(config)# interface eth-0-17
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 192.168.17.2/24
Switch3(config-if)# no shutdown
Switch3(config-if)# vxlan uplink enable
Switch3(config-if)# quit
Switch3(config)# interface loopback 0
Switch3(config-if)# ip address 10.3.3.3/32
Switch3(config-if)# qui
Switch3(config)# ip route 10.2.2.2/32 192.168.17.1
Switch3(config)# evpn
Switch3(config-evpn)# exit
Switch3(config)# interface nve 0
Switch3(config-if)# source loopback 0
Switch3(config-if)# member vni 1000 associate-vrf
Switch3(config-if)# quit
Switch3(config)# router bgp 200
Switch3(config-router)# bgp router-id 10.3.3.3
Switch3(config-router)# neighbor 10.2.2.2 remote-as 100
Switch3(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch3(config-router)# neighbor 10.2.2.2 ebgp-multihop 255
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 10.2.2.2 activate
Switch3(config-router-af)# neighbor 10.2.2.2 send-community both
Switch3(config-router-af)# end
```

Step 3 Validation

Switch1 can learn remote routes

```
Switch1# show ip bgp vpnv4 vrf evpn
BGP table version is 1, local router ID is 192.168.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inter
nal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1000:1 (Default for VRF evpn)
*> 192.168.10.0/24  0.0.0.0                  100  32768 i
*>i192.168.20.0/24  192.168.1.2           0    100     0 200 i
Switch1#
Switch1# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
```

```
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C      192.168.1.0/24 is directly connected, eth-0-9
C      192.168.1.1/32 is in local loopback, eth-0-9
C      192.168.10.0/24 is directly connected, vlan10
C      192.168.10.1/32 is in local loopback, vlan10
B      192.168.20.0/24 [200/0] via 192.168.1.2, eth-0-9, 00:04:55
```

Switch4 can learn remote routes

```
Switch4# show ip bgp vpnv4 vrf evpn
BGP table version is 1, local router ID is 192.168.20.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
          S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1000:4 (Default for VRF evpn)
*>i192.168.10.0/24  192.168.2.2            0    100     0 100 i
*> 192.168.20.0/24  0.0.0.0                100  32768 i
Switch4#
Switch4# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C      192.168.2.0/24 is directly connected, eth-0-9
C      192.168.2.1/32 is in local loopback, eth-0-9
B      192.168.10.0/24 [200/0] via 192.168.2.2, eth-0-9, 00:07:34
C      192.168.20.0/24 is directly connected, vlan20
C      192.168.20.1/32 is in local loopback, vlan20
```

Host 1 and Host 3 are reachable

```
PING 192.168.20.10 from 192.168.10.10: 56 data bytes

64 bytes from 192.168.20.10: icmp_seq=1 ttl=60 time=0.730 ms



--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.730/0.730/0.730/0.000 ms
```

BGP EVPN segmented VXLAN Tunnel

Topology

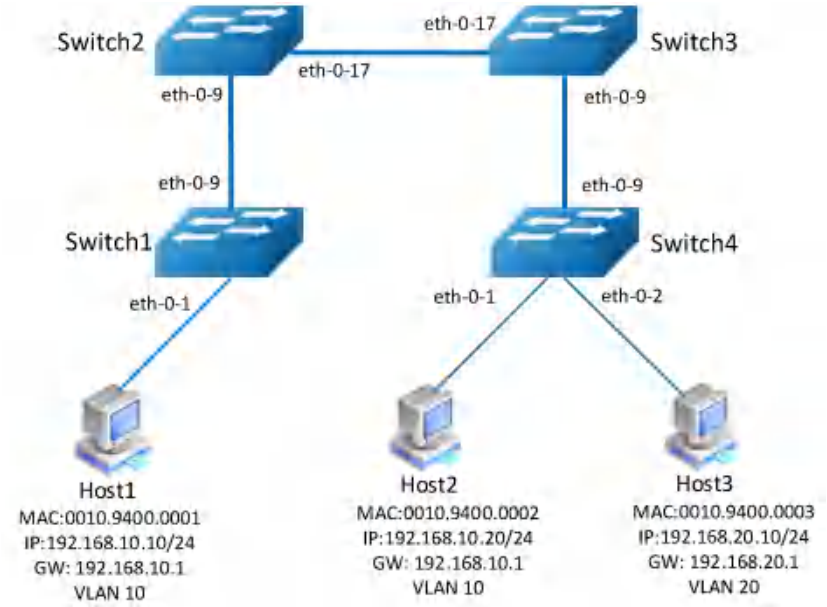


Figure 18-42 GP EVPN segmented VXLAN Tunnel

Networking requirement

As shown in the figure above, HOST1/2/3 are tenement test, it has same segments and it also has different segment, and they need network with each other. Switch 2 and Switch 3 are edge devices, tunnel need to be created by segments, which is the VXLAN tunnel between Switch 1 and Switch 2， Switch 2 and Switch 3， Switch 3 and Switch 4. Finally, it will finish interflow between Hosts.

Configuration idea

EBGP Neighbors are created between switches, and it will change a piece of hop information about releasing routing to its own local address, to reach the goal of creating VXLAN tunnel; at the same time, it will not forward layer 2 traffic with each other since horizontal division between tunnels by default. Therefore, it has to close horizontal division on Switch 2 and Switch 3.

Configuration Steps

Step 1 Configure routes to make sure the layer 3 network available

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# interface loopback 0
Switch1(config-if)# ip address 10.1.1.1/32
Switch1(config-if)# exit
Switch1(config)# interface eth-0-9
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.1.1/24
Switch1(config-if)# vxlan uplink enable
Switch1(config-if)# no shutdown
Switch1(config-if)# exit
Switch1(config)# ip route 10.2.2.2/32 192.168.1.2
Switch1(config)# ip route 10.3.3.3/32 192.168.1.2
Switch1(config)# ip route 10.4.4.4/32 192.168.1.2
Switch1(config)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Switch2(config)# interface loopback 0
Switch2(config-if)# ip address 10.2.2.2/32
Switch2(config-if)# exit
Switch2(config)# interface eth-0-9
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.1.2/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 192.168.2.1/24
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# ip route 10.1.1.1/32 192.168.1.1
Switch2(config)# ip route 10.3.3.3/32 192.168.2.2
Switch2(config)# ip route 10.4.4.4/32 192.168.2.2
Switch2(config)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface loopback 0
Switch3(config-if)# ip address 10.3.3.3/32
Switch3(config-if)# exit
Switch3(config)# interface eth-0-9
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 192.168.3.1/24
Switch3(config-if)# vxlan uplink enable
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# interface eth-0-17
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 192.168.2.2/24
Switch3(config-if)# vxlan uplink enable
Switch3(config-if)# no shutdown
Switch3(config-if)# exit
Switch3(config)# ip route 10.1.1.1/32 192.168.2.1
Switch3(config)# ip route 10.2.2.2/32 192.168.2.1
Switch3(config)# ip route 10.4.4.4/32 192.168.3.2
Switch3(config)# end
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# interface loopback 0
Switch4(config-if)# ip address 10.4.4.4/32
Switch4(config-if)# exit
Switch4(config)# interface eth-0-9
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 192.168.3.2/24
Switch4(config-if)# vxlan uplink enable
Switch4(config-if)# no shutdown
Switch4(config-if)# exit
```

```
Switch4(config)# ip route 10.1.1.1/32 192.168.3.1
Switch4(config)# ip route 10.2.2.2/32 192.168.3.1
Switch4(config)# ip route 10.3.3.3/32 192.168.3.1
Switch4(config)# end
```

Step 2 Configure VLAN

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10
Switch1(config-vlan)# vlan 10 overlay enable
Switch1(config-vlan)# exit
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 10
Switch1(config-if)# no shutdown
Switch1(config-if)# end
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# vlan database
Switch4(config-vlan)# vlan 10,20
Switch4(config-vlan)# vlan 10 overlay enable
Switch4(config-vlan)# vlan 20 overlay enable
Switch4(config-vlan)# exit
Switch4(config)# interface eth-0-1
Switch4(config-if)# switchport mode trunk
Switch4(config-if)# switchport trunk allowed vlan add 10
Switch4(config-if)# no shutdown
Switch4(config-if)# exit
Switch4(config)# interface eth-0-2
Switch4(config-if)# switchport mode trunk
Switch4(config-if)# switchport trunk allowed vlan add 20
Switch4(config-if)# no shutdown
Switch4(config-if)# end
```

Step 3 Configure distributed gateway

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# ip vrf evpn
Switch1(config-vrf)# rd 1000:1
Switch1(config-vrf)# route-target import 1000:1000 evpn
Switch1(config-vrf)# vxlan vni 1000
Switch1(config-vrf)# exit
Switch1(config)# interface vlan 10
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# ip address 192.168.10.1/24
Switch1(config-if)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# rd 1000:2
Switch2(config-vrf)# route-target import 1000:1000 evpn
Switch2(config-vrf)# vxlan vni 1000
Switch2(config-vrf)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# ip vrf evpn
Switch3(config-vrf)# rd 1000:3
Switch3(config-vrf)# route-target import 1000:1000 evpn
Switch3(config-vrf)# vxlan vni 1000
Switch3(config-vrf)# end
Switch3#
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# ip vrf evpn
Switch4(config-vrf)# rd 1000:4
Switch4(config-vrf)# route-target import 1000:1000 evpn
Switch4(config-vrf)# vxlan vni 1000
Switch4(config-vrf)# exit
Switch4(config)# interface vlan 10
Switch4(config-if)# ip vrf forwarding evpn
Switch4(config-if)# overlay distributed-gateway enable
Switch4(config-if)# ip address 192.168.10.1/24
Switch4(config-if)# exit
Switch4(config)# interface vlan 20
Switch4(config-if)# ip vrf forwarding evpn
Switch4(config-if)# overlay distributed-gateway enable
Switch4(config-if)# ip address 192.168.20.1/24
Switch4(config-if)# end
```

Step 4 Configure BGP

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# evpn
Switch1(config-evpn)# exit
Switch1(config)# router bgp 100
Switch1(config-router)# bgp router-id 10.1.1.1
Switch1(config-router)# neighbor 10.2.2.2 remote-as 101
Switch1(config-router)# neighbor 10.2.2.2 ebgp-multihop 255
Switch1(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 10.2.2.2 activate
Switch1(config-router-af)# neighbor 10.2.2.2 send-community both
Switch1(config-router-af)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# evpn
Switch2(config-evpn)# exit
Switch2(config)# router bgp 101
Switch2(config-router)# bgp router-id 10.2.2.2
Switch2(config-router)# neighbor 10.1.1.1 remote-as 100
Switch2(config-router)# neighbor 10.1.1.1 ebgp-multihop 255
Switch2(config-router)# neighbor 10.1.1.1 update-source loopback 0
Switch2(config-router)# neighbor 10.3.3.3 remote-as 201
Switch2(config-router)# neighbor 10.3.3.3 ebgp-multihop 255
Switch2(config-router)# neighbor 10.3.3.3 update-source loopback 0
Switch2(config-router)# address-family l2vpn evpn
Switch2(config-router-af)# neighbor 10.1.1.1 activate
Switch2(config-router-af)# neighbor 10.1.1.1 send-community both
Switch2(config-router-af)# neighbor 10.3.3.3 activate
Switch2(config-router-af)# neighbor 10.3.3.3 send-community both
Switch2(config-router-af)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# evpn
Switch3(config-evpn)# exit
Switch3(config)# router bgp 201
Switch3(config-router)# bgp router-id 10.3.3.3
Switch3(config-router)# neighbor 10.2.2.2 remote-as 101
Switch3(config-router)# neighbor 10.2.2.2 ebgp-multihop 255
Switch3(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch3(config-router)# neighbor 10.4.4.4 remote-as 200
Switch3(config-router)# neighbor 10.4.4.4 ebgp-multihop 255
Switch3(config-router)# neighbor 10.4.4.4 update-source loopback0
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 10.2.2.2 activate
Switch3(config-router-af)# neighbor 10.2.2.2 send-community both
Switch3(config-router-af)# neighbor 10.4.4.4 activate
Switch3(config-router-af)# neighbor 10.4.4.4 send-community both
Switch3(config-router-af)# end
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# evpn
Switch4(config-evpn)# exit
Switch4(config)# router bgp 200
Switch4(config-router)# bgp router-id 10.4.4.4
Switch4(config-router)# neighbor 10.3.3.3 remote-as 201
Switch4(config-router)# neighbor 10.3.3.3 ebgp-multihop 255
Switch4(config-router)# neighbor 10.3.3.3 update-source loopback 0
Switch4(config-router)# address-family l2vpn evpn
Switch4(config-router-af)# neighbor 10.3.3.3 activate
Switch4(config-router-af)# neighbor 10.3.3.3 send-community both
Switch4(config-router-af)# end
```

Step 5 Configure EVPN instance, enable and distribute the irb route

Configuring Switch 1

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# overlay
Switch1(config-overlay)# vlan 10 vni 10
Switch1(config-overlay)# quit
Switch1(config)# interface nve 0
Switch1(config-if)# source loopback 0
Switch1(config-if)# member vni 10
Switch1(config-if)# member vni 1000 associate-vrf
Switch1(config-if)# quit
Switch1(config)# evpn
Switch1(config-evpn)# vni 10
Switch1(config-evi)# rd 10:1
Switch1(config-evi)# route-target both 10:10
Switch1(config-evi)# route-target export 1000:1000
Switch1(config-evi)# quit
Switch1(config-evpn)# quit
Switch1(config)# interface vlan 10
Switch1(config-if)# overlay host-collect enable
Switch1(config-evpn)# end
```

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# interface nve 0
Switch2(config-if)# source loopback 0
Switch2(config-if)# member vni 1000 associate-vrf
Switch2(config-if)# end
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface nve 0
Switch3(config-if)# source loopback 0
Switch3(config-if)# member vni 1000 associate-vrf
Switch3(config-if)# end
```

Configuring Switch 4

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# overlay
Switch4(config-overlay)# vlan 10 vni 10
Switch4(config-overlay)# vlan 20 vni 20
Switch4(config-overlay)# exit
Switch4(config)# interface nve 0
Switch4(config-if)# source loopback 0
Switch4(config-if)# member vni 10
Switch4(config-if)# member vni 20
Switch4(config-if)# member vni 1000 associate-vrf
Switch4(config-if)# quit
Switch4(config)# evpn
Switch4(config-evpn)# vni 10
Switch4(config-evi)# rd 10:4
Switch4(config-evi)# route-target both 10:10
```

```
Switch4(config-evi)# route-target export 1000:1000
Switch4(config-evi)# vni 20
Switch4(config-evi)# rd 20:4
Switch4(config-evi)# route-target both 20:20
Switch4(config-evi)# route-target export 1000:1000
Switch4(config-evi)# end
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# interface vlan 10
Switch4(config-if)# overlay host-collect enable
Switch4(config-if)# interface vlan 20
Switch4(config-if)# overlay host-collect enable
Switch4(config-if)# end
```

Step 6 Configure EVPN instance, disable split horizon

Configuring Switch 2

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 10
Switch2(config-vlan)# vlan 10 overlay enable
Switch2(config-vlan)# exit
Switch2(config)# overlay
Switch2(config-overlay)# vlan 10 vni 10
Switch2(config-overlay)# exit
Switch2(config)# evpn
Switch2(config-evpn)# vni 10
Switch2(config-evi)# rd 10:2
Switch2(config-evi)# route-target import 1000:1000
Switch2(config-evi)# exit
Switch2(config-evpn)# exit
Switch2(config)# interface nve 0
Switch2(config-if)# member vni 10
Switch2(config-if)# split-horizon disable
Switch2(config-if)# end
Switch2# clear bgp evpn *
```

Configuring Switch 3

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# vlan database
Switch3(config-vlan)# vlan 10
Switch3(config-vlan)# vlan 10 overlay enable
Switch3(config-vlan)# exit
Switch3(config)# overlay
Switch3(config-overlay)# vlan 10 vni 10
Switch3(config-overlay)# exit
Switch3(config)# evpn
Switch3(config-evpn)# vni 10
Switch3(config-evi)# rd 10:3
Switch3(config-evi)# route-target import 1000:1000
Switch3(config-evi)# exit
Switch3(config-evpn)# exit
Switch3(config)# interface nve 0
Switch3(config-if)# member vni 10
```

```
Switch3(config-if)# split-horizon disable
Switch3(config-if)# end
Switch3# clear bgp evpn *
```

Step 7 Validation

Switch1:

```
Switch1# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   MAC Address     Type      Ports
----   -----------     --------  -----
10     0010.9400.0002   dynamic    VxLAN: 10.1.1.1->10.2.2.2(EI)
10     0010.9400.0001   dynamic    eth-0-1(EO)
Switch1# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     Dc - DHCP Client
     [*] - [AD/Metric]
     * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.1/32 is in local loopback, vlan10
B       192.168.10.20/32 [20/0] is in overlay remote vxlan vtep:10.1.1.1->10.2.2.2, vni:1000
B       192.168.20.10/32 [20/0] is in overlay remote vxlan vtep:10.1.1.1->10.2.2.2, vni:1000
Switch1# show ip arp vrf evpn
 Protocol   Address         Age (min)  Hardware Addr   Interface
 Internet   192.168.10.1          -   001e.080a.73d6  vlan10
 Internet   192.168.10.10         1   0010.9400.0001  vlan10
```

Switch2:

```
Switch2# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   MAC Address     Type      Ports
----   -----------     --------  -----
10     0010.9400.0002   dynamic    VxLAN: 10.2.2.2->10.3.3.3(EI)
10     0010.9400.0001   dynamic    VxLAN: 10.2.2.2->10.1.1.1(EI)
Switch2# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
     O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     Dc - DHCP Client
```

```
    [*] - [AD/Metric]
    * - candidate default

B      192.168.10.10/32 [20/0] is in overlay remote vxlan vtep:10.2.2.2->10.1.1.1, vni:1000
B      192.168.10.20/32 [20/0] is in overlay remote vxlan vtep:10.2.2.2->10.3.3.3, vni:1000
B      192.168.20.10/32 [20/0] is in overlay remote vxlan vtep:10.2.2.2->10.3.3.3, vni:1000
```

Switch3:

```
Switch3# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   MAC Address      Type      Ports
----   -----------      --------  -----
10     0010.9400.0001   dynamic   VxLAN: 10.3.3.3->10.2.2.2(EI)
10     0010.9400.0002   dynamic   VxLAN: 10.3.3.3->10.4.4.4(EI)
Switch3# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

B      192.168.10.10/32 [20/0] is in overlay remote vxlan vtep:10.3.3.3->10.2.2.2, vni:1000
B      192.168.10.20/32 [20/0] is in overlay remote vxlan vtep:10.3.3.3->10.4.4.4, vni:1000
B      192.168.20.10/32 [20/0] is in overlay remote vxlan vtep:10.3.3.3->10.4.4.4, vni:1000
```

Switch4:

```
Switch4# show mac address-table
       MAC Address Table
-------------------------------------------
(*)  - Security Entry    (M)  - MLAG Entry
(MO) - MLAG Output Entry  (MI) - MLAG Input Entry
(E)  - EVPN Entry       (EO) - EVPN Output Entry
(EI) - EVPN Input Entry
Vlan   MAC Address      Type      Ports
----   -----------      --------  -----
20     0010.9400.0003   dynamic   eth-0-2(EO)
10     0010.9400.0001   dynamic   VxLAN: 10.4.4.4->10.3.3.3(EI)
10     0010.9400.0002   dynamic   eth-0-1(EO)
Switch4# show ip arp vrf evpn
 Protocol   Address        Age (min)  Hardware Addr   Interface
 Internet   192.168.10.1         -   001e.080a.72ae  vlan10
 Internet   192.168.10.20        4   0010.9400.0002  vlan10
 Internet   192.168.20.1         -   001e.080a.72ae  vlan20
 Internet   192.168.20.10        4   0010.9400.0003  vlan20
Switch4# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
```

```
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
Dc - DHCP Client
[*] - [AD/Metric]
* - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.1/32 is in local loopback, vlan10
B       192.168.10.10/32 [20/0] is in overlay remote vxlan vtep:10.4.4.4->10.3.3.3, vni:1000
C       192.168.20.0/24 is directly connected, vlan20
C       192.168.20.1/32 is in local loopback, vlan20
```

MLAG TOR in EVPN Network

Topology

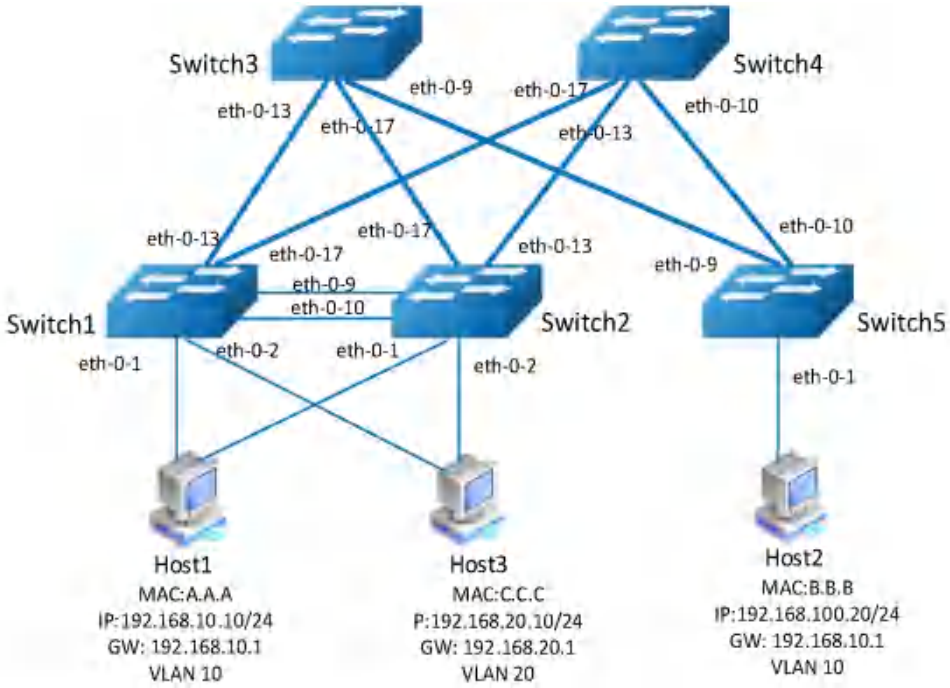


Figure 18-43 MLAG TOR in EVPN Network

Networking requirement

Networking requirement

As shown in the figure above, SWITCH 1/2/5 are TOR switches, SWITCH 3/4 are Spine, HOSTs are belongs to the same user, it requires to guarantee that hosts from the same network segments and different segments can interflow with each other when isolate with other users. To make sure the reliability, some hosts need to access by active-active.

Configuration idea

Configure MLAG on SWTIH 1/2 and setup virtual ip as host gateway to guarantee active-active access; to make sure that virtual machine immigrates does not change gateway address, deploy distributed gateway and use the same ip on SWITH A/B/E. Using IBGP to build EVPN network, Switch 1 and Switch 2 use the same VTEP address and are shown as a single VTEP externally; enables route reflector on Switch 3 and Switch 4 to reduce the number of BGP neighbors and suppose SWITCH 1/2/5 to learn EVPN route.

Configuration Steps

Step 1 Configure MLAG

configure MLAG on Switch 1/2, which is used for active-active host accesses, Switch 5 uses single interface connection.

configure peer-link

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# interface range eth-0-9 to eth-0-10
Switch1(config-if-range)# no shutdown
Switch1(config-if-range)# static-channel-group 55
Switch1(config-if-range)# exit
Switch1(config)# interface agg 55
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan all
Switch1(config-if)# spanning-tree port disable
Switch1(config-if)# exit
Switch1(config)# mlag configuration
Switch1(config-mlag)# peer-link agg 55
Switch1(config-mlag)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# interface range eth-0-9 to eth-0-10
Switch2(config-if-range)# no shutdown
Switch2(config-if-range)# static-channel-group 55
Switch2(config-if-range)# exit
Switch2(config)# interface agg 55
Switch2(config-if)# switchport mode trunk
Switch2(config-if)# switchport trunk allowed vlan all
Switch2(config-if)# spanning-tree port disable
Switch2(config-if)# exit
Switch2(config)# mlag configuration
Switch2(config-mlag)# peer-link agg 55
Switch2(config-mlag)# end
```

configure peer-address

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 4094
Switch1(config-vlan)# exit
Switch1(config)# interface vlan 4094
Switch1(config-if)# ip address 40.94.0.1/24
Switch1(config-if)# exit
Switch1(config)# mlag configuration
Switch1(config-mlag)# peer-address 40.94.0.2
Switch1(config-mlag)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 4094
```

```
Switch2(config-vlan)# exit
Switch2(config)# interface vlan 4094
Switch2(config-if)# ip address 40.94.0.2/24
Switch2(config-if)# exit
Switch2(config)# mlag configuration
Switch2(config-mlag)# peer-address 40.94.0.1
Switch2(config-mlag)# end
```

configure downlink interface

SWITC 1/2 uses MLAG double downlink, Switch 5 uses normal single downlink

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10,20
Switch1(config-vlan)# exit
Switch1(config)# interface eth-0-1
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 10
Switch1(config-if)# no shutdown
Switch1(config-if)# static-channel-group 1
Switch1(config-if)# exit
Switch1(config)# interface eth-0-2
Switch1(config-if)# switchport mode trunk
Switch1(config-if)# switchport trunk allowed vlan add 20
Switch1(config-if)# no shutdown
Switch1(config-if)# static-channel-group 2
Switch1(config-if)# exit
Switch1(config)# interface agg 1
Switch1(config-if)# mlag 1
Switch1(config-if)# exit
Switch1(config)# interface agg 2
Switch1(config-if)# mlag 2
Switch1(config-if)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 10,20
Switch2(config-vlan)# exit
Switch2(config)# interface eth-0-1
Switch2(config-if)# switchport mode trunk
Switch2(config-if)# switchport trunk allowed vlan add 10
Switch2(config-if)# static-channel-group 1
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface eth-0-2
Switch2(config-if)# switchport mode trunk
Switch2(config-if)# switchport trunk allowed vlan add 20
Switch2(config-if)# static-channel-group 2
Switch2(config-if)# no shutdown
Switch2(config-if)# exit
Switch2(config)# interface agg 1
```

```
Switch2(config-if)# mlag 1
Switch2(config-if)# exit
Switch2(config)# interface agg 2
Switch2(config-if)# mlag 2
Switch2(config-if)# end
```

Switch5:

```
Switch5# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch5(config)# vlan database
Switch5(config-vlan)# vlan 10
Switch5(config-vlan)# exit
Switch5(config)# interface eth-0-1
Switch5(config-if)# switchport mode trunk
Switch5(config-if)# switchport trunk allowed vlan add 10
Switch5(config-if)# no shutdown
Switch5(config-if)# end
```

Step 2 Setup gateway's address and enable distributed gateway, use virtual ip on MLAG devices

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# ip vrf evpn
Switch1(config-vrf)# rd 1000:1
Switch1(config-vrf)# vxlan vni 1000
Switch1(config-vrf)# exit
Switch1(config)# interface vlan 10
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# ip address 192.168.10.253/24
Switch1(config-if)# ip virtual-router address 192.168.10.1
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# exit
Switch1(config)# interface vlan 20
Switch1(config-if)# ip vrf forwarding evpn
Switch1(config-if)# ip address 192.168.20.253/24
Switch1(config-if)# ip virtual-router address 192.168.20.1
Switch1(config-if)# overlay distributed-gateway enable
Switch1(config-if)# exit
Switch1(config)# ip virtual-router mac 0.0.1
Switch1(config)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# rd 1000:2
Switch2(config-vrf)# vxlan vni 1000
Switch2(config-vrf)# exit
Switch2(config)# interface vlan 10
Switch2(config-if)# ip vrf forwarding evpn
Switch2(config-if)# ip address 192.168.10.254/24
Switch2(config-if)# ip virtual-router address 192.168.10.1
Switch2(config-if)# overlay distributed-gateway enable
Switch2(config-if)# exit
Switch2(config)# interface vlan 20
```

```
Switch2(config-if)# ip vrf forwarding evpn
Switch2(config-if)# ip address 192.168.20.254/24
Switch2(config-if)# ip virtual-router address 192.168.20.1
Switch2(config-if)# overlay distributed-gateway enable
Switch2(config-if)# exit
Switch2(config)# ip virtual-router mac 0.0.1
Switch2(config)# end
```

Switch5:

```
Switch5# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch5(config)# ip vrf evpn
Switch5(config-vrf)# rd 1000:5
Switch5(config-vrf)# vxlan vni 1000
Switch5(config-vrf)# qui
Switch5(config)# interface vlan 10
Switch5(config-if)# ip vrf forwarding evpn
Switch5(config-if)# ip address 192.168.10.254/24
Switch5(config-if)# ip virtual-router address 192.168.10.1
Switch5(config-if)# overlay distributed-gateway enable
Switch5(config-if)# exit
Switch5(config)# ip virtual-router mac 0.0.2
Switch5(config)# end
```

Step 3 Configure routes to make sure the layer 3 network available

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# interface loopback 0
Switch1(config-if)# ip address 10.1.1.1/32
Switch1(config-if)# exit
Switch1(config)# interface loopback 1
Switch1(config-if)# ip address 10.11.11.11/32
Switch1(config-if)# exit
Switch1(config)# interface eth-0-13
Switch1(config-if)# no switchport
Switch1(config-if)# no shutdown
Switch1(config-if)# vxlan uplink enable
Switch1(config-if)# ip address 13.1.1.1/24
Switch1(config-if)# exit
Switch1(config)# interface eth-0-17
Switch1(config-if)# no switchport
Switch1(config-if)# no shutdown
Switch1(config-if)# vxlan uplink enable
Switch1(config-if)# ip address 17.1.1.1/24
Switch1(config-if)# exit
Switch1(config)# router ospf
Switch1(config-router)# router-id 10.1.1.1
Switch1(config-router)# network 10.1.1.1 0.0.0.0 area 0
Switch1(config-router)# network 10.11.11.11 0.0.0.0 area 0
Switch1(config-router)# network 13.1.1.0 0.0.0.255 area 0
Switch1(config-router)# network 17.1.1.0 0.0.0.255 area 0
Switch1(config)#end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# interface loopback 0
Switch2(config-if)# ip address 10.2.2.2/32
Switch2(config-if)# exit
Switch2(config)# interface loopback 1
Switch2(config-if)# ip address 10.11.11.11/32
Switch2(config-if)# exit
Switch2(config)# interface eth-0-13
Switch2(config-if)# no switchport
Switch2(config-if)# no shutdown
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# ip address 13.2.2.1/24
Switch2(config-if)# exit
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# no shutdown
Switch2(config-if)# vxlan uplink enable
Switch2(config-if)# ip address 17.2.2.1/24
Switch2(config-if)# exit
Switch2(config)# router ospf
Switch2(config-router)# router-id 10.2.2.2
Switch2(config-router)# network 10.2.2.2 0.0.0.0 area 0
Switch2(config-router)# network 10.11.11.11 0.0.0.0 area 0
Switch2(config-router)# network 13.2.2.0 0.0.0.255 area 0
Switch2(config-router)# network 17.2.2.0 0.0.0.255 area 0
Switch2(config)# end
```

Switch3:

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# interface loopback 0
Switch3(config-if)# ip address 10.3.3.3/32
Switch3(config-if)# exit
Switch3(config)# interface eth-0-9
Switch3(config-if)# no shutdown
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 9.1.1.2/24
Switch3(config-if)# exit
Switch3(config)# interface eth-0-13
Switch3(config-if)# no shutdown
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 13.1.1.2/24
Switch3(config-if)# exit
Switch3(config)# interface eth-0-17
Switch3(config-if)# no shutdown
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 17.2.2.2/24
Switch3(config-if)# exit
Switch3(config)# router ospf
Switch3(config-router)# router-id 10.3.3.3
Switch3(config-router)# network 9.1.1.0 0.0.0.255 area 0
Switch3(config-router)# network 13.1.1.0 0.0.0.255 area 0
Switch3(config-router)# network 17.2.2.0 0.0.0.255 area 0
```

```
Switch3(config-router)# network 10.3.3.3 0.0.0.0 area 0
Switch3(config)# end
```

Switch4:

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# interface loopback 0
Switch4(config-if)# ip address 10.4.4.4/32
Switch4(config-if)# exit
Switch4(config)# interface eth-0-10
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 10.10.10.2/24
Switch4(config-if)# exit
Switch4(config)# interface eth-0-13
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 13.2.2.2/24
Switch4(config-if)# exit
Switch4(config)# interface eth-0-17
Switch4(config-if)# no shutdown
Switch4(config-if)# no switchport
Switch4(config-if)# ip address 17.1.1.2/24
Switch4(config-if)# exit
Switch4(config)# router ospf
Switch4(config-router)# router-id 10.4.4.4
Switch4(config-router)# network 10.10.10.0 0.0.0.255 area 0
Switch4(config-router)# network 13.2.2.0 0.0.0.255 area 0
Switch4(config-router)# network 17.1.1.0 0.0.0.255 area 0
Switch4(config-router)# network 10.4.4.4 0.0.0.0 area 0
Switch4(config)# end
```

Switch5:

```
Switch5# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch5(config)# interface loopback 0
Switch5(config-if)# ip address 10.5.5.5/32
Switch5(config-if)# exit
Switch5(config)# interface eth-0-9
Switch5(config-if)# no switchport
Switch5(config-if)# no shutdown
Switch5(config-if)# vxlan uplink enable
Switch5(config-if)# ip address 9.1.1.1/24
Switch5(config-if)# exit
Switch5(config)# interface eth-0-10
Switch5(config-if)# no switchport
Switch5(config-if)# no shutdown
Switch5(config-if)# vxlan uplink enable
Switch5(config-if)# ip address 10.10.10.1/24
Switch5(config-if)# exit
Switch5(config)# router ospf
Switch5(config-router)# router-id 10.5.5.5
Switch5(config-router)# network 9.1.1.0 0.0.0.255 area 0
Switch5(config-router)# network 10.10.10.0 0.0.0.255 area 0
```

```
Switch5(config-router)# network 10.5.5.5 0.0.0.0 area 0
Switch5(config)#end
```

Step 4 Configure BGP EVPN Neighbors

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# evpn
Switch1(config-evpn)# exit
Switch1(config)# router bgp 100
Switch1(config-router)# bgp router-id 10.1.1.1
Switch1(config-router)# neighbor 10.3.3.3 remote-as 100
Switch1(config-router)# neighbor 10.3.3.3 update-source loopback 0
Switch1(config-router)# neighbor 10.4.4.4 remote-as 100
Switch1(config-router)# neighbor 10.4.4.4 update-source loopback 0
Switch1(config-router)# address-family l2vpn evpn
Switch1(config-router-af)# neighbor 10.3.3.3 activate
Switch1(config-router-af)# neighbor 10.3.3.3 send-community both
Switch1(config-router-af)# neighbor 10.4.4.4 activate
Switch1(config-router-af)# neighbor 10.4.4.4 send-community both
Switch1(config-router-af)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# evpn
Switch2(config-evpn)# exit
Switch2(config)# router bgp 100
Switch2(config-router)# bgp router-id 10.2.2.2
Switch2(config-router)#  neighbor 10.3.3.3 remote-as 100
Switch2(config-router)#  neighbor 10.3.3.3 update-source loopback0
Switch2(config-router)#  neighbor 10.4.4.4 remote-as 100
Switch2(config-router)#  neighbor 10.4.4.4 update-source loopback0
Switch2(config-router)#  address-family l2vpn evpn
Switch2(config-router-af)# neighbor 10.3.3.3 activate
Switch2(config-router-af)# neighbor 10.3.3.3 send-community both
Switch2(config-router-af)# neighbor 10.4.4.4 activate
Switch2(config-router-af)# neighbor 10.4.4.4 send-community both
Switch2(config-router-af)# end
```

Switch3:

```
Switch3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch3(config)# evpn
Switch3(config-evpn)# exit
Switch3(config)# router bgp 100
Switch3(config-router)# bgp router-id 10.3.3.3
Switch3(config-router)# neighbor 10.1.1.1 remote-as 100
Switch3(config-router)# neighbor 10.1.1.1 update-source loopback 0
Switch3(config-router)# neighbor 10.2.2.2 remote-as 100
Switch3(config-router)# neighbor 10.2.2.2 update-source loopback 0
Switch3(config-router)# neighbor 10.5.5.5 remote-as 100
Switch3(config-router)# neighbor 10.5.5.5 update-source loopback 0
Switch3(config-router)# address-family l2vpn evpn
Switch3(config-router-af)# neighbor 10.1.1.1 activate
```

```
Switch3(config-router-af)# neighbor 10.1.1.1 send-community both
Switch3(config-router-af)# neighbor 10.1.1.1 route-reflector-client
Switch3(config-router-af)# neighbor 10.2.2.2 activate
Switch3(config-router-af)# neighbor 10.2.2.2 send-community both
Switch3(config-router-af)# neighbor 10.2.2.2 route-reflector-client
Switch3(config-router-af)# neighbor 10.5.5.5 activate
Switch3(config-router-af)# neighbor 10.5.5.5 send-community both
Switch3(config-router-af)# neighbor 10.5.5.5 route-reflector-client
Switch3(config-router-af)# end
```

Switch4:

```
Switch4# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch4(config)# evpn
Switch4(config-evpn)# exit
Switch4(config)# router bgp 100
Switch4(config-router)# bgp router-id 10.4.4.4
Switch4(config-router)# neighbor 10.1.1.1 remote-as 100
Switch4(config-router)# neighbor 10.1.1.1 update-source loopback0
Switch4(config-router)# neighbor 10.2.2.2 remote-as 100
Switch4(config-router)# neighbor 10.2.2.2 update-source loopback0
Switch4(config-router)# neighbor 10.5.5.5 remote-as 100
Switch4(config-router)# neighbor 10.5.5.5 update-source loopback0
Switch4(config-router)#  address-family l2vpn evpn
Switch4(config-router-af)# neighbor 10.1.1.1 activate
Switch4(config-router-af)# neighbor 10.1.1.1 route-reflector-client
Switch4(config-router-af)# neighbor 10.1.1.1 send-community both
Switch4(config-router-af)# neighbor 10.2.2.2 activate
Switch4(config-router-af)# neighbor 10.2.2.2 route-reflector-client
Switch4(config-router-af)# neighbor 10.2.2.2 send-community both
Switch4(config-router-af)# neighbor 10.5.5.5 activate
Switch4(config-router-af)# neighbor 10.5.5.5 route-reflector-client
Switch4(config-router-af)# neighbor 10.5.5.5 send-community both
Switch4(config-router-af)# end
```

Switch5:

```
Switch5# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch5(config)# evpn
Switch5(config-evpn)# exit
Switch5(config)# router bgp 100
Switch5(config-router)# bgp router-id 10.5.5.5
Switch5(config-router)# neighbor 10.3.3.3 remote-as 100
Switch5(config-router)# neighbor 10.3.3.3 update-source loopback0
Switch5(config-router)# neighbor 10.4.4.4 remote-as 100
Switch5(config-router)# neighbor 10.4.4.4 update-source loopback0
Switch5(config-router)# address-family l2vpn evpn
Switch5(config-router-af)# neighbor 10.3.3.3 activate
Switch5(config-router-af)# neighbor 10.3.3.3 send-community both
Switch5(config-router-af)# neighbor 10.4.4.4 activate
Switch5(config-router-af)# neighbor 10.4.4.4 send-community both
Switch5(config-router-af)# end
```

Step 5 Configure EVPN instance

Switch1:

```
Switch1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)# evpn virtual-mac enable
Switch1(config)# vlan database
Switch1(config-vlan)# vlan 10 overlay enable
Switch1(config-vlan)# vlan 20 overlay enable
Switch1(config-vlan)# qui
Switch1(config)# overlay
Switch1(config-overlay)# vlan 10 vni 10
Switch1(config-overlay)# vlan 20 vni 20
Switch1(config-overlay)# quit
Switch1(config)# interface vlan 10
Switch1(config-if)# overlay host-collect enable
Switch1(config-if)# interface vlan 20
Switch1(config-if)# overlay host-collect enable
Switch1(config-if)# exit
Switch1(config)# evpn
Switch1(config-evpn)# vni 10
Switch1(config-evi)# rd 10:1
Switch1(config-evi)# route-target both 10:10
Switch1(config-evi)# route-target export 1000:1000
Switch1(config-evi)# vni 20
Switch1(config-evi)# rd 20:1
Switch1(config-evi)# route-target both 20:20
Switch1(config-evi)# route-target export 1000:1000
Switch1(config-evi)# exit
Switch1(config-evpn)# exit
Switch1(config)# ip vrf evpn
Switch1(config-vrf)# route-target import 1000:1000 evpn
Switch1(config-vrf)# exit
Switch1(config)# interface nve 0
Switch1(config-if)# source loopback 1
Switch1(config-if)# member vni 10
Switch1(config-if)# member vni 20
Switch1(config-if)# member vni 1000 associate-vrf
Switch1(config-if)# end
```

Switch2:

```
Switch2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch2(config)# evpn virtual-mac enable
Switch2(config)# vlan database
Switch2(config-vlan)# vlan 10 overlay enable
Switch2(config-vlan)# vlan 20 overlay enable
Switch2(config-vlan)# quit
Switch2(config)# overlay
Switch2(config-overlay)# vlan 10 vni 10
Switch2(config-overlay)# vlan 20 vni 20
Switch2(config-overlay)# qui
Switch2(config)# interface vlan 10
Switch2(config-if)# overlay host-collect enable
Switch2(config-if)# interface vlan 20
Switch2(config-if)# overlay host-collect enable
Switch2(config-if)# exit
Switch2(config)# evpn
```

```
Switch2(config-evpn)# vni 10
Switch2(config-evi)# rd 10:2
Switch2(config-evi)# route-target both 10:10
Switch2(config-evi)# route-target export 1000:1000
Switch2(config-evi)# vni 20
Switch2(config-evi)# rd 20:2
Switch2(config-evi)# route-target both 20:20
Switch2(config-evi)# route-target export 1000:1000
Switch2(config-evi)# exit
Switch2(config-evpn)# exit
Switch2(config)# ip vrf evpn
Switch2(config-vrf)# route-target import 1000:1000 evpn
Switch2(config-vrf)# exit
Switch2(config)# interface nve 0
Switch2(config-if)# source loopback 1
Switch2(config-if)# member vni 10
Switch2(config-if)# member vni 20
Switch2(config-if)# member vni 1000 associate-vrf
Switch2(config-if)# exit
Switch2(config)# end
```

Switch5:

```
Switch5# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch5(config)# vlan database
Switch5(config-vlan)# vlan 10 overlay enable
Switch5(config-vlan)# qui
Switch5(config)# overlay
Switch5(config-overlay)# vlan 10 vni 10
Switch5(config-overlay)# quit
Switch5(config)# interface vlan 10
Switch5(config-if)# overlay host-collect enable
Switch5(config-if)# exit
Switch5(config)# evpn
Switch5(config-evpn)# vni 10
Switch5(config-evi)# rd 10:5
Switch5(config-evi)# route-target both 10:10
Switch5(config-evi)# route-target export 1000:1000
Switch5(config-evi)# exit
Switch5(config-evpn)# exit
Switch5(config)# ip vrf evpn
Switch5(config-vrf)# route-target import 1000:1000 evpn
Switch5(config-vrf)# exit
Switch5(config)# interface nve 0
sSwitch5(config-if)# source loopback 0
Switch5(config-if)# member vni 10
Switch5(config-if)# member vni 1000 associate-vrf
Switch5(config-if)# end
```

Step 6 Validation

Check the IP route

Switch1:

```
Switch1# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
```

```
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.253/32 is in local loopback, vlan10
C       192.168.10.1/32 is directly connected, vlan10
B       192.168.10.20/32 [200/0] is in overlay remote vxlan vtep:10.11.11.11->10.5.5.5, vni:1000
C       192.168.20.0/24 is directly connected, vlan20
C       192.168.20.253/32 is in local loopback, vlan20
C       192.168.20.1/32 is directly connected, vlan20
```

Switch2:

```
Switch2# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.254/32 is in local loopback, vlan10
C       192.168.10.1/32 is directly connected, vlan10
B       192.168.10.20/32 [200/0] is in overlay remote vxlan vtep:10.11.11.11->10.5.5.5, vni:1000
C       192.168.20.0/24 is directly connected, vlan20
C       192.168.20.254/32 is in local loopback, vlan20
C       192.168.20.1/32 is directly connected, vlan20
```

Switch2:

```
Switch5# show ip route vrf evpn
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
    O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    Dc - DHCP Client
    [*] - [AD/Metric]
    * - candidate default

C       192.168.10.0/24 is directly connected, vlan10
C       192.168.10.254/32 is in local loopback, vlan10
C       192.168.10.1/32 is directly connected, vlan10
B       192.168.10.10/32 [200/0] is in overlay remote vxlan vtep:10.5.5.5->10.11.11.11, vni:1000
B       192.168.20.10/32 [200/0] is in overlay remote vxlan vtep:10.5.5.5->10.11.11.11, vni:1000
```

Check the connectivity between the hosts

```
PING 192.168.10.10 from 192.168.10.20: 56 data bytes
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.445 ms
```

```
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 0.445/0.445/0.445/0.000 ms

PING 192.168.20.10 from 192.168.10.20: 56 data bytes
64 bytes from 192.168.20.10: icmp_seq=1 ttl=62 time=3.744 ms
--- ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
rtt min/avg/max/stddev = 3.744/3.744/3.744/0.000 ms
```

18.5.4 Deployment Suggestion

concentrated gateway Solution

Overview of concentrated gateway

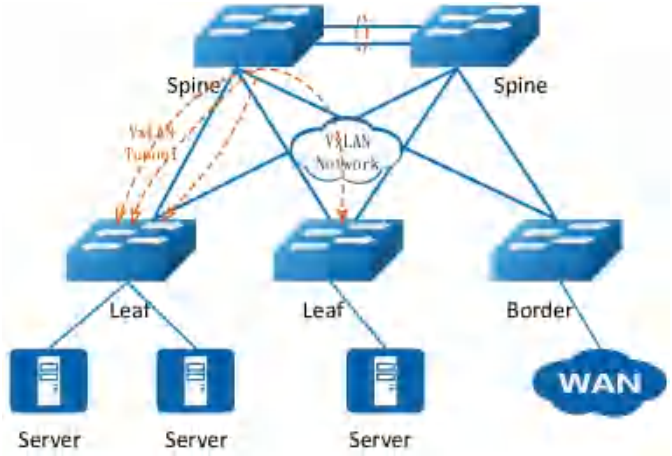


Figure 18-44 Concentrated gateway

Characteristic of concentrated gateway:

- The gateway is deployed in Spine
- VTEP is deployed in Leaf
- EVPN is used to establish tunnels between devices and synchronize routes between hosts
- There are one or more devices that are dedicated to connecting to the external network. We call it border-leaf. Spine and Border communicate with each other through default routes
- Three layers of traffic under the same Leaf need to be detoured to the Spine
- The Spine needs to store the MAC and ARP of the entire network
- Suitable for small - and medium-sized networking

Advantage:

- The access device (Leaf) does not need to support large-capacity ARP entries.
- The access device (Leaf) does not need to support the Layer-3 forwarding entry search function based on the VXLAN network, which reduces the cost of access devices.

Shortcoming:

- The core device(Spine) needs to reserve ARP entries for the entire network; The network scale is limited.
- Layer 3 traffic from the same Leaf needs to be routed to the Spine and then forwarded, which is not the optimal path.

Layer 2 access solution of the concentrated gateway

There are two modes for Layer 2 packet forwarding in a VXLAN network

- Unicast: the packets with MAC address table in the VXLAN network can be forwarding in unicast mode.
- BUM: Non-unicast packets are forwarded by BUM(Broadcast, Unknown unicast, or Multicast) mode.

To implement unicast or BUM, the control plane must first establish an entry, which is described in detail below

Control plane Establishes entries required for Layer 2 BUM: On a VXLAN network, BUM packets are forwarded through the replication list at the end of the VXLAN tunnel. The EVPN Type 3 route is used to transfer L2 VNI and L2 VTEP, and create a header replication list. The process is as follows.

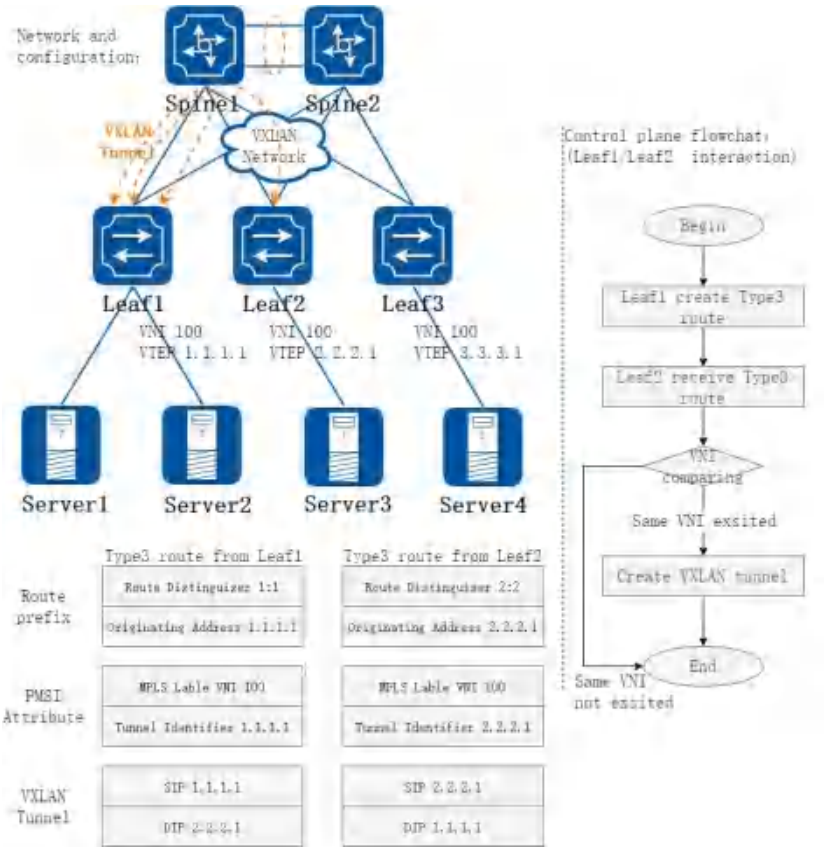


Figure 18-45 Layer 2 packet control plane Creating tunneling

Note:

- Leaf1 Generates Type3 routes and sends them to remote ends
- After receiving a remote VTEP neighbor Type 3 route, Leaf2 finds that the L2 VNI is the same as the local one, establishes a VXLAN tunnel to the remote end, and creates a replication list to the remote VXLAN tunnel header.
- The above process is repeated between Leaf1 and Leaf3, and between Leaf2 and Leaf3, to establish tunnels and header replication lists.

The control plane establishes the required entries for Layer 2 unicast: After a Leaf learns the MAC address of a Server, the MAC address table is synchronized to all Leaves through EVPN to implement Layer 2 informed unicast forwarding. Synchronize MAC addresses based on Type2 routes.

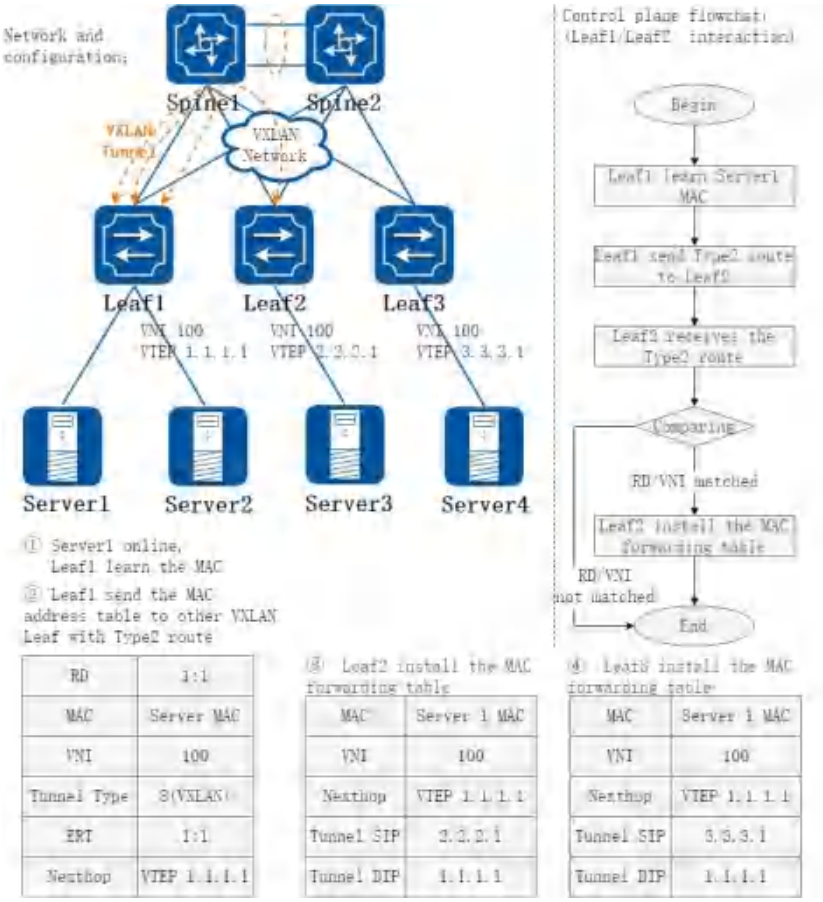


Figure 18-46 Schematic diagram of learning MAC addresses on the Layer 2 packet control plane

Note:

- Leaf1 collects MAC and/or information about local host Server1, encapsulates an EVPN Type2 route, and sends it to Leaf2.
- After receiving Type2 routes, Leaf2 compares the local Route-target with the L2-VNI corresponding to the EVPN instance, and adds the received MAC addresses to the local MAC forwarding table.
- Server1 MAC is also learned by Leaf3.
- The MAC addresses of Server2, Server3, and Server4 are also learned by repeating the above steps and synchronized to all Leafs.

Forwarding plane Layer 2 access flow

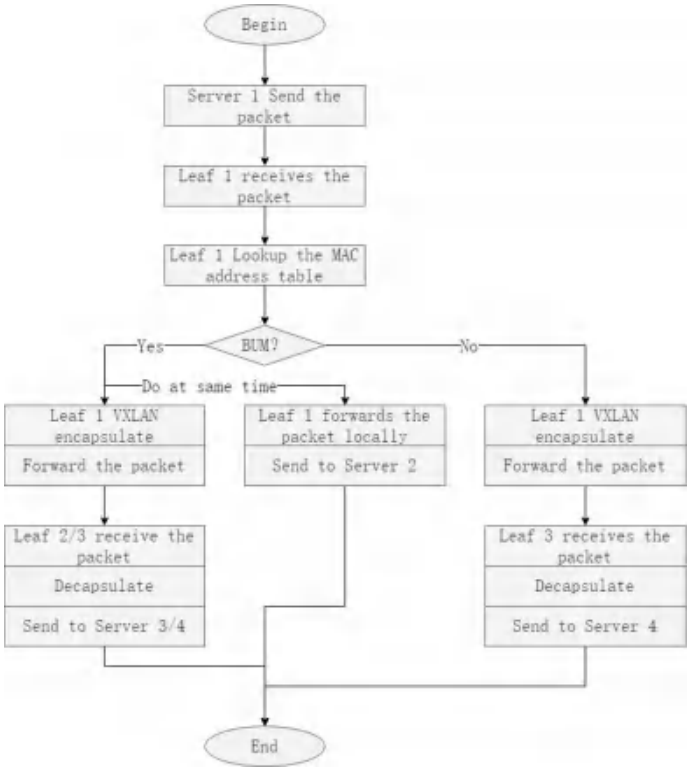


Figure 18-47 Forwarding plane process

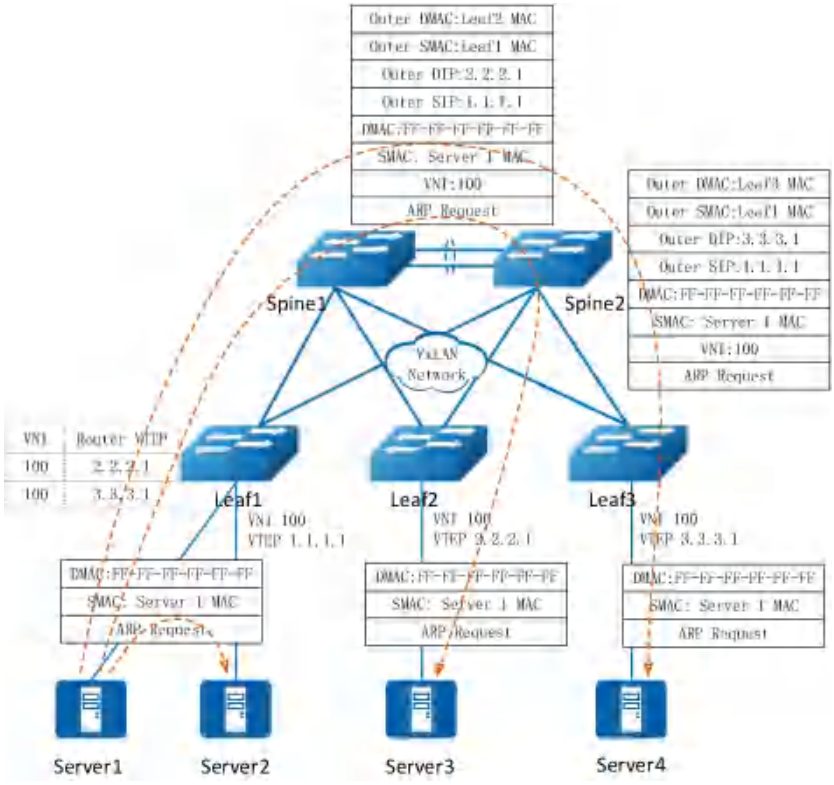


Figure 18-48 BUM

BUM Description:

- Leaf1 receives a packet from host Server1, obtains the Layer 2 broadcast domain based on the port and VLAN information connected to the packet, and determines that the destination MAC address of the packet is BUM MAC.
- A VTEP on Leaf1 obtains the header replication tunnel list of the corresponding VNI based on the Layer 2 broadcast domain, replicates packets based on the obtained tunnel list, encapsulates packets with a VXLAN, and forwards packets through the outbound port.
- After receiving a VXLAN packet, a VTEP on Leaf2/Leaf3 checks the validity of the packet based on the destination port number, source/destination IP address, and -VNI. Obtain the corresponding layer 2 broadcast domain based on the VNI, and then decapsulate the VXLAN to obtain the inner layer 2 packets.
- If the destination mac address of a layer 2 packet matches the mac address table, the packet is encapsulated and forwarded according to the port found in the mac address table. Otherwise, the packet is broadcast to the user port of the Leaf.
- In this networking scenario, Layer 2 packets are directly forwarded between Server1 and Server2 without VXLAN encapsulation.

Note: BUM packets of the VXLAN can only be broadcast to the user side, but not to the VXLAN side. Otherwise, a loop may occur.

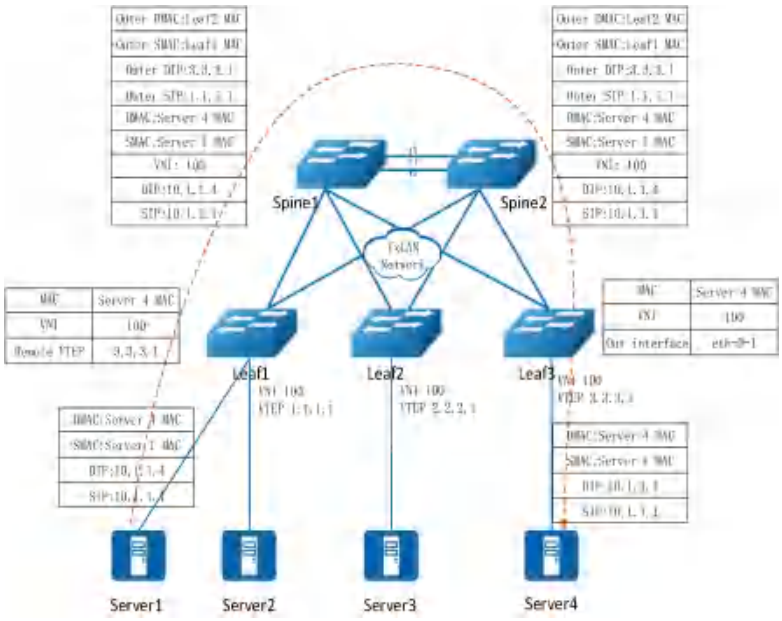


Figure 18-49 Unicast

Unicast description:

- Leaf1 receives a packet from a host Server, obtains a Layer 2 broadcast domain based on the port and VLAN information connected to the packet, and determines that the destination MAC of the packet is a informed unicast MAC.
- A VTEP on Leaf1 encapsulates and forwards packets by VXLAN based on the obtained VNI and remote VTEP addresses. Encapsulated VXLAN packets are forwarded through the IP network between Leaf1 and Leaf3.
- After receiving a VXLAN packet, a VTEP on Leaf3 checks the validity of the packet based on the destination port number, source/destination IP address, and VNI. Obtain the corresponding layer 2 broadcast domain based on the VNI, and then decapsulate the VXLAN to obtain the inner layer 2 packets.
- Leaf3 Searches the destination MAC addresses of inner - and layer-2 packets, obtains the outbound interfaces and encapsulation information in the MAC address table, adds VLAN tags to the packets, and forwards the packets to corresponding terminal servers.

Layer 3 access solution of the concentrated gateway

In a concentrated gateway environment, Layer-3 forwarding and routing are processed on the concentrated gateway node. The Spine device needs to store all Layer-3 entries, including ARP entries and network segment routes. In the following topology, spine1 and spine2 form MLAG to ensure reliability.

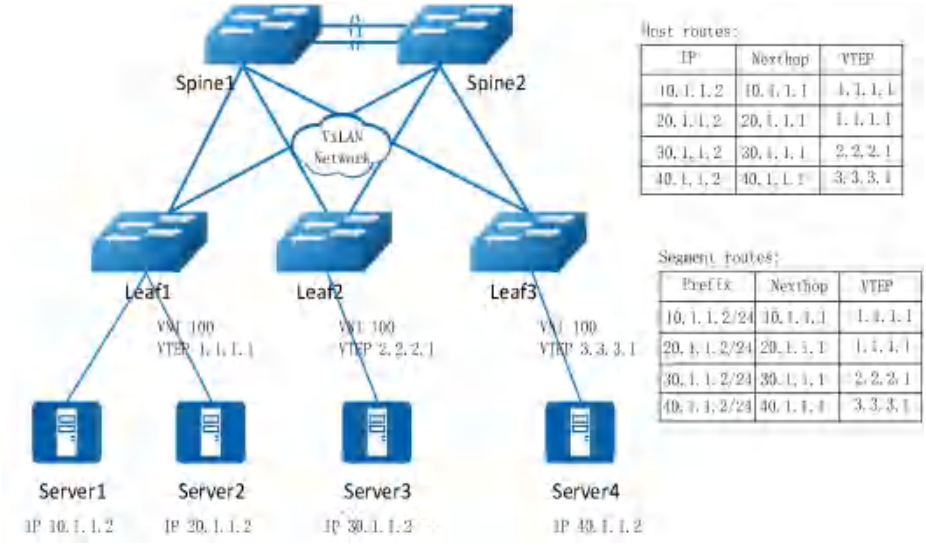


Figure 18-50 Layer 3 networking of the concentrated gateway

To implement Layer 3 forwarding, the control plane needs to learn host routing (ARP) entries and network segment routing entries. The learning process is described below.

The control plane learns ARP entries:

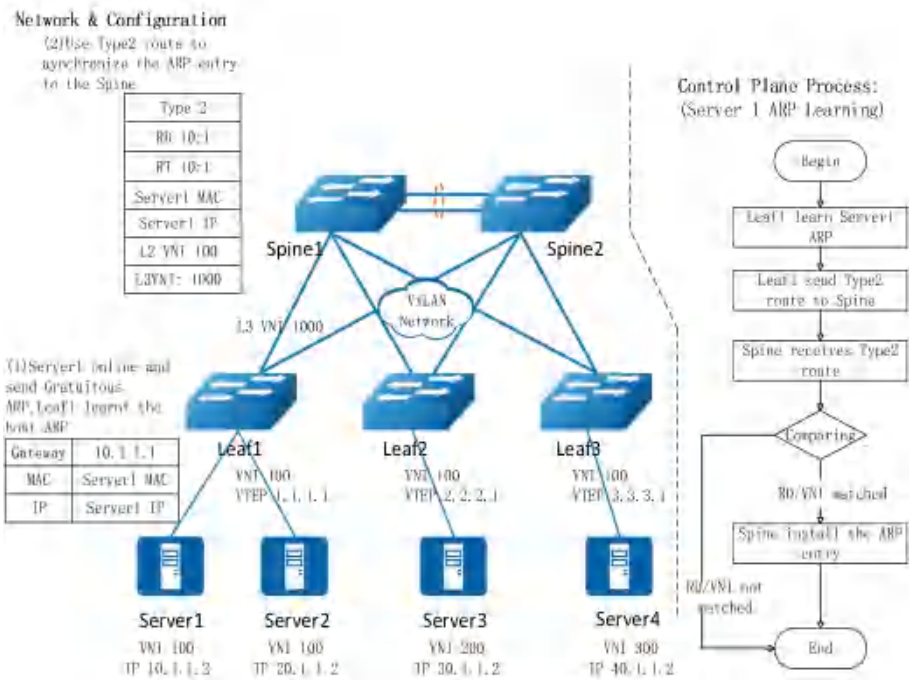


Figure 18-51 concentrated gateway ARP learning

Control plane learning network segment routing:

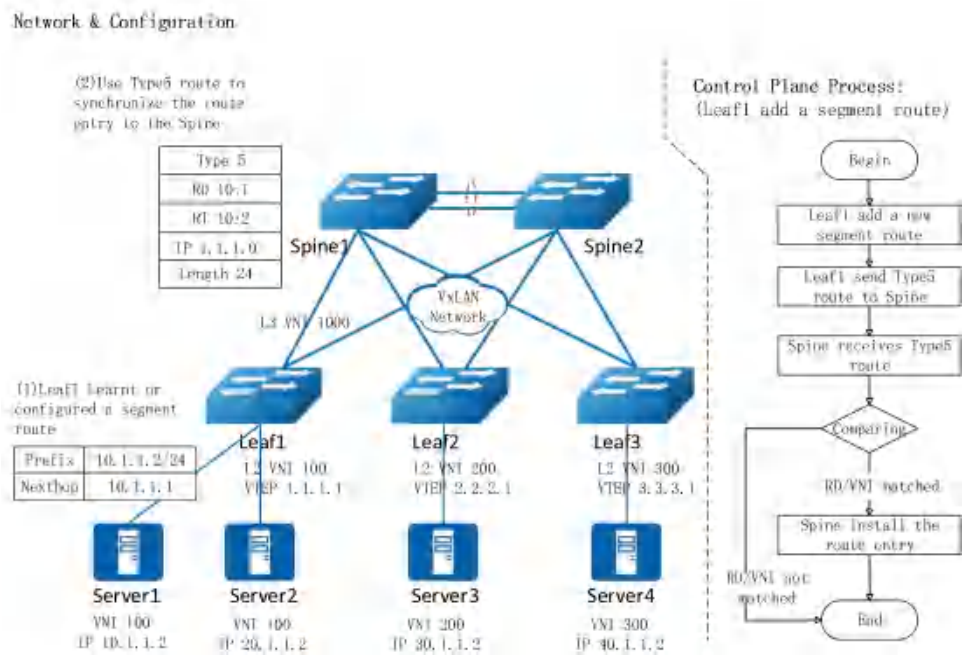


Figure 18-52 Route learning in the concentrated gateway segment

Note:

- ARP is synchronized to the spine using Type2 routing.
- Network segment routes are synchronized to the spine using Type5 routing.
- In this networking scenario, repeat the steps in the preceding figure to synchronize all serverARP and network segment routes on all Leaves.

Forwarding plane three-layer communication flow

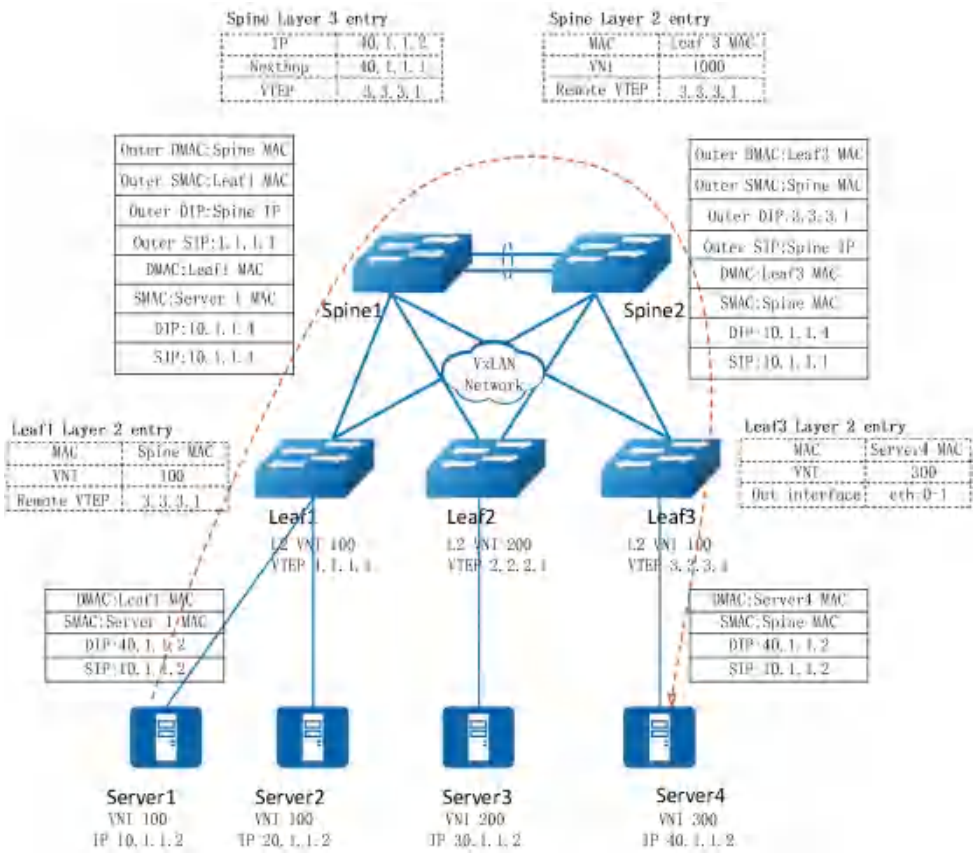


Figure 18-53 Topology of Layer 3 packet forwarding on the concentrated gateway

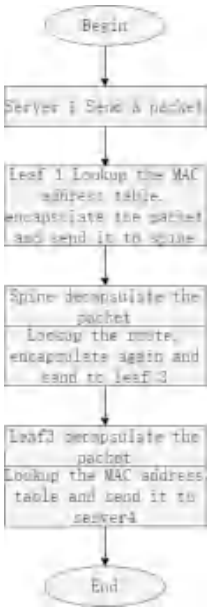


Figure 18-54 Process of Layer 3 packet forwarding on the concentrated gateway

Note:

- After receiving the VXLAN packet, the gateway decapsulates the packet, obtains the inner layer 2 packet, determines that the destination mac address of the layer 2 packet is the local mac address, and enters the Layer 3 forwarding process. Check the routing table based on the destination IP address of the inner packet and perform Layer 3 forwarding.

- The router obtains the next-hop IP address from the routing table and searches the ARP table according to the IP address. If ARP. If the outbound interface is a VXLAN tunnel, follow the ARP table's VXLAN information the packet is encapsulated and then sent out.

Distributed gateway Solution

Overview of Distributed gateway

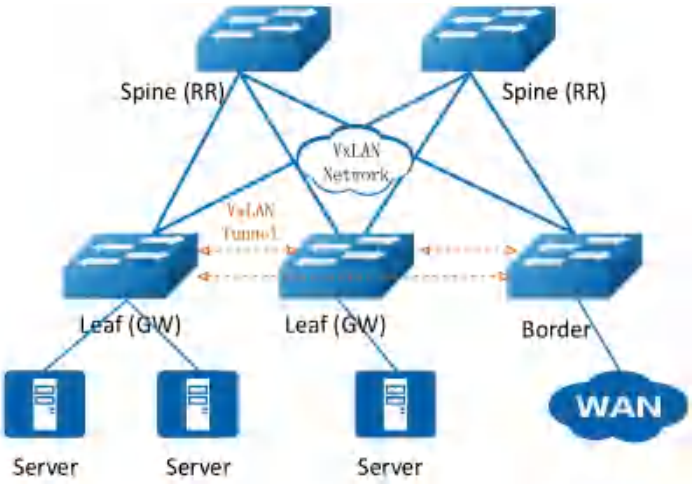


Figure 18-55 Distributed gateway

Characteristic of distributed gateway:

- The gateway is deployed in Leaf, and the gateway IP address and MAC address of the same VXLAN are the same
- Leaf and Spine communicate with each other through IP ECMP
- The Leaf node also supports VXLAN L2/L3 forwarding
- EVPN is used to establish tunnels between leafs and synchronize routes between hosts
- The Layer 3 traffic in the same Leaf does not need to be routed to the spine. The path of all traffic is optimal.
- The Leaf contains the MAC of all endpoints in the Layer 2 domain and the ARP of all endpoints in the Layer 3 domain (including local and remote endpoints) that need to communicate with each other
- Suitable for medium and large scale networking

Advantage:

- The network is simple, and the broadcast domain is restricted below the access layer
- By running routing protocols between Leaf and spine, a layer 3 reachable IP network of ECMP is constructed and the reliability is high
- The Layer-3 traffic in Leaf does not need to sned to spine, and all traffic paths are optimal

Shortcoming:

- Higher performance specifications for access devices (Leaf).

Layer 2 access solution of distributed gateway

The upper-layer access process of the ascending gateway is the same as that of the centralized gateway.

Layer 3 access solution of distributed gateway

The distributed gateway synchronizes Layer 3 routing information EVEN between leaves. The Spine acts as a route reflector (RR). To implement Layer 3 forwarding, each Leaf needs to learn host routing (ARP) entries and network segment routing entries. The learning process is described below.

The control plane learns ARP entries:

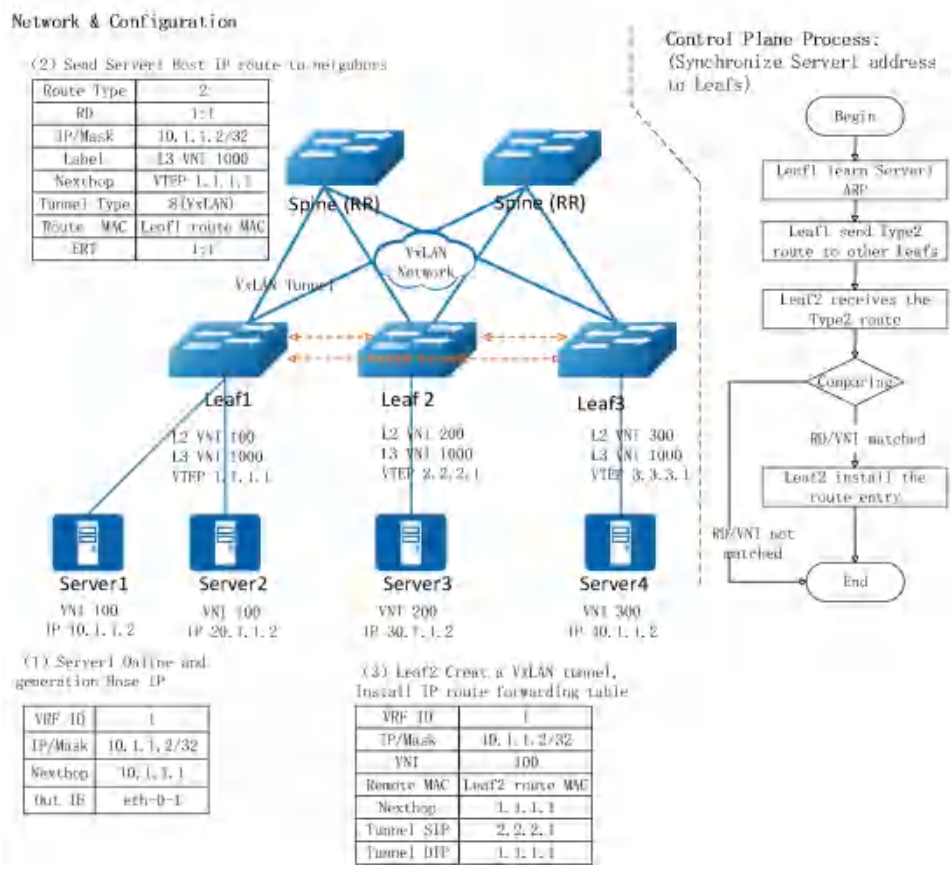


Figure 18-56 ARP learning of the distributed gateway

The control plane learns segment routing:

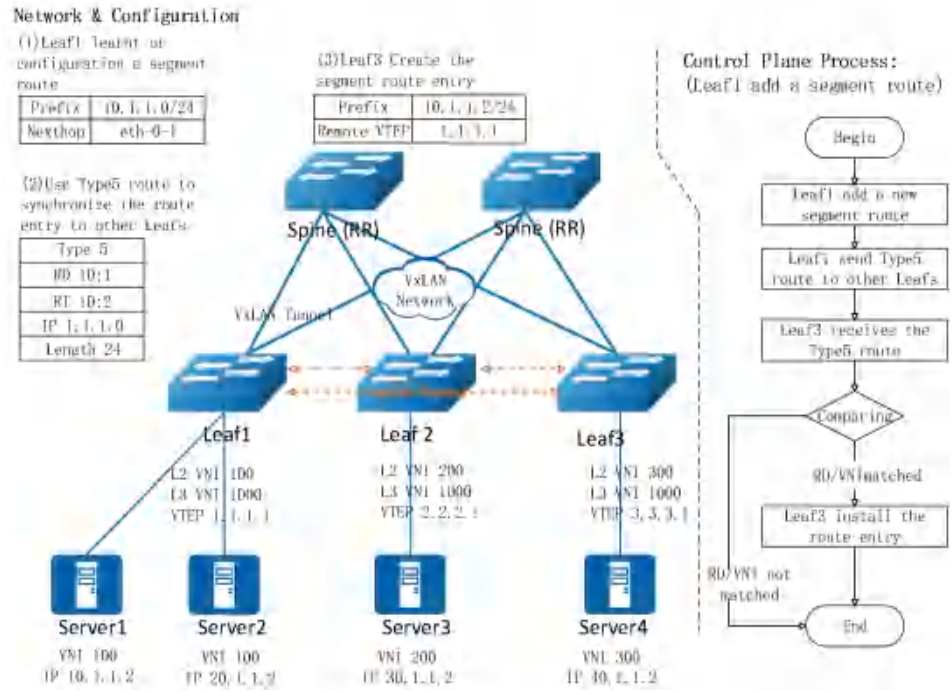


Figure 18-57 Route learning on the distributed gateway

Control plane process description:

- After the host online, Leaf learns the host's IP route and send it to other neighbors by EVPN. When sending the host's IP route, the packet should carry the L3-VNI of the corresponding tenant, change the next hop to the local VTEP-address
- After the remote Leaf learns the host's IP route, the Leaf will deliver it to the corresponding VPN instance, then trigger the creation of a dynamic L3 VXLAN tunnel based on the nexthop, and associate the route with the tunnel, then install the IP routing table.
- In the figure above, the ARP of each Server needs to be learned and synchronized to each Leaf of the entire VXLAN network.

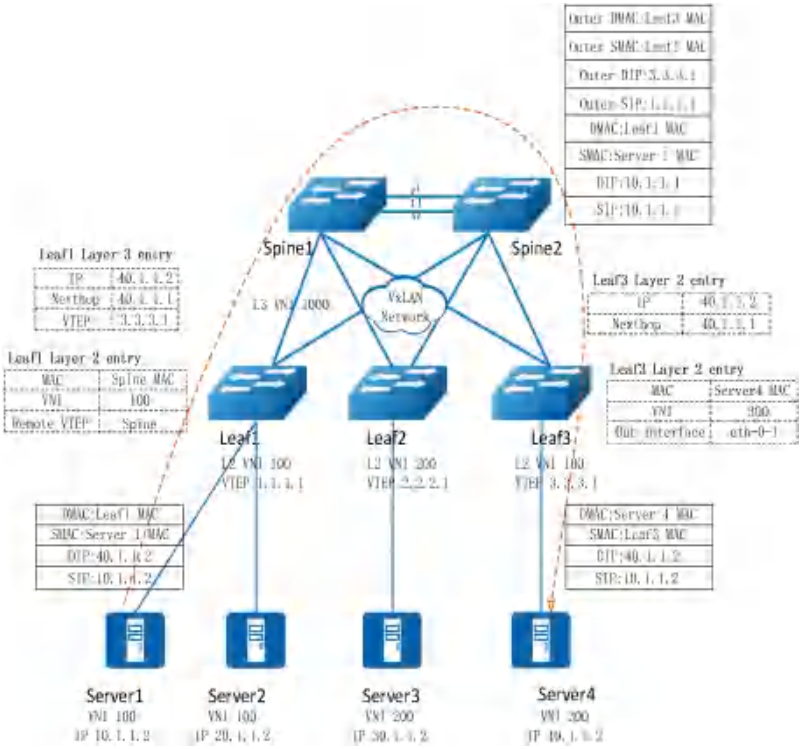


Figure 18-58 Layer 3 packet forwarding on the distributed gateway

The flow on the forwarding plane is as follows:

- If Leaf1 receives a packet from Server1 and the destination MAC of the packet is the gateway MAC, that is, the MAC of the Leaf its own, Leaf1 uses Layer 3 routing and forwarding to find the corresponding VPN instance through the inbound interface of the packet.
- In the found VPN instance, the router searches the routing table based on the destination IP address of the packet, and encapsulates the VXLAN packet with L3 VNI and RouterMac, then send it to Leaf3.
- After receiving VXLAN packets, Leaf3 checks whether the DMAC and notice that the DMAC is its own system Mac, therefore, Layer 3 forwarding is required. The router searches for the routing table in the VPN instance by VNI, finds that the nexthop of the route is the gateway interface, and sends the packet to Server4.

VM Migration Solution

Vm migration refers to the migration of a VM from one Leaf node to another Leaf node. In this case, keep the original MAC address and IP of the VM. Therefore, MAC addresses, ARP entries, and routing entries in the VXLAN must be updated in time.

After VMS are migrated, they usually send gratuitous ARP packets to refresh host routing and ARP information.

During the migration process, the EVPN control plane notifies the host of ARP, and the ARP must carry the migration attribute through the extended community attribute Mac mobility in BGP update message in order to advertise. Each migration requires the segNum in Mac mobility to increase with the step of 1.

The Mac mobility extended community packet format is as follows:

Type (0x6)	1 Byte
Sub-Type (0x0)	1 Byte
Flags	1 Byte
Reserved(0x0)	1 Byte
Sequence Number	4 Byte

Figure 18-59 Extended community attributes for Mac mobility

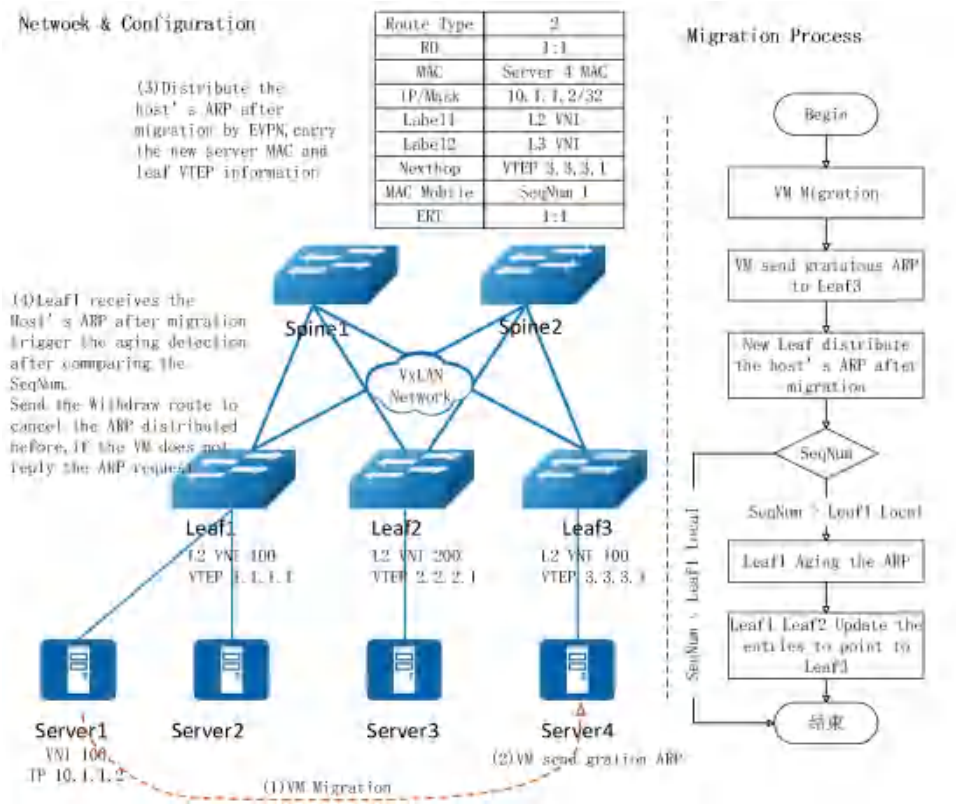


Figure 18-60 VM Migration

Note:

- Virtual machine 10.1.1.2 In the server1 location, the ARP and host routes of the virtual machine have been learned and synchronized to all leaves.
- Vm 10.1.1.2 is migrated from server1 to server4
- After a migrated VM is powered on, Leaf3 receives gratuitous ARP packets from the VM.
- After discovering that this ARP is sent by Leaf1, Leaf3 encapsulates the ARP update information when it refreshes local entries and sends it to other neighbors.
- After receiving packets from Leaf3, Leaf1 finds that the ARP is its own virtual machine and needs to compare the seqNum of Mac mobility. The update is considered valid only if seqNum is greater than the seqNum value in the last local message; otherwise, it is not processed.
- If the received SegNum value is greater than the last SeqNum value, Leaf1 needs to rapidly age the original ARP entries. ARP request packets are sent to detect VMS before migration. If the ARP request times out and is not returned, Leaf1 sends the withdraw route to announce all the previous ARP/host routes are invalid.
- After receiving withdraw messages from Leaf1, Leaf2 and Leaf3 delete old entries. Leaf1 and Leaf2 receive update messages from Leaf3 to add new local entries.

Dual host active Solution

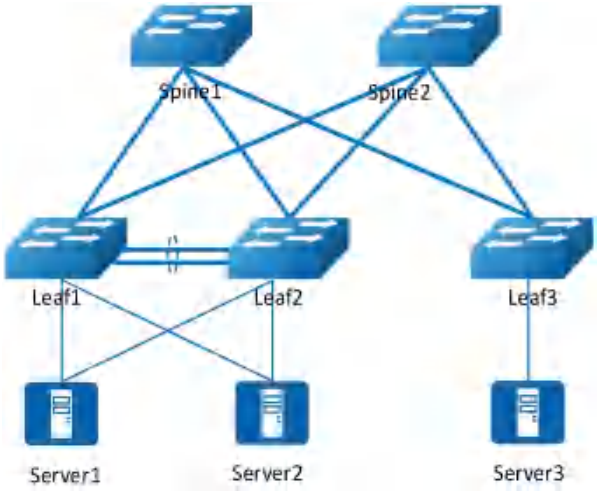


Figure 18-61 Dual host active

Host access reliability which means the Leaf can be connected to the host through MLAG, and the Leaf can connect to the host through cross-device LAG. Configure the same IP address as the VTEP to establish a tunnel with other devices. A link is deployed between the two devices to protect paths in case of a fault.

In normal cases, the device directly forwards the traffic received from a dual-homing interface to other servers or sends the traffic to other devices through an encapsulated tunnel. The device receives the tunnel encapsulated traffic, strips the encapsulated traffic and forwards it to the host.

If the dual-homing interface fails, the traffic received from the host or tunnel is forwarded to the another host through the interconnection link between the dual-homing devices.

Run BGP EVPN can advertise the same VTEP-IP and RouterMAC when the local host route is advertised externally.

19 Intelligent Lossless Network Configuration Guide

19.1 Configuring Prioprity-based Flow Control

19.1.1 Overview

Function Introduction

In a network path that normally consists of multiple hops between source and destination, lack of feedback between transmitters and receivers at each hop is one of the main causes of unreliability. Transmitters can send packets faster than receivers accept packets, and as the receivers run out of available buffer space to absorb incoming flows, they are forced to silently drop all traffic that exceeds their capacity. These semantics work fine at Layer 2, so long as upper-layer protocols handle drop-detection and retransmission logic.

For applications that cannot build reliability on upper layers, the addition of flow control functions at Layer 2 can offer a solution. Flow control enables feedback from a receiver to its sender to communicate buffer availability. Its first implementation in IEEE 802.3 Ethernet uses the IEEE 802.3x PAUSE control frames. IEEE 802.3x PAUSE is defined in Annex 31B of the IEEE 802.3 specification. Simply put, a receiver can generate a MAC control frame and send a PAUSE request to a sender when it predicts the potential for buffer overflow. Upon receiving a PAUSE frame, the sender responds by stopping transmission of any new packets until the receiver is ready to accept them again.

IEEE 802.3x PAUSE works as designed, but it suffers a basic disadvantage that limits its field of applicability: after a link is paused, a sender cannot generate any more packets. As obvious as that seems, the consequence is that the application of IEEE 802.3x PAUSE makes an Ethernet segment unsuitable for carrying multiple traffic flows that might require different quality of service (QoS). Thus, enabling IEEE 802.3x PAUSE for one application can affect the performance of other network applications. IEEE 802.1Qbb PFC extends the basic IEEE 802.3x PAUSE semantics to multiple CoSs, enabling applications that require flow control to coexist on the same wire with applications that perform better without it. PFC uses the IEEE 802.1p CoS values in the IEEE 802.1Q VLAN tag to differentiate up to eight CoSs that can be subject to flow control independently.

Principle Description

N/A

19.1.2 Configuration

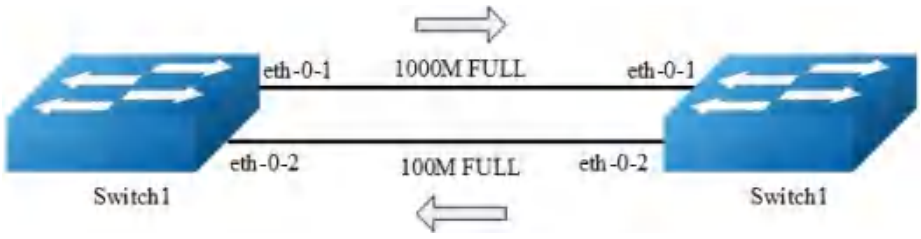


Figure 19-1 Priority-based Flow Control

In the following example, interface eth-0-1 of switch1 and switch2 are connected, interface eth-0-2 of switch1 and switch2 are connected, all interface enable PFC for priority 2/3/4.

The following configuration are same for switch1 and 2.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Enable lldp globally

```
Switch1(config)# lldp enable
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config-if)#lldp enable txrx
Switch(config-if)# lldp tlv 8021-org-specific dcbx
```

```
Switch(config-if)# priority-flow-control mode  on
Switch(config-if)# priority-flow-control enable priority 2 3 4
Switch(config-if)# exit

Switch(config)# interface eth-0-2
Switch(config-if)#lldp enable txrx
Switch(config-if)# lldp tlv 8021-org-specific dcbx
Switch(config-if)# priority-flow-control mode  auto
Switch(config-if)# priority-flow-control enable priority 2 3 4
Switch (config-if)# exit
```

step 4 Exit the configure mode

```
Switch(config)# end
```

step 5 Validation

Display the result on Switch1:

```
switch# show priority-flow-control
Port          PFC-enable        PFC-enable on priority
          admin   oper      admin    oper
-------------------------------------------------------------------
eth-0-1      on     on       234     234
eth-0-2      auto    off       234      off
eth-0-3      off     off       off      off
eth-0-4      off     off       off      off
eth-0-5      off     off       off      off
eth-0-6      off     off       off      off
eth-0-7      off     off       off      off
eth-0-8      off     off       off      off
eth-0-9      off     off       off      off
eth-0-10     off     off       off      off
eth-0-11     off     off       off      off
eth-0-12     off     off       off      off
eth-0-13     off     off       off      off
eth-0-14     off     off       off      off
eth-0-15     off     off       off      off
eth-0-16     off     off       off      off
eth-0-17     off     off       off      off
eth-0-18     off     off       off      off
eth-0-19     off     off       off      off
eth-0-20     off     off       off      off
eth-0-21     off     off       off      off
eth-0-22     off     off       off      off
eth-0-23     off     off       off      off
eth-0-24     off     off       off      off
```

Display the result on Switch2:

```
switch# show priority-flow-control
Port          PFC-enable        PFC-enable on priority
          admin   oper      admin    oper
-------------------------------------------------------------------
eth-0-1      on     on       234     234
eth-0-2      auto    on       234      off
eth-0-3      off     off       off      off
eth-0-4      off     off       off      off
```

```
eth-0-5     off    off      off    off
eth-0-6     off    off      off    off
eth-0-7     off    off      off    off
eth-0-8     off    off      off    off
eth-0-9     off    off      off    off
eth-0-10    off    off      off    off
eth-0-11    off    off      off    off
eth-0-12    off    off      off    off
eth-0-13    off    off      off    off
eth-0-14    off    off      off    off
eth-0-15    off    off      off    off
eth-0-16    off    off      off    off
eth-0-17    off    off      off    off
eth-0-18    off    off      off    off
eth-0-19    off    off      off    off
eth-0-20    off    off      off    off
eth-0-21    off    off      off    off
eth-0-22    off    off      off    off
eth-0-23    off    off      off    off
eth-0-24    off    off      off    off
```

19.2 PFC deadlock detection configuration

Step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configuring priority-flow-control deadlock precision (Optional)

The default precision is 10 milliseconds.

```
Switch(config)# priority-flow-control deadlock precision 20
```

step 3 Enable PFC deadlock detection

User can enable PFC deadlock detection for 1 or 2 priority on one interface.

User can specify the detect time multiple (with valid range 1-30). The precision value configured before multiples by this detect time multiple is the time period to determine the deadlock. During this time period, if the PAUSE OFF packets are received continuously, system will judge this priority of interface in PFC deadlock status. If the time multiple is not specified, system use the default value 15.

User can specify the recover time in the range of 0-3000 milliseconds. After this recover time, the deadlock is resolved and system start to detect again. The default recover time is 150 milliseconds.

The following example enables PFC deadlock detection on priority 5/6 of the interface eth-0-1. The detect multiple is 20, so according to the precision 20 milliseconds, the deadlock period should be 400 milliseconds. If the interface eth-0-1 received PAUSE OFF for priority 5 or 6 in continuously in 400 milliseconds, it will in deadlock status. The deadlock status will recover in 300 milliseconds. (But it may enter deadlock status again after 400 milliseconds)

In The deadlock status, system does NOT deal with the PAUSE OFF messages.

```
Switch(config)# interface eth-0-1
Switch(config-if)# priority-flow-control deadlock priority 5 6 detect 20 recover 300
```

step 4 Limit the PFC deadlock detection function(Optional)

User can enable the deadlock limit feature. It can enable for 1 or 2 priority on one interface.

User can specify the deadlock times and detect period. If the deadlock times achieve the threshold in the specified period, system will disable PFC for this priority of the interface, and wait for the administrator to deal with the network abnormal condition.

The default threshold is 10 times in 1 minute.

The following example enables PFC deadlock limit on priority 5/6 of interface eth-0-1, if deadlock 100 times in 5 minutes, system should disable PFC for this priority.

```
Switch(config)# interface eth-0-1
Switch(config-if)# priority-flow-control deadlock limit priority 5 6 frequency 100 period 5
```

19.3 Configuring EFD

19.3.1 Overview

Function Introduction

Elephant Flow Detect (EFD). According to the academic institutions of the actual network of the study found that more than 80% of the bandwidth is occupied by elephant flow, the bandwidth and transmission cache of these flow is large, but not sensitive to delay, which is sensitive to delay The flow caused a great impact. If elephant flow is recognized and some forwarding policies are implemented (such as reducing the forwarding priority of elephant flow appropriately, let mice flow be forwarded first), it can improve the transmission efficiency of network.

EFD function can be used to detect some abnormal traffic in the network (such as large bandwidth flow). After detecting, you can encapsulate the characteristics in the protocol packets and sent it to the specified server for further analysis.

Principle Description

terminology：

- EFD：Elephant Flow Detect

19.3.2 Configuration

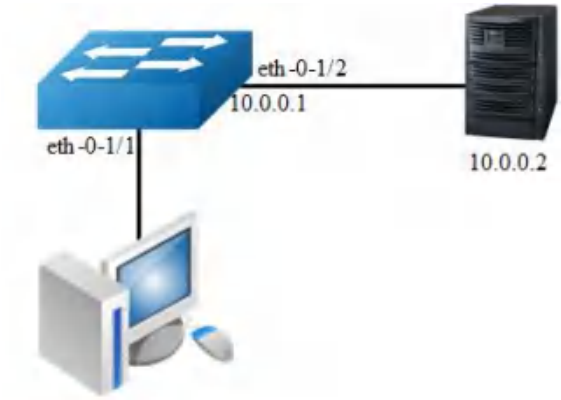


Figure 19-2 EFD

In the following example, it specifies the characteristics field and threshold of the traffic. When the flow rate exceed the specified threshold, the characteristics of the packets will be encapsulated into the user-defined UDP packets and sent to the server.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 1 Set the parameters for EFC

Specify ipda to calculate packet's hash value

```
Switch(config)# hash-value global
Switch(config-hash-value-global)# efd select ipda
```

Configure the speed threshold of EFD. The flows which has the rate large than 1000Mbps will be marked as Elephant Flow. The default value is 50Mbps.

```
Switch(config)# efd detect speed 1000
```

Enable EFD notify feature, and specify the ipda and UDP port of notification packet

```
Switch(config)# efd notify enable 10.0.0.2 20007
```

step 3 Enter the interface configure mode and set the attributes of the interface

```
Switch(config)# interface eth-0-1/1
Switch(config-if)# efd enable
Switch(config-if)# exit

Switch(config)# int eth-0-1/2
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.0.0.1/24
Switch(config-if)# exit
```

step 4 Create a static arp entry (Optional)

```
Switch(config)# arp 10.0.0.2 0.1.2
```

step 5 Exit the configure mode

```
Switch(config)# end
```

step 6 Validation

```
Switch# show efd configuration
Elephant flow detection configuration information:
-----------------------------------------------------
Detect rate          : 1000 Mbps
Detect granularity    : 16B
Detect time interval   : 1000 ms
EFD aging time       : 120 ms ~ 150 ms
EFD detect packet type : All IP packets
EFD IPG              : disable
EFD redirect interface : N/A
EFD flow hash fields   : destination-ip
EFD enabled interface :
------------------------- -------------------------
eth-0-1/1
```

When the flow received from eth-0-1 exceed 1000Mb, we can find this flow has been learned as EFD flow via the CLI below:

```
Switch# show efd flow information decap
EFD flow issued at:07:29:40 UTC Mon Aug 01 2016
From:eth-0-1, FlowId: 1701
-------------------------------------------------------------
MACDA:0000.00aa.bbbb, MACSA:0000.00bb.bbbb
IPv4 Packet, IP Protocol is TCP(6)
IPDA:22.22.22.101, IPSA: 11.11.11.11
L4SourcePort:43690, L4DestinationPort:43741
-------------------------------------------------------------
00 00 00 aa bb bb 00 00 00 bb bb bb 08 00 45 00
00 32 00 00 40 00 c8 06 70 35 0b 0b 0b 0b 16 16
16 65 aa aa aa dd aa aa aa dd aa aa aa dd aa aa
aa dd aa aa aa dd aa aa aa dd aa aa aa dd aa aa
```

Server 10.0.0.2 Tcpdump result:

```
12:41:28.286993 92:fd:58:d7:8f:00 > 00:00:00:01:00:02, ethertype IPv4 (0x0800), length 60: IP (tos 0x0, ttl  64, id 0, offset 0, flags [DF], proto 17, length: 44) 10.0.0.1.49071 > 10.0.0.2.20007: [udp sum ok] UDP, length 16
    0x0000:  0000 0001 0002 92fd 58d7 8f00 0800 4500  ........X.....E.
    0x0010:  002c 0000 4000 4011 26bf 0a00 0001 0a00  .,..@.@.&.......
    0x0020:  0002 bfaf 4e27 0018 b05b 0000 0101 0000  ....N'...[......
    0x0030:  0008 0001 0004 1616 1665 0000            .........e..
```


EFD packet head description. The red part above is part of EFD packet information, specific analysis is as follows:

- 0000: reserved, no specific meaning. Part of EFD packet head.
- 01:EFD packt version number, only support 0x01. Part of EFD packet head.
- 01:EFD flow opcode, 0x01: This flow is first recognized as elephant flow. 0x02: This flow has been recognized as elephant flow before. Part of EFD packet head.
- 0000 0008: EFD packet data part length(include data part type). Part of EFD packet head.
- 0001: EFD packet data part type. 0x0001 means data part is IPDA.
- 0004: EFD packet data part length.
- 16161665:date part, means IPDA is 22.22.22.101

19.4 ConfiguringECN

19.4.1 Overview

Brief Introduction

1. ECN (Configuring Explicit Congestion Notification)

Congestion occurs when the bandwidth of the downstream link of the forwarding device is insufficient to carry the traffic of the upstream link.As the congestion continues, the queue cache of the forwarding device is gradually compressed until it is exhausted. In this case, packet loss occurs.To solve this problem, RFC3168 defines Explicit Congestion Notification (ECN) to inform traffic senders to slow down the sending speed and bring the problem under control before packet loss.

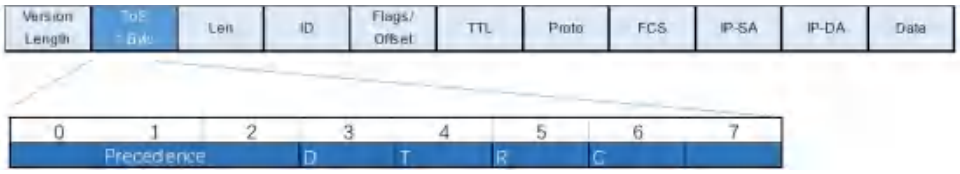


Figure 19-3 ToS Field

Eight bits are allocated in the ToS field of the IP packet header.

Bits 0 to 2 represent the eight priorities of IP packets. Priorities 7 and 6 are generally used to carry protocol packets.

Bits 0-5 are the DSCP of IP packets and define more service types.

Bits 6-7 are ECN fields.The values and meanings of ECN are as follows:

- If the ECN field is 00, the message does not support ECN.
- If the ECN field is 01 or 10, the packet supports ECN.
- If the ECN field is 11, it indicates that congestion occurs on the path.

Therefore, the ECN mechanism is as follows: When the intermediate switch is congested, the ECN flag is set to 11, and packets continue to be transmitted downstream.When a packet reaches the receiving end, the receiving end finds that the ECN field is 11 and considers that there is congestion in the path. Therefore, the receiving end sends CNP(Congestion Notification Packet) to the sending end to request the sending end to reduce the sending rate to prevent congestion

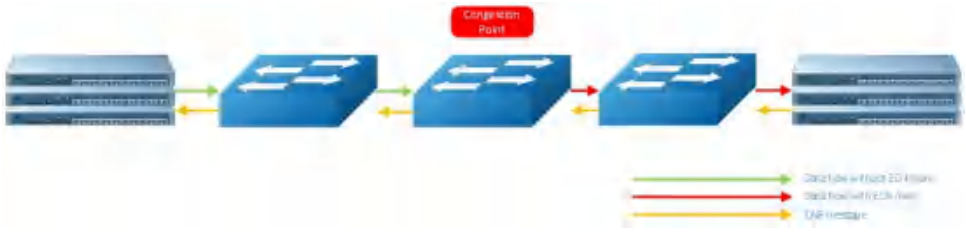


Figure 19-4 ECN mechanism

19.4.2 Configuration Examples

Configuration of ECN on priority queues

1. Configuration Steps

In the following example, ECN is configured on queue1,4,6 of Switch1. The threshould of queue4's ecn is 2100 and 5000. Other ecn threshould is configured as default value.

step 1 Enter the configure mode

```
Switch# configure terminal
```

step 2 Configure priority mapping

```
Switch(config)# class-map type qos pri-4
Switch(config-cmap-qos)# match dscp ef
Switch(config-cmap-qos)# exit
Switch(config)# class-map type qos pri-6
Switch(config-cmap-qos)# match dscp cs6
Switch(config-cmap-qos)# exit
Switch(config)# class-map type qos pri-1
Switch(config-cmap-qos)# match dscp cs1
Switch(config-cmap-qos)# exit
Switch(config)# policy-map type qos pri-pmap
Switch(config-pmap-qos)# class type qos pri-4
Switch(config-pmap-qos-c)# set traffic-class 4
Switch(config-pmap-qos-c)# exit
Switch(config-pmap-qos)# class type qos pri-6
Switch(config-pmap-qos-c)# set traffic-class 6
Switch(config-pmap-qos-c)# exit
Switch(config-pmap-qos)# class type qos pri-1
Switch(config-pmap-qos-c)# set traffic-class 1
Switch(config-pmap-qos-c)# interface eth-0-1
Switch(config-if)# service-policy type qos input pri-pmap
Switch(config-if)# end
```

step 3 Configure queue scheduling and ECN threshould

```
Switch# configure terminal
Switch(config)# class-map type traffic-class traffic-pri-4
Switch(config-cmap-tc)# match traffic-class 4
Switch(config-cmap-tc)# class-map type traffic-class traffic-pri-6
Switch(config-cmap-tc)# match traffic-class 6
Switch(config-cmap-tc)# class-map type traffic-class traffic-pri-1
Switch(config-cmap-tc)# match traffic-class 1
Switch(config-cmap-tc)# exit
Switch(config)# policy-map type traffic-class traffic-pmap
```

```
Switch(config-pmap-tc)# class type traffic-class traffic-pri-4
Switch(config-pmap-tc-c)# random-detect maximum-threshold 5000 minimum-threshold 2100 ecn
Switch(config-pmap-tc-c)# class type traffic-class traffic-pri-6
Switch(config-pmap-tc-c)# random-detect maximum-threshold 8192 minimum-threshold 64 ecn
Switch(config-pmap-tc-c)# class type traffic-class traffic-pri-1
Switch(config-pmap-tc-c)# random-detect maximum-threshold 8192 minimum-threshold 64 ecn
Switch(config-pmap-tc)# class type traffic-class class-default
Switch(config-pmap-tc-c)# queue-limit dynamic 5
Switch(config-pmap-tc-c)# exit
Switch(config-pmap-tc)# exit
```

step 4 Apply this ecn policy on an interface

```
Switch(config)# interface eth-0-1
Switch(config-if)# service-policy type traffic-class traffic-pmap
Switch(config-if)# exit
```

step 5 Exit the configure mode

```
Switch(config)# end
```

20 Stacking Configuration Guide

20.1 Overview

20.1.1 Background

At present, there are mainly two types of communication equipment in the network: box-type equipment and frame-type distributed equipment. Comparing them, we will find that:

- Box devices are low-cost, but lack high availability support and uninterrupted service protection, and cannot be used in important occasions (such as core layer, aggregation layer, production network, data center, etc.); in complex networking environments, the shortcomings of poor scalability of box devices are very obvious, and users have to maintain more network devices, and in order to add these devices, they have to modify the early networking structure;
- Frame-type distributed devices have the advantages of high availability, high performance, and high port density, so they are often used in some important occasions (such as core layer, aggregation layer, production network, data center, etc.). However, compared with box switches, they also have some disadvantages, such as high initial investment cost and high cost per port.

In view of these characteristics of box-type devices and frame-type distributed devices, a stacking technology that combines the advantages of both devices has emerged. Stacking is to connect multiple devices through stacking ports to form a virtual logical device, as shown in the following figure. Users manage this virtual device to manage all devices in the stack. This virtual device has the low cost advantage of box-type devices and the scalability and high reliability advantages of frame-type distributed devices.

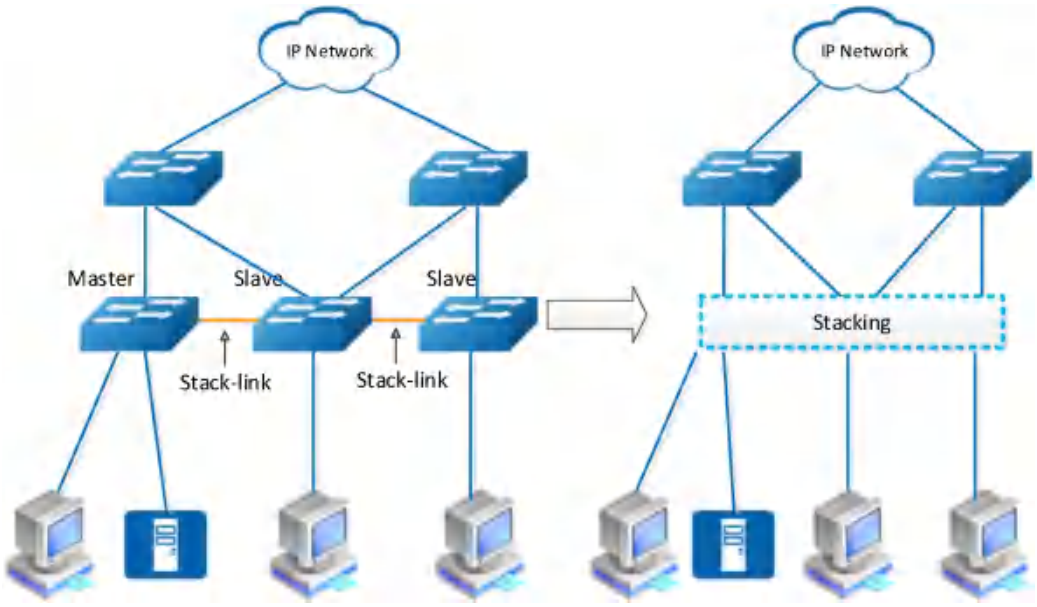


Figure 20-1 Stack Virtual Device

20.1.2 Technical Advantages

Stacking has the following main advantages:

- Simplified management. After the Stack is formed, users can log in to the Stack system by connecting to any port of the Master device. By configuring the Master device, users can manage the entire Stack and all member devices in the stack without having to physically connect to each member device to configure and manage them separately.
- Simplify network operation. The various control protocols running in the virtual devices formed by the Stack also run as a single device. For example, the routing protocol will be calculated as a single device. This saves the interaction of a large number of protocol messages between devices, simplifies network operation, and shortens the convergence time during network turbulence. This feature of Stack

technology is not available in common cluster technology, which can only unify device management, while the devices in the cluster still run as independent nodes in the network.

- Low cost. Stack technology virtualizes some lower-end devices into a relatively high-end device, so that it has the port density and bandwidth of high-end devices and the cost of low-end devices. It has a cost advantage over directly using high-end devices.
- Powerful network expansion capability. By adding member devices, the number of ports, bandwidth and processing power of the stacking system can be easily expanded.
- Protect user investment. With powerful expansion capability, when users upgrade their network, they do not need to replace the original equipment, but only need to add new equipment. This protects user investment very well.
- High reliability. The high reliability of the stack is reflected in many aspects, such as: the stacking physical ports between member devices support aggregation function, and the physical connection between the stacking system and the upper and lower devices also supports aggregation function, which improves the reliability of the stacking system through multi-link backup; the stacking system is composed of multiple member devices, the Master device is responsible for the operation, management and maintenance of the stack, and the Slave device can also handle business while serving as a backup. Once the Master device fails, the system will quickly and automatically elect a new Master to ensure that the business through the stack is not interrupted, thus realizing device-level 1:N backup. Stack is the best solution for network reliability assurance.
- High performance. Since the Stack device is composed of multiple stand-alone devices that support the Stack feature, the switching capacity and port number of the Stack device is the sum of the switching capacity and port number of all stand-alone devices in the Stack. Therefore, the Stack technology can easily expand the core switching capacity of the device and the density of user ports by several times through the stacking of multiple stand-alone devices, thereby greatly improving the performance of the device.
- Wide range of product support. As a universal virtualization technology, Stack technology realizes the integration of stacking of products of different forms, using the same technology to support stacking between the same type of devices.

20.2 Principle Description

20.2.1 Basic Concepts

Stacking technology refers to combining multiple physical switches and virtualizing them into a logical device for management and use. This achieves high network reliability and large data forwarding, while simplifying network management. In stacking, there are the following basic concepts.

Device ID

Device ID, also known as device number or slot number, is used to identify and manage member switches. The stack member ID is unique for all member switches in the stack.

Role

A single switch in a stack is called a member switch, which can be divided into the following roles according to its function:

- Master device: A member device that manages the entire stack. In a stack system, only one member device can be the master device at a time.
- Slave device: A member device that belongs to the Master device and operates as a member device of the Master device. Except for the Master device, all other devices in the stack are Slave devices. There may be multiple Slave devices in the stack.

Stacking Member

Used to identify and manage member switches. The stack member ID of all switches in the stack is unique.

Stacking priority

Stack priority is an attribute of a member switch, which is mainly used to determine the role of a member switch during the role election process. A larger priority value indicates a higher priority and a greater possibility of being elected as the Master switch.

Stacking physical member port

A stack physical member port, that is, a physical port configured in stack mode, is used to connect stack member switches.

Stacking port

A stack port is a logical port dedicated to stacking and needs to be bound to a stack physical member port. A stack port can be bound to one or more stack physical member ports to improve the bandwidth and reliability of the link. Each device supports two stack ports.

Stacking link

The link connecting the stack ports between stack member devices is used to transmit stack protocol messages and forward traffic across boards.

Stacking domain

Create a stack of domain IDs. A stacking system can be established between domains with the same ID, but not between domains with different IDs.

Stacking system MAC

The stacking system appears externally as a network device participating in network communication, possessing a unique MAC address, referred to as the stacking system MAC.

20.2.2 Stack Formation

Connection and Topology

For Stack to work properly, member devices need to be physically connected first. The physical port on the device used for stack connection is called stack physical port. Stack physical port can be configured as stack port, which is a logical interface. A stack port may correspond to one stack physical port, or it may be formed by aggregating multiple stack physical ports (called aggregate stack port) to achieve link backup effect.

The stacking physical ports support the use of common ports and do not require special stacking cables (stacking cards).

There are two connection topologies for stacking physical ports:

- Linear topology: Using business cables to connect the left port (right port) of one device to the right port (left port) of another device, and so on. The right port (left port) of the first device is not connected to the left port (right port) of the last device with business cables. This connection method is called linear topology, as shown on the left side of the figure below.
- Ring topology: Connecting the right port (left port) of the first device and the left port (right port) of the last device in a linear topology, this connection method is called ring topology, as shown on the right side of the figure below. Ring topology is more reliable than linear topology. When a link failure occurs in a ring topology, the stacked system can still maintain normal operation; when a link failure occurs in a linear topology, it can cause the stack to split.

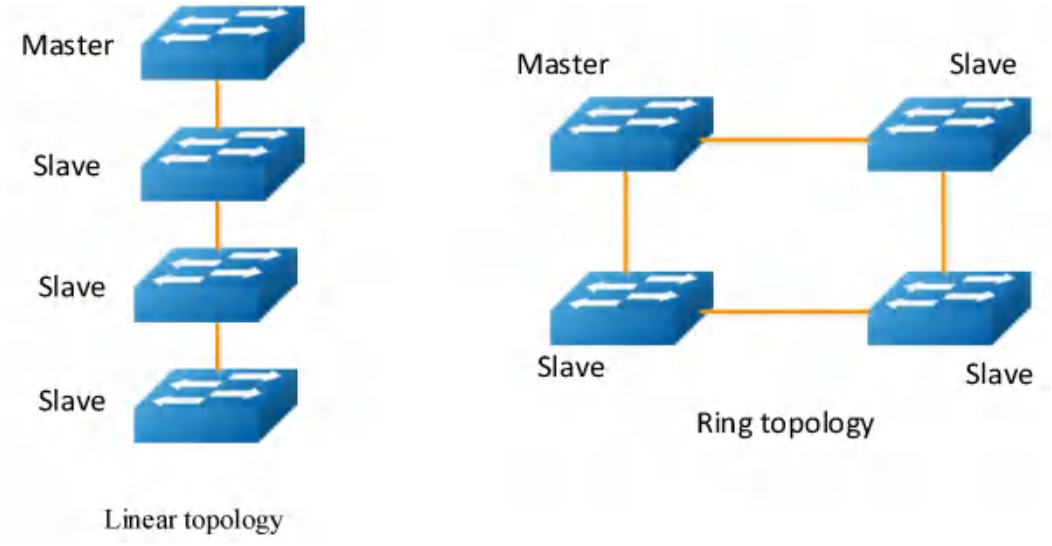


Figure 20-2 Schematic diagram of the physical connection of the stack

Topology Collection

Each device in the stack collects the topology of the entire stack by exchanging Hello messages with other member devices that are directly adjacent to it. The Hello message carries topology information, including stack port connection relationship, member device number, member device priority, member bridge MAC of member device, etc.

Each member device records its known topology information locally. Initially, the member device only records its own topology information. When the stack port status changes to up, the member device will periodically send the known topology information from the stack port in the up state. After receiving the topology information from the direct neighbor, the member device will update the locally recorded topology information. After a period of collection, all devices will collect complete topology information (called topology convergence). At this time, the role election phase will begin.

Role Election

A stack system consists of multiple stack member devices, each of which has a certain role, Master or Slave. The process of determining the role of a member device is called role election.

Role election occurs when the topology changes, such as stack establishment, new device joining, stack splitting, or two stacks merging. The role election rules are as follows (start judging from the first rule. If there are multiple optimal members participating in the election, continue judging the next rule until the only optimal member is found, and then stop the election. This optimal member is the Master device of the stack, and the other devices are all Slave devices):

- Members with larger priority values have priority;
- Members with smaller slot numbers take priority.

When the device starts, it first collects topology and participates in role election. Only after the process is successful can the stack system be formed and operate normally. After the role election is completed, the stack will enter the management and maintenance phase.

The virtual device formed by stacking box-type devices is equivalent to a frame-type distributed device. The Master in the stack is equivalent to the main control board of the virtual device, and the Slave device is equivalent to the interface board of the virtual device, as shown in figure below.

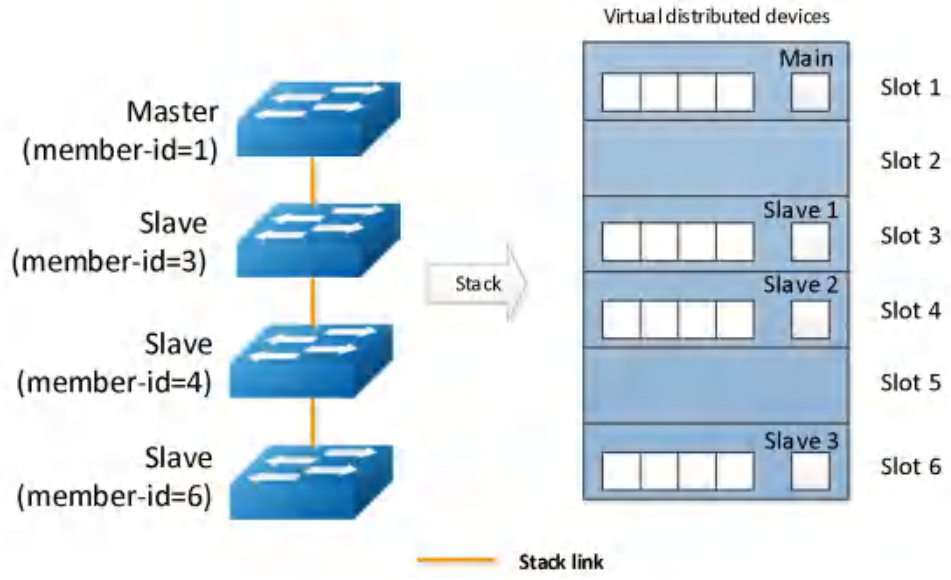


Figure 20-3 Box-type equipment stacking effect diagram

20.2.3 Stack Management

The virtual device formed by the Stack connection can be managed as a single entity. Users can log in to any member device in the Stack using the Console port or Telnet to manage and configure the entire Stack.

The Master device is the management center of the Stack system and is responsible for responding to user login requests. That is, no matter how the user logs in to the Stack or through which member device, the Master device is ultimately configured, and the Master device is responsible for sending the user's configuration to each Slave device. This method can keep the configuration of each device in the Stack highly unified.

The stacking system uses the slot ID as the member ID to identify and manage member devices. The member IDs of all devices in the stack are unique. The member ID is introduced into the port number. The first dimension of the interface number indicates the member ID of the device where the interface is located, which is convenient for users to configure and identify the interfaces on the member devices. The member ID is also introduced into the file system management. For example, the path slot2#flash:/test.cfg indicates that there is a file named test.cfg in the root directory of the Flash on the member device (member ID 2). The Master can directly access the Flash directories of all member devices.

20.2.4 Stack Maintenance

The main function of stack maintenance is to monitor the joining and leaving of member devices, collect new topologies at any time, and maintain the existing topology.

Member Device Joins

During stack maintenance, the topology collection continues. When a new member device is found, different processing is performed according to the status of the newly added device:

- If the newly added device does not form a stack itself (for example, the newly added device is configured with the stacking function, then powered off, and then connected to the existing stacking system, and then powered on and restarted), the device will be selected as a Slave.
- The device that is added has already formed a stack system (for example, the newly added device is configured with the stack function and then connected to the existing stack system), which is equivalent to merging two stacks (please note that it is generally not recommended to use this method to form a stack). In this case, the two stacks will conduct a stack election. All stack member devices on the losing side need to be restarted and then all join the winning side as slave devices.

If the member device joins successfully, it is equivalent to increasing the port resources of the interface board for the stacking system.

Possible reasons for member devices to join include: artificially adding members to the stack system; fault recovery. When a device failure or link failure is restored, the restored device will rejoin the stack.

Member Device Leaves

During stack maintenance, the following two methods are used to determine whether a member device has left:

- Under normal circumstances, directly adjacent member devices will exchange Hello messages every 1 second. If no Hello message is received from a direct neighbor for three consecutive cycles, it is considered that the member device has left the stack system, and the stack will isolate the member device from the topology.

If the Slave device leaves, the system is equivalent to losing the port resources of an interface board; if the Master device leaves, the Slave device of the stack system will take over all the functions of the original Master.

When a single device leaves the stack, it returns to an independent operating state. When multiple connected devices leave the stack, they form two independent stacks. This situation is called stack splitting.

Possible reasons for member device leaving include: artificially changing the topology and removing the member device; member device failure; connection failure.

Topology Update

A simple topology change refers to the change of the device topology from a Ring topology to a Linear topology, or from a Linear topology to a Ring topology. For example, for a ring-connected device, when a link fails, it may become a Linear topology; for another example, when adding a device, the original Ring topology needs to be changed to a Linear topology before the new device can be connected.

For simple topology changes, the composition of the Stack members and the Master will not change. The forwarding path will only be automatically changed when necessary, which will not affect the normal use of the device.

20.2.5 Stack splitting and Dual-Active Detection

Stack splitting

Stack splitting refers to the removal of some member switches from a stably operating stacking system, or the failure of a stacking connection channel, resulting in one stacking system being split into multiple stacking systems.

The original master switch will recalculate the stacking topology, delete the topology information of the removed member switch, and synchronize the new topology information to other member switches. Meanwhile, the removed member switch will automatically reset and undergo a new election to form a new stacking system. As shown in the figure below, after the stacking system is split, the original master switch Switch A deletes the topology information of Switch D and Switch E, and synchronizes the new topology information to Switch B and Switch C. After Switch D and Switch E are restarted, a new stacking system is established, with Switch D becoming the new master switch.

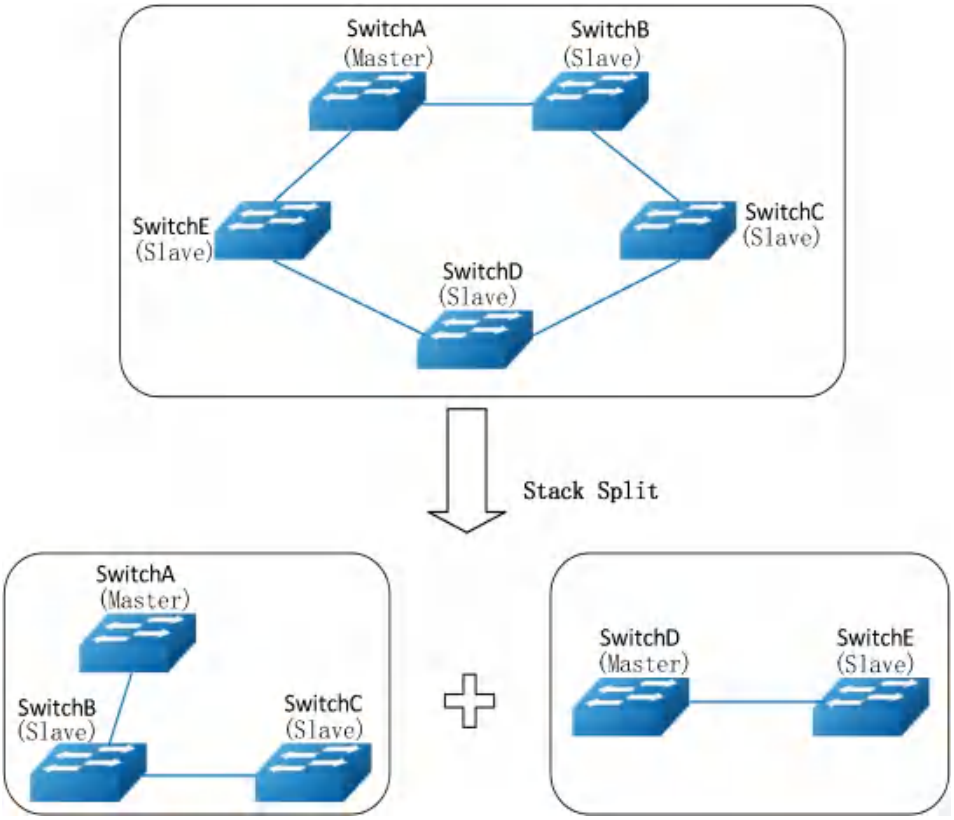


Figure 20-4 Schematic diagram of stacking splitting

Dual-Active Detection

Dual-Active Detect (DAD) is a protocol designed to detect and handle stack splitting, enabling the detection of stack splitting, conflict resolution, and fault recovery, thereby minimizing the impact of stack splitting on business operations. Currently, we support a direct-connection detection method for business ports, where dual-active detection is performed between stack member switches through links connected via business ports, as illustrated in fllow figure.

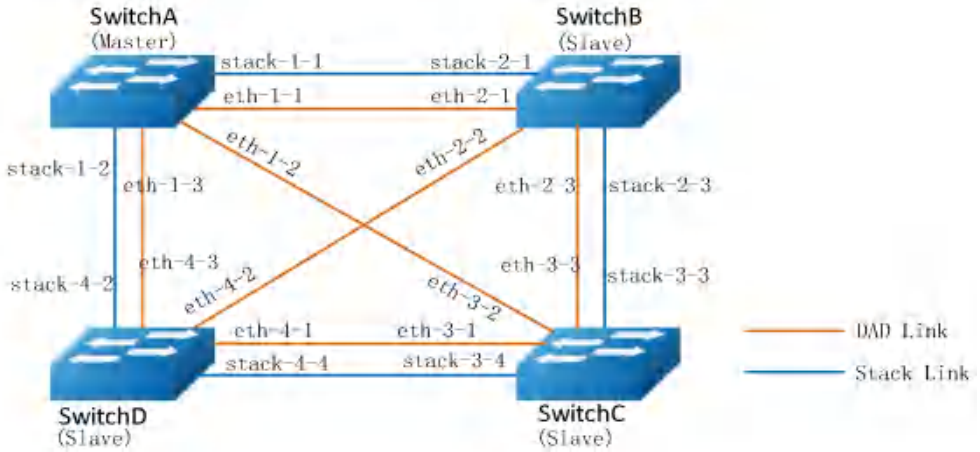


Figure 20-5 Schematic diagram of DAD business port direct connection detection

In the direct connection detection method for business ports, the DAD message utilizes BPDU messages. Therefore, the direct connection detection link can also be connected through intermediate devices, which need to be configured with the function of transparently transmitting BPDU messages. It is recommended to adopt the Full-mesh full connection method between stack member devices (i.e., every pair of member devices is connected).

After configuring dual-master detection, the master switch sends DAD competition messages on the DAD detection link. After stacking splitting, the split multi-part stacking system sends competition messages to each other, and compares the received competition message information with its own competition information. If its own competition wins, it does not perform any processing, maintains its working state, and forwards business messages normally; if its own competition fails, all business ports except the reserved port are Error-Down, entering a business disabled state and stopping forwarding business messages.

After the stack link failure is repaired, the split stack system is merged. Switches that were in a service disabled state will be restarted, and the service ports that were Error-Down will be restored to normal. The entire stack system will be restored.

The DAD competition rules are as follows (judge from the first rule in order until the optimal switch is found and then stop comparing):

1) In the comparison of stacking priorities, switches with higher stacking priorities have priority in competition and win.

2) When comparing equipment slot numbers, switches with lower slot numbers have priority in the competition and are more likely to win.

3) When comparing device MAC addresses, switches with smaller MAC addresses have priority in the competition and are more likely to win.

20.2.6 Principle of packet forwarding in stacking

Stack uses distributed elastic forwarding technology to implement Layer 2/3 forwarding of messages, maximizing the processing power of each member. Each member device in the stack system has complete Layer 2/3 forwarding capabilities. When it receives a Layer 2/3 message to be forwarded, it can query the Layer 2/3 forwarding table of the local device to obtain the outbound interface (and next hop) of the message, and then send the message out from the correct outbound interface. This outbound interface can be on the local device or on other member devices. Sending the message from the local device to another member device is a purely internal implementation, which is completely transparent and imperceptible to the outside world. That is, for a Layer 3 message, no matter how many member devices it passes through in the stack system, the number of hops is only increased by 1, which is manifested as passing through only one network device.

As shown in the figure below, the incoming and outgoing interfaces of the forwarded message are on the same member device. When Slave 1 receives the message, it searches the local forwarding table and finds that the outgoing interface is on the local device. Slave 1 then directly sends the message out through this outgoing interface.

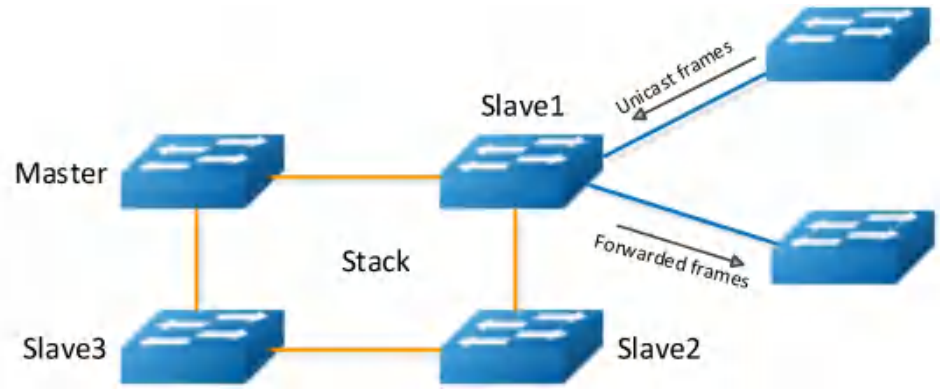


Figure 20-6 Forwarding unicast messages within a device

As shown in the figure below, the incoming and outgoing interfaces of the forwarded message are on different member devices. When Slave 1 receives the message, it searches the local forwarding table and finds that the outgoing interface is on the Master. Slave 1 forwards the message to the Master according to the optimal path, and the Master forwards the message to the end user through the outgoing interface.

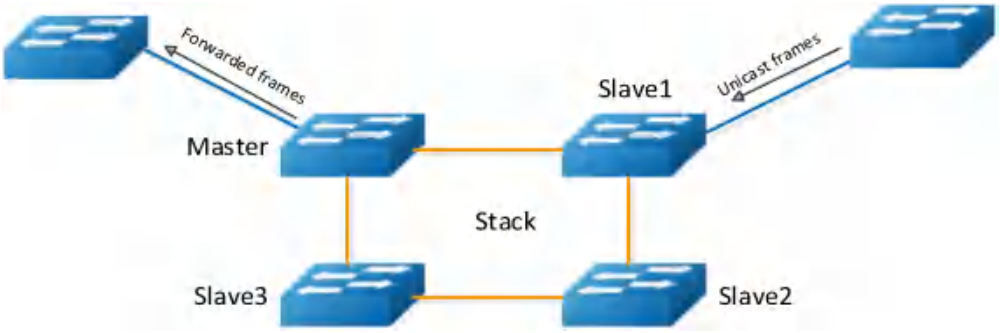


Figure 20-7 Cross-device unicast message forwarding

The follow figure shows the schematic diagram of the stack processing multicast messages. When Slave 1 receives a multicast message, it first searches the local multicast forwarding table. If there is a local receiver, it will directly send it to the local multicast group member. Slave 1 forwards the multicast message to the Master and Slave 2. The Master and Slave 2 also search the local multicast forwarding table first, and then forward it to the neighboring member device of the stack system on the other side. Due to the loop protection mechanism, Slave 3 will discard the multicast message from Slave 2. In this way, for multicast messages, each member will only copy the message as needed to ensure that only one copy of the message is transmitted between devices.

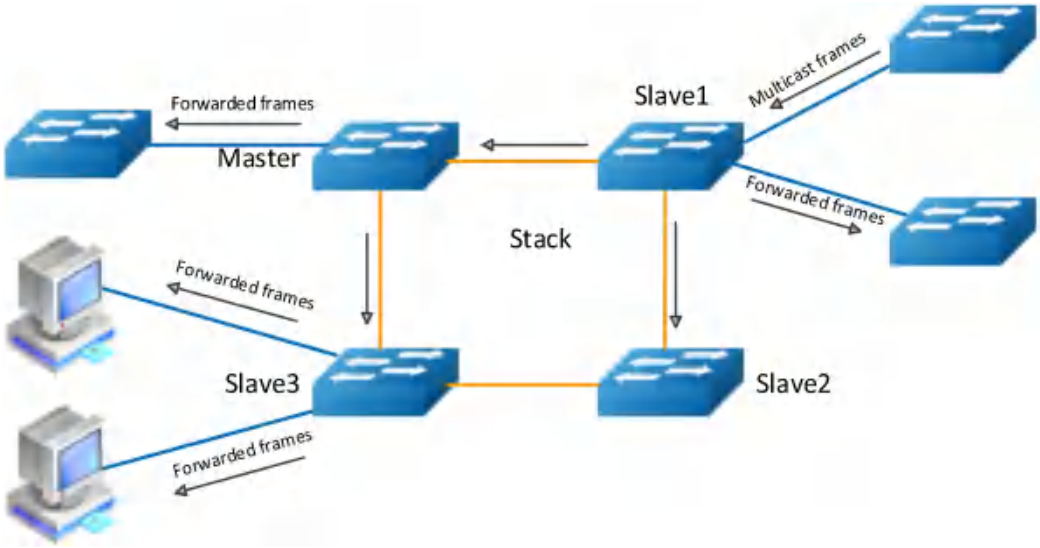


Figure 20-8 Multicast message forwarding

20.3 Configuration Notes

20.3.1 Stacking Device Requirements

Currently, the device series that support stacking include all device types on the Vastlakes platform, including: S5600 series, S5500 series, S5000 series, S4600 series, and DynamicDuo series switches.

The maximum number of stacking member switches supported in the same stacking domain is 8,with device models that can be the same or different.

20.3.2 Device Version and License Requirements

- The version of the newly added device is compatible with the version of the stack system.
- The newly added device has a license corresponding to the stacking function.

20.3.3 Stack Port and Cable Requirements

- Port types include: 10GE electrical port, 10GE/25GE/40GE/100GE optical port.
- Number of member ports: 1 to 8; it is recommended to add at least 2 member ports to each stack port to improve the bandwidth and reliability of the link.
- The slotid number of a member device is unique in a stack system, and the port number of a stack link is unique in a stack system and consistent at both ends of the link.
- Cable requirements: Stacking does not require dedicated stacking cables, and ordinary service cables can be used. Optical ports can use high-speed cables, AOC optical cables, or optical modules and optical fibers.

20.3.4 Stack Feature Dependencies and Limitations

Precautions for Setting Up a Stack

When setting up a stack, please note the following:

- When connecting stack cables, stack member devices must be directly connected without any relay devices.
- It is recommended to use at least 2 or more links to form a stack.
- After the stack is successfully established, you need to run the wirte memory slot all command to save the stack system configuration.
- When the stack system is working stably, do not shut down the stack port; otherwise, the stack will split and service traffic cannot be forwarded normally.

Stack Upgrade Precautions

When performing a stack upgrade, please note the following:

- During the stack upgrade process, do not perform hardware operations such as removing or inserting optical modules or powering on or off devices.
- During the stack upgrade process, please keep the network stable and do not perform other service configurations on the upgraded network.
- Supports upgrading all switches in the stacking system at once, and then restarting all switches in the stacking to complete the image upgrade. The command example is as follows:

➢ copy mgmt-if tftp://192.168.1.100/hos-vastlakes-v3.0.21.bin slotall#flash:/boot/hos-vastlakes-v3.0.21.bin

➢ boot system flash:/boot/hos-vastlakes-v3.0.21.bin slot all

➢ reload slot all

Stacking Feature Support

The software features currently supported by the stacking environment are as follows:

Basic configuration:

- System Administration
- User Management
- FTP
- FTP Server
- TFTP
- Telnet
- SSH
- HTTP
- WEB
- Time and Time Zones

Ethernet:

- L2 and L3 Interface
- Errdisable
- MAC Address Table
- Link Aggregation
- VLAN
- VLAN Classification
- VLAN Mapping
- QinQ
- STP/RSTP/MSTP
- Storm Control
- Flow Control
- L2 Protocol Tunnel
- Loopback Detection
- Hash Load Balancing

IP business:

- ARP
- ARP Proxy
- DHCP Server
- DHCP Client
- DHCP Relay

IP routing:

- static routing
- RIP
- OSPF
- ISIS
- BGP
- Prefix-List
- Route Map
- PBR

IP multicast:

- Static multicast
- MVR
- IGMP Snooping
- IGMP
- PIM-SM

Traffic management:

- QoS

Safety:

- VLAN Security
- ACL
- 802.1X
- Dynamic ARP Inspection
- DHCP Snooping
- IP Source Guard
- AAA
- Port Security
- Port Isolate
- Port Block

- COPP

Reliability:

- EFM
- CFM
- CPU Traffic
- UDLD
- ERPS
- Track
- VRRP
- VARP
- BFD

Network virtualization:

- VXLAN
- Overlay

Equipment management:

- STM
- log management
- Mirror
- Equipment Management
- Bootrom

Network management:

- Network Diagnostics
- SNMP
- LLDP

20.4 Stack Configuration Example

20.4.1 Example for Stacking of two switches(Linear topology)

Topology



Figure 20-9 Stacking map

Configuration steps

The following example describes the process of configuring a stacking system, taking a system consisting of two switches as an example.

If the following configuration does not specify which switch configuration it refers to, it indicates that all switches have the same configuration:

Step 1 Enter configuration mode

```
Switch# configure terminal
```

Step 2 Enable stacking

```
Switch(config)# stack enable
% Configuration will not take effect until save configuration and system reload.
```

NOTE After enabling or disabling the stack and restarting, the port-related configuration will be cleared.

Step 3 Configure the domainid of the stacking member

```
Switch(config)# stack domainid 65533
% Configuration will not take effect until save configuration and system reload.
```

Step 4 Configure the slotid of the stacking member

Configure on Switch1:

```
Switch(config)# stack slotid 1
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch2:

```
Switch(config)# stack slotid 2
% Configuration will not take effect until save configuration and system reload
```

Step 5 (Optional) Configure Priority

Configure on Switch1:

```
Switch(config)# stack slotid 1 priority 225
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch2:

```
Switch(config)# stack slotid 2 priority 125
% Configuration will not take effect until save configuration and system reload
```

NOTE Use this command to configure the stack priority of the device. The higher the value, the higher the priority. During stacking role negotiation, the priority and MAC address of the device will be compared for role election. The default priority value is 100.

Step 6 Configure the stacking ports of the member devices

Configure on Switch1：

```
Switch(config)# interface stack-0-1
Switch(config-if)# member-port eth-0-26
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Configure on Switch2：

```
Switch(config)# interface stack-0-1
Switch(config-if)# member-port eth-0-26
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Step 7 Exit configuration mode

```
Switch(config-if)# end
```

Step 8 Save Configuration

```
Switch# write
```

Step 9 Restart

```
Switch# reload
```

Step 10 Command Verification

The following command can be used to view the status of stacking

```
Switch(Master:1)# show stack
Stack domainID: 65533

SlotID   Role        MAC          State  priority
================================================================
 1    Master    001e:080d:6d62      Run    225
 2    Slave     001e:082b:79e9      Run    125

The System MAC of the Stacking is:001e:080d:6d62

Switch(Master:1)#
```

The following command can be used to view the interface status of stacking

```
Switch(Master:1)# show stack link
Stack Interface Status      Member       Member Status
============================================================
stack-01-01     up       eth-01-26      up
stack-02-01     up       eth-02-26      up
Switch(Master:1)#
```

The following command can be used to view the stack topology

```
Switch(Master:1)# show stack topology
 M        S
+-----+     +-----+
| 1 | 1*** 1| 2 |
+-----+     +-----+

Known Ucast Forward:
SlotID  Stack-Interface  Remote-SlotID
============================================
 1    stack-1-1          2

Switch(Master:1)#
```

20.4.2 Example for Stacking of multiple devices(Ring topology)

Topology

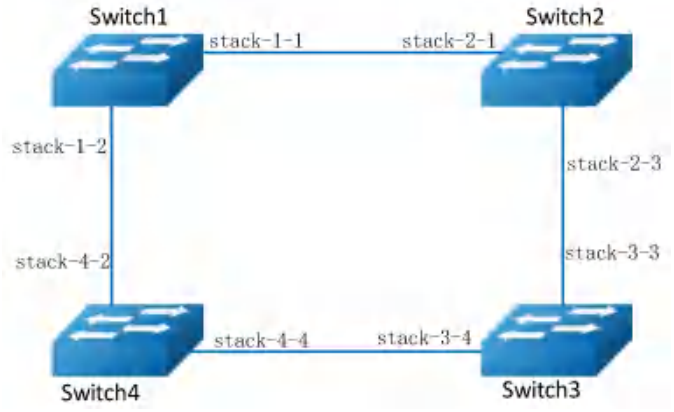


Figure 20-10 Stacking map

Configuration steps

The following example describes the process of configuring a stacking system, taking a system consisting of four switches as an example.

If the following configuration does not specify which switch configuration it refers to, it indicates that all switches have the same configuration:

Step 1 Enter configuration mode

```
Switch# configure terminal
```

Step 2 Enable stacking

```
Switch(config)# stack enable
% Configuration will not take effect until save configuration and system reload.
```

NOTE After enabling or disabling the stack and restarting, the port-related configuration will be cleared.

Step 3 Configure the domainid of the stacking member

```
Switch(config)# stack domainid 60000
% Configuration will not take effect until save configuration and system reload.
```

Step 4 Configure the slotid of the stacking member

Configure on Switch1:

```
Switch(config)# stack slotid 1
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch2:

```
Switch(config)# stack slotid 2
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch3:

```
Switch(config)# stack slotid 3
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch4:

```
Switch(config)# stack slotid 4
% Configuration will not take effect until save configuration and system reload
```

Step 5 (Optional) Configure Priority

Configure on Switch1:

```
Switch(config)# stack slotid 1 priority 190
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch2:

```
Switch(config)# stack slotid 2 priority 180
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch3:

```
Switch(config)# stack slotid 1 priority 170
% Configuration will not take effect until save configuration and system reload
```

Configure on Switch4:

```
Switch(config)# stack slotid 2 priority 160
% Configuration will not take effect until save configuration and system reload
```

NOTE Use this command to configure the stack priority of the device. The higher the value, the higher the priority. During stacking role negotiation, the priority and MAC address of the device will be compared for role election. The default priority value is 100.

Step 6 Configure the stacking ports of the member devices

Configure on Switch1：

```
Switch(config)# interface stack-0-1
Switch(config-if)# member-port eth-0-25
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
Switch(config)# interface stack-0-2
Switch(config-if)# member-port eth-0-26
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Configure on Switch2：

```
Switch(config)# interface stack-0-1
Switch(config-if)# member-port eth-0-49
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
Switch(config)# interface stack-0-3
Switch(config-if)# member-port eth-0-50
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Configure on Switch3：

```
Switch(config)# interface stack-0-3
Switch(config-if)# member-port eth-0-29
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
Switch(config)# interface stack-0-4
Switch(config-if)# member-port eth-0-30
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Configure on Switch4：

```
Switch(config)# interface stack-0-2
Switch(config-if)# member-port eth-0-30
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
Switch(config)# interface stack-0-4
Switch(config-if)# member-port eth-0-29
It may take a few seconds, please wait.....
Please add stacking member ports in pairs on both sides!
```

Step 7 Exit configuration mode

```
Switch(config-if)# end
```

Step 8 Save Configuration

```
Switch# write
```

Step 9 Restart

```
Switch# reload
```

Step 10 Command Verification

The following command can be used to view the status of stacking

```
Switch(Master:1)# show stack
Stack domainID: 60000

SlotID   Role        MAC          State  priority
================================================================
 1    Master    bc97:40e5:c873      Run    190
 2     Slave    d85b:2217:9372      Run    180
 3     Slave    001e:082b:79e9      Run    170
 4     Slave    001e:080d:6d62      Run    160

The System MAC of the Stacking is:bc97:40e5:c873

Switch(Master:1)#
```

The following command can be used to view the interface status of stacking

```
Switch(Master:1)# show stack link
Stack Interface Status      Member       Member Status
============================================================
stack-01-01     up       eth-01-25      up
stack-01-02     up       eth-01-26      up
stack-02-01     up       eth-02-49      up
stack-02-03     up       eth-02-50      up
stack-03-03     up       eth-03-29      up
stack-03-04     up       eth-03-30      up
stack-04-02     up       eth-04-30      up
stack-04-04     up       eth-04-29      up
Switch(Master:1)#
```

The following command can be used to view the stack topology

```
Switch(Master:1)# show stack topology
 M       S       S       S       M
+-----+    +-----+    +-----+    +-----+    +-----+
| 1 | 1*** 1| 2 | 3*** 3| 3 | 4*** 4| 4 | 2*** 2| 1 |
+-----+    +-----+    +-----+    +-----+    +-----+

Known Ucast Forward:
SlotID  Stack-Interface  Remote-SlotID
=============================================
 1    stack-1-1         2
 1    stack-1-1         3
 1    stack-1-2         4

Switch(Master:1)#
```

20.4.3 Example for Stacking of Dual-Active Detection

Topology

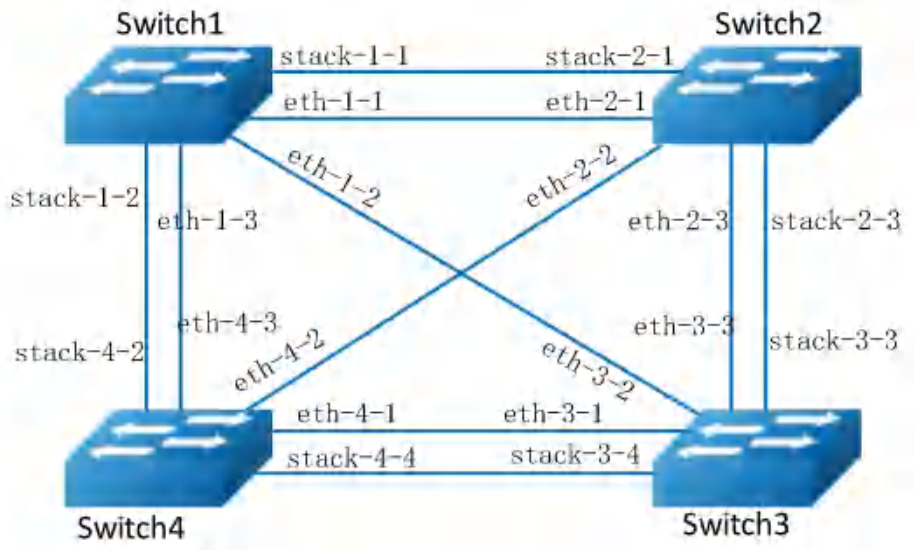


Figure 20-11 dual-active detection map

Configuration steps

The following example describes the process of configuring dual-active detection in a direct connection manner. Switch 1, 2, 3, and 4 form a stacking system, with each switch having a direct connection port to all other switches for dual-active detection, forming a full-mesh topology.

If the following configuration does not specify which switch configuration it is referring to, it indicates that all switches have the same configuration:

Step 1 Enter configuration mode

```
Switch# configure terminal
```

Step 2 Configure dual-active detection ports

Configure on Master：

```
Switch(config)# interface range eth-1-1 - 3
Switch(config-if)# stack dual-active detect mode direct
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Switch(Master:1)(config-if-range)# exit
Switch(Master:1)(config)#
Switch(Master:1)(config)# int range eth-2-1 - 3
Switch(Master:1)(config-if-range)# stack dual-active detect mode direct
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Switch(Master:1)(config-if-range)# exit
Switch(Master:1)(config)#
Switch(Master:1)(config)# int range eth-3-1 - 3
Switch(Master:1)(config-if-range)# stack dual-active detect mode direct
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Switch(Master:1)(config-if-range)# exit
Switch(Master:1)(config)#
```

```
Switch(Master:1)(config)# int range eth-4-1 - 3
Switch(Master:1)(config-if-range)# stack dual-active detect mode direct
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Warning: The interface will block common data packets, Continue? [no]y
Switch(Master:1)(config-if-range)# exit
Switch(Master:1)(config)#
end
Switch(Master:1)#
```

NOTE The dual-active detection ports will block the forwarding of ordinary packets, such as known-unicast, known-multicast, unknown-unicast, unknown-multicast, and broadcast.

Step 3 Configure reserved ports of dual-active detection

```
Switch(Master:1)(config)# stack dual-active detect reserved interface eth-1-12
Switch(Master:1)(config)# stack dual-active detect reserved interface eth-2-15
Switch(Master:1)(config)# stack dual-active detect reserved interface eth-3-5
Switch(Master:1)(config)# stack dual-active detect reserved interface eth-4-18
Switch(Master:1)(config)#
```

NOTE The dual-active detection reserved ports can be configured according to the actual networking setup. If the dual-active detection competition fails, the business ports of the failed party's stacking system, except for the stacking ports and reserved ports, will be errdisable down.

Step 4 Exit configuration mode

```
Switch(Master:1)(config)# end
```

Step 5 Save Configuration

```
Switch(Master:1)# write memory slot all
```

Step 6 Command Verification

```
Switch(Master:1)# show stack dual-active
Stack domainID: 60000
Dual-active status: Normal
Dual-active detect mode: Direct
Dual-active detect interface configured:
 eth-1-1     up
 eth-1-2     up
 eth-1-3     up
 eth-2-1     up
 eth-2-2     up
 eth-2-3     up
 eth-3-1     up
 eth-3-2     up
 eth-3-3     up
 eth-4-1     up
 eth-4-2     up
 eth-4-3     up
Dual-active reserved interface configured:
 eth-1-12     down
 eth-2-15     down
 eth-3-5     down
 eth-4-18     up
Switch(Master:1)#
```

```
Switch(Master:1)#
```

21 MPLS Configuration Guide

21.1 Configuring LDP

21.1.1 Overview

Brief Introduction

This chapter describes how to configure LDP.

A fundamental concept in MPLS is that two Label Switching Routers (LSRs) must agree on the meaning of the labels used to forward traffic between and through them. This common understanding is achieved by using a set of procedures, called label distribution protocol -LDP. The OS software supports these features:

- Downstream unsolicited label distribution with liberal retention mode.
- Supports control-mode modification.
- Supports lsr-id and transport-address modification.
- Supports target peer setting.
- Supports outbound label filtering.
- Supports explicit null label.

This configuration guide will describe the basic configuration of LDP in our system and give some examples for it.

More information about LDP, please see RFC3031 and FRC3036.

21.1.2 Configuration Examples

LDP Configuration

Topology

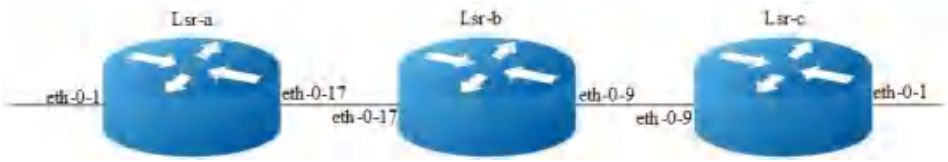


Figure 21-1 LSP map

Configuration Steps

The following example will describe how to use LDP to set up a label switching path (LSP) from Switch1 to Switch3.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for Switch1, interface need enable ldp and enable label switch:

```
Switch1(config)# interface eth-0-17
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 11.11.17.1/24
Switch1(config-if)# enable-ldp
Switch1(config-if)# label-switching
Switch1(config-if)# exit
Switch1(config)# interface loopback0
```

```
Switch1(config-if)# ip address 1.1.1.1/32
Switch1(config-if)# exit
```

Interface configuration for Switch2, interface need enable ldp and enable label switch:

```
Switch2(config)# interface eth-0-17
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 11.11.17.2/24
Switch2(config-if)# enable-ldp
Switch2(config-if)# label-switching
Switch2(config-if)# exit
Switch2(config)# interface eth-0-9
Switch2(config-if)# no switchport
Switch2(config-if)# ip address 11.11.9.1/24
Switch2(config-if)# enable-ldp
Switch2(config-if)# label-switching
Switch2(config-if)# exit
Switch2(config)# interface loopback0
Switch2(config-if)# ip address 2.2.2.2/32
Switch2(config-if)# exit
```

Interface configuration for Switch3, interface need enable ldp and enable label switch:

```
Switch3(config)# interface eth-0-9
Switch3(config-if)# no switchport
Switch3(config-if)# ip address 11.11.9.2/24
Switch3(config-if)# enable-ldp
Switch3(config-if)# label-switching
Switch3(config-if)# exit
Switch3(config)# interface loopback0
Switch3(config-if)# ip address 3.3.3.3/32
Switch3(config-if)# exit
```

Step 3 Enable router ldp

Configuration for Switch1:

```
Switch1(config)# router ldp
Switch1(config-router)# router-id 1.1.1.1
Switch1(config-router)# exit
```

Configuration for Switch2:

```
Switch2(config)# router ldp
Switch2(config-router)# router-id 2.2.2.2
Switch2(config-router)# exit
```

Configuration for Switch3:

```
Switch3(config)# router ldp
Switch3(config-router)# router-id 3.3.3.3
Switch3(config-router)# exit
```

Step 4 Enable router rip

```
Switch(config)# router rip
Switch(config-router)# network 0.0.0.0/0
Switch(config-router)# exit
```

Step 5 Exit the configure mode

```
Switch(config)# end
```

Step 6 Validation

Display the result of Switch1 ldp session state:

```
Switch1# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
2.2.2.2           eth-0-17  Passive  OPERATIONAL  30
```

Display the result of Switch2 ldp session state:

```
Switch2# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
3.3.3.3           eth-0-9   Active   OPERATIONAL  30
1.1.1.1           eth-0-17  Active   OPERATIONAL  30
```

Display the result of Switch3 ldp session state:

```
Switch3# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
2.2.2.2           eth-0-9   Passive  OPERATIONAL  30
```

21.2 Configuring MPLS

21.2.1 Overview

Brief Introduction

MPLS stands for "Multiprotocol Label Switching", multiprotocol, because its techniques are applicable to ANY network layer protocol. In this document, however, we focus on the use of IP as the network layer protocol.

Packet headers contain considerably more information than is needed simply to choose the next hop. Choosing the next hop can therefore be thought of as the composition of two functions. The first function partitions the entire set of possible packets into a set of "Forwarding Equivalence Classes (FECs)". Secondly maps each FEC to a next hop. So far as the forwarding decision is concerned, different packets which get mapped into the same FEC are indistinguishable. All packets which belong to a particular FEC and which travel from a particular node will follow the same path (or if certain kinds of multi-path routing are in use, they will all follow one of a set of paths associated with the FEC). In conventional IP forwarding, a particular router will typically consider two packets to be in the same FEC if there is some address prefix X in that router's routing tables such that X is the "longest match" for each packet's destination address. As the packet traverses the network, each hop in turn reexamines the packet and assigns it to a FEC.

In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded as a short fixed length value known as a "label". When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "labeled" before they are forwarded. At subsequent hops, there is no further analysis of the packet's network layer header. Rather, the label is used as an index into a table which specifies the next hop, and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop.

In the MPLS forwarding paradigm, once a packet is assigned to a FEC, no further header analysis is done by subsequent routers; all forwarding is driven by the labels.

21.2.2 Configuration Examples

MPLS LSP Configuration

Topology

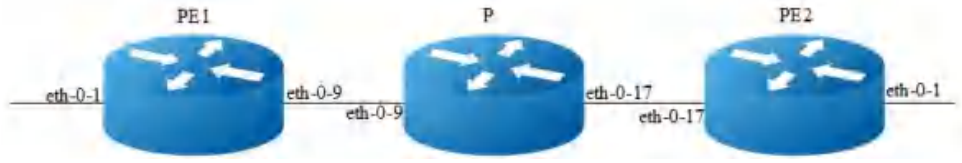


Figure 21-2 MPLS LSP model

Configuration Steps

The following example will describe how to configure MPLS LSP.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1, interface need enable label switch:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 11.11.9.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface eth-0-1
PE1(config-if)# no switchport
PE1(config-if)# ip address 10.10.10.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for P, interface need enable label switch:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 11.11.9.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 11.11.17.2/24
P(config-if)# label-switching
P(config-if)# exit
```

Interface configuration for PE2, interface need enable label switch:

```
PE2(config)# interface eth-0-17
PE2(config-if)# no switchport
PE2(config-if)# ip address 11.11.17.3/24
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface eth-0-1
PE2(config-if)# no switchport
PE2(config-if)# ip address 20.20.20.1/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 3 Configure static ftn/ilm

Static ftn for PE1:

```
PE1(config)# mpls ftn-entry 172.22.4.1/24 100 11.11.9.2
```

Static ilm for P:

```
P(config)# mpls ilm-entry swap 100 11.11.17.3 200
```

Static ilm for PE2:

```
PE2(config)# mpls ilm-entry php 200 20.20.20.2
```

Step 4 Exit the configure mode

```
Switch(config)# end
```

Step 5 Validation

Display the ftn lists on PE1:

```
PE1# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
    L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
    U - unknown FTN

Code   FEC             Out-Label   Nexthop       Out-Intf
K>     172.22.4.0/24     100       11.11.9.2     eth-0-9
```

Display the ilm lists on P:

```
P# show mpls ilm-database
Codes: > - selected ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
    L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut,
    U - unknown ILM

Code    FEC           I/O Label        Nexthop       Out-Intf
K>      0.0.0.0/0       100/200         11.11.17.3    eth-0-17
```

Display the ilm lists on PE2:

```
PE2# show mpls ilm-database
Codes: > - selected ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
    L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut,
    U - unknown ILM

Code    FEC           I/O Label        Nexthop       Out-Intf
K>      0.0.0.0/0       200/3          20.20.20.2    eth-0-1
```

21.3 Configuring VPLS

21.3.1 Overview

Brief Introduction

This chapter describes how to configure VPLS. Virtual Private LAN Service (VPLS) provides a way to enable transparent Layer-2 Ethernet LAN services to geographically dispersed customer sites connected by a Wide Area Network (WAN) by providing support for traditional Layer-2 broadcast and multicast services.

21.3.2 Configuration Examples

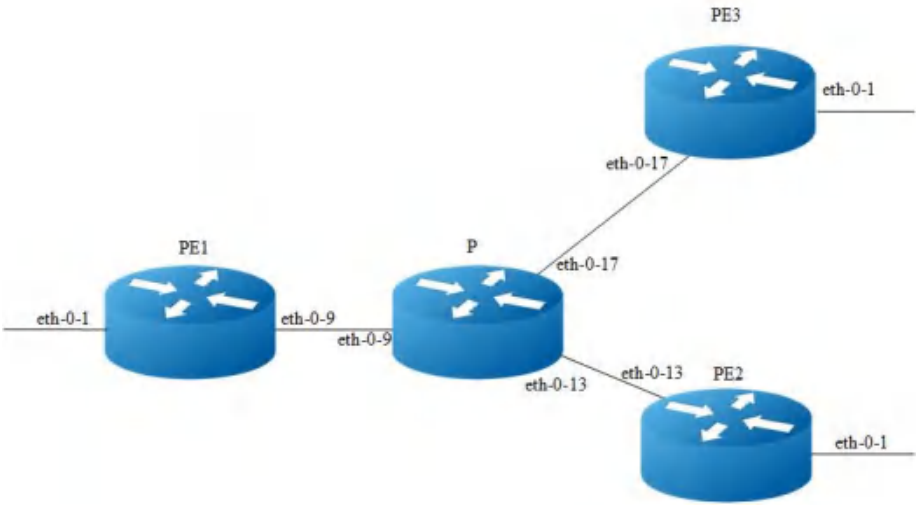


Figure 21-3 VPLS model

Configuring VPLS using LDP

The following example will describe how to use LDP to configure VPLS:

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1, eth-0-9 need enable ldp and enable label switch:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 11.11.9.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 11.11.1.1/32
PE1(config-if)# exit
```

Interface configuration for PE2, eth-0-13 need enable ldp and enable label switch:

```
PE2(config)# interface eth-0-13
PE2(config-if)# no switchport
PE2(config-if)# ip address 11.11.13.4/24
PE2(config-if)# enable-ldp
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 11.11.4.4/32
PE2(config-if)# exit
```

Interface configuration for PE3, eth-0-17 need enable ldp and enable label switch:

```
PE3(config)# interface eth-0-17
PE3(config-if)# no switchport
```

```
PE3(config-if)# ip address 11.11.17.3/24
PE3(config-if)# enable-ldp
PE3(config-if)# label-switching
PE3(config-if)# exit
PE3(config)# interface loopback 0
PE3(config-if)# ip address 11.11.3.3/32
PE3(config-if)# exit
```

Interface configuration for P, interface need enable ldp and enable label switch:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 11.11.9.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-13
P(config-if)# no switchport
P(config-if)# ip address 11.11.13.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 11.11.17.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface loopback 0
P(config-if)# ip address 11.11.2.2/32
P(config-if)# exit
```

Step 3 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 11.11.1.1
PE1(config-router)# targeted-peer 11.11.3.3
PE1(config-router)# targeted-peer 11.11.4.4
PE1(config-router)# transport-address 11.11.1.1
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 11.11.4.4
PE2(config-router)# transport-address 11.11.4.4
PE2(config-router)# targeted-peer 11.11.1.1
PE2(config-router)# targeted-peer 11.11.3.3
PE2(config-router)# exit
```

LDP configuration for PE3:

```
PE3(config)# router ldp
PE3(config-router)# router-id 11.11.3.3
PE3(config-router)# transport-address 11.11.3.3
PE3(config-router)# targeted-peer 11.11.1.1
```

```
PE3(config-router)# targeted-peer 11.11.4.4
PE3(config-router)# exit
```

LDP configuration for P:

```
P(config)# router ldp
P(config-router)# router-id 11.11.2.2
P(config-router)# exit
```

Step 4 Enable router rip

```
Switch(config)# router rip
Switch(config-router)# network 11.11.1.1/16
Switch(config-router)# exit
```

Step 5 Create a VPLS instance

Config PE1, PE2 and PE3 VPLS PW raw mode, and assign their vpls peers.

VPLS instance for PE1:

```
PE1(config)# mpls vpls v1 100
PE1(config-vpls)# vpls-peer 11.11.3.3 raw
PE1(config-vpls)# vpls-peer 11.11.4.4 raw
PE1(config-vpls)# exit
```

VPLS instance for PE2:

```
PE2(config)# mpls vpls v4 100
PE2(config-vpls)# vpls-peer 11.11.1.1 raw
PE2(config-vpls)# vpls-peer 11.11.3.3 raw
PE2(config-vpls)# exit
```

VPLS instance for PE3:

```
PE3(config)# mpls vpls v3 100
PE3(config-vpls)# vpls-peer 11.11.1.1 raw
PE3(config-vpls)# vpls-peer 11.11.4.4 raw
PE3(config-vpls)# exit
```

Step 6 bind the interface and the VPLS instance

Config AC of PE1, PE2 and PE3 VLAN access mode.

Interface configuration for PE1:

```
PE1(config)# interface eth-0-1
PE1(config-if)# switchport mode trunk
PE1(config-if)# mpls-vpls v1 vlan 2
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-1
PE2(config-if)# switchport mode trunk
PE2(config-if)# mpls-vpls v4 vlan 2
PE2(config-if)# exit
```

Interface configuration for PE3:

```
PE3(config)# interface eth-0-1
PE3(config-if)# switchport mode trunk
PE3(config-if)# mpls-vpls v3 vlan 2
PE3(config-if)# exit
```

Step 7 Exit the configure mode

```
Switch(config)# end
```

Step 8 Validation

Use the show ldp session and the show mpls vpls mesh commands respectively to display complete information about theVPLS. Show ldp session command can get LDP peer's state. Show mpls vpls mesh command can get vpls peer's state and the inner labels vpls using.The following are the sample outputs for the show commands displaying VPLS.

Display the result on PE1:

```
PE1# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
11.11.3.3         eth-0-9   Passive  OPERATIONAL  30
11.11.4.4         eth-0-9   Passive  OPERATIONAL  30
11.11.2.2         eth-0-9   Passive  OPERATIONAL  30
PE1# show mpls vpls mesh
VPLS-ID   Peer Addr/name    In-Label  Out-Intf  Out-Label  Type    St
100       11.11.3.3/-       32768     eth-0-9   32768      RAW     Up
100       11.11.4.4/-       32773     eth-0-9   32768      RAW     Up
```

Display the result on PE2 :

```
PE2# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
11.11.1.1         eth-0-13  Active   OPERATIONAL  30
11.11.3.3         eth-0-13  Active   OPERATIONAL  30
11.11.2.2         eth-0-13  Passive  OPERATIONAL  30
PE2# show mpls vpls mesh
VPLS-ID   Peer Addr/name    In-Label  Out-Intf  Out-Label  Type    St
100       11.11.1.1/-       32768     eth-0-13  32773      RAW     Up
100       11.11.3.3/-       32769     eth-0-13  32770      RAW     Up
```

Display the result on PE3 :

```
PE3# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
11.11.1.1         eth-0-17  Active   OPERATIONAL  30
11.11.4.4         eth-0-17  Passive  OPERATIONAL  30
11.11.2.2         eth-0-17  Passive  OPERATIONAL  30
PE3# show mpls vpls mesh
VPLS-ID   Peer Addr/name    In-Label  Out-Intf  Out-Label  Type    St
100       11.11.1.1/-       32768     eth-0-17  32768      RAW     Up
100       11.11.4.4/-       32770     eth-0-17  32769      RAW     Up
```

Display the result on P :

```
P# show ldp session
Peer IP Address   IF Name   My Role  State       KeepAlive
11.11.1.1         eth-0-9   Active   OPERATIONAL  30
11.11.3.3         eth-0-17  Active   OPERATIONAL  30
11.11.4.4         eth-0-13  Active   OPERATIONAL  30
```

Configuring VPLS using static conmmand

The following example will describe how to configure static VPLS:

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1, eth-0-9 need enable label switch:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 11.11.9.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for PE2, eth-0-13 need enable label switch:

```
PE2(config)# interface eth-0-13
PE2(config-if)# no switchport
PE2(config-if)# ip address 11.11.13.1/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Interface configuration for PE3, eth-0-17 need enable label switch:

```
PE3(config)# interface eth-0-17
PE3(config-if)# no switchport
PE3(config-if)# ip address 11.11.17.1/24
PE3(config-if)# label-switching
PE3(config-if)# exit
```

Interface configuration for P, eth-0-9, eth-0-13 and eth-0-17 need enable label switch:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 11.11.9.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 11.11.17.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-13
P(config-if)# no switchport
P(config-if)# ip address 11.11.13.2/24
P(config-if)# label-switching
P(config-if)# exit
```

Step 3 Configure ftn entry

Interface configuration for PE1:

```
PE1(config)# mpls ftn-entry 11.11.17.1/24 97 11.11.9.2
PE1(config)# mpls ftn-entry 11.11.13.1/24 93 11.11.9.2
```

Interface configuration for PE2:

```
PE2(config)# mpls ftn-entry  11.11.9.1/32 44 11.11.13.2
```

Interface configuration for PE3:

```
PE3(config)#mpls ftn-entry 11.11.9.1/32 33 11.11.17.2
```

Step 4 Create a VPLS instance

Config PE1, PE2 and PE3 VPLS PW raw mode, and assign their vpls peers.

VPLS instance for PE1:

```
PE1(config)# mpls vpls vpls1 1
PE1(config-vpls)# vpls-peer 11.11.17.1 raw manual
PE1(config-vpls)# vpls-peer 11.11.13.1 raw manual
```

VPLS instance for PE2:

```
PE2(config)# mpls vpls vpls1 1
PE2(config-vpls)# vpls-peer 11.11.9.1 raw manual
```

VPLS instance for PE3:

```
PE3(config)# mpls vpls vpls1 1
PE3(config-vpls)# vpls-peer 11.11.9.1 raw manual
```

Step 5 bind the interface and the VPLS instance

Config AC of PE1, PE2 and PE3 VLAN access mode.

Interface configuration for PE1:

```
PE1(config)# interface eth-0-1
PE1(config-if)# switchport mode trunk
PE1(config-if)# mpls-vpls vpls1 vlan 100
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-1
PE2(config-if)# switchport mode trunk
PE2(config-if)# mpls-vpls vpls1 vlan 100
PE2(config-if)# exit
```

Interface configuration for PE3:

```
PE3(config)#interface eth-0-1
PE3(config-if)# switchport mode trunk
PE3(config-if)# mpls-vpls vpls1 vlan 100
PE3(config-if)# exit
```

Step 6 Configure VPLS FIB

VPLS FIB for PE1:

```
PE1(config)# vpls-fib-add vpls1 peer 11.11.17.1 103 31
PE1(config)# vpls-fib-add vpls1 peer 11.11.13.1 102 201
```

VPLS FIB for PE2:

```
PE2(config)# vpls-fib-add vpls1 peer 11.11.9.1 201 102
```

VPLS FIB for PE3:

```
PE3(config)# vpls-fib-add vpls1 peer 11.11.9.1 31 103
```

Step 7 Configure static ilm

Static ilm for P:

```
P(config)# mpls ilm-entry php 97 11.11.17.1
P(config)# mpls ilm-entry php 93 11.11.13.1
P(config)# mpls ilm-entry php 33 11.11.9.1
P(config)# mpls ilm-entry php 44 11.11.9.1
```

Step 8 Exit the configure mode

```
Switch(config)# end
```

Step 9 Validation

Show mpls vpls mesh command can get vpls peer's state and the inner labels vpls using.

Display the result on PE1:

```
PE1# show mpls vpls mesh
VPLS-ID   Peer Addr/name      In-Label  Out-Intf  Out-Label  Type    St
1         11.11.13.1/-        102       eth-0-9   201        RAW     Up
1         11.11.17.1/-        103       eth-0-9   31         RAW     Up
```

Display the result on PE2:

```
PE2# show mpls vpls mesh
VPLS-ID   Peer Addr/name      In-Label  Out-Intf  Out-Label  Type    St
1         11.11.9.1/-         201       eth-0-13  102        RAW     Up
```

Display the result on PE3:

```
PE3# show mpls vpls mesh
VPLS-ID   Peer Addr/name      In-Label  Out-Intf  Out-Label  Type    St
1         11.11.9.1/-         31        eth-0-17  103        RAW     Up
```

Display the result on P:

```
P# show mpls ilm-database
Codes: > - selected ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
    L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut,
    U - unknown ILM
Code   FEC          I/O Label       Nexthop       Out-Intf
K>     0.0.0.0/0    33/3            11.11.9.1     eth-0-9
K>     0.0.0.0/0    44/3            11.11.9.1     eth-0-9
K>     0.0.0.0/0    93/3            11.11.13.1    eth-0-13
K>     0.0.0.0/0    97/3            11.11.17.1    eth-0-17
```

Configuring Tunnel L2 protocol packets by VPLS

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure.

The following example will display how to tunnel STP protocol packets by vpls. Users can configure other L2 protocol packets like that. The following configuration is also based on Figure VPLS model topology.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enable L2 protocol globally

```
Switch(config)# l2protocol enable
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1, eth-0-9 need enable ldp and enable label switch:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 11.11.9.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface loopback 0
```

```
PE1(config-if)# ip address 11.11.1.1/32
PE1(config-if)# exit
```

Interface configuration for PE2, eth-0-13 need enable ldp and enable label switch:

```
PE2(config)# interface eth-0-13
PE2(config-if)# no switchport
PE2(config-if)# ip add 11.11.13.4/24
PE2(config-if)# enable-ldp
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 11.11.4.4/32
PE2(config-if)# exit
```

Interface configuration for PE3, eth-0-17 need enable ldp and enable label switch:

```
PE3(config)# interface eth-0-17
PE3(config-if)# no switchport
PE3(config-if)# ip address 11.11.17.3/24
PE3(config-if)# enable-ldp
PE3(config-if)# label-switching
PE3(config-if)# exit
PE3(config)# interface loopback 0
PE3(config-if)# ip address 11.11.3.3/32
PE3(config-if)# exit
```

Interface configuration for P, interface need enable ldp and enable label switch:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 11.11.9.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-13
P(config-if)# no switchport
P(config-if)# ip address 11.11.13.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 11.11.17.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface loopback 0
P(config-if)# ip address 11.11.2.2/32
P(config-if)# exit
```

Step 4 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 11.11.1.1
PE1(config-router)# targeted-peer 11.11.3.3
PE1(config-router)# targeted-peer 11.11.4.4
```

```
PE1(config-router)# transport-address 11.11.1.1
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 11.11.4.4
PE2(config-router)# transport-address 11.11.4.4
PE2(config-router)# targeted-peer 11.11.1.1
PE2(config-router)# targeted-peer 11.11.3.3
PE2(config-router)# exit
```

LDP configuration for PE3:

```
PE3(config)# router ldp
PE3(config-router)# router-id 11.11.3.3
PE3(config-router)# transport-address 11.11.3.3
PE3(config-router)# targeted-peer 11.11.1.1
PE3(config-router)# targeted-peer 11.11.4.4
PE3(config-router)# exit
```

LDP configuration for P:

```
P(config)# router ldp
P(config-router)# router-id 11.11.2.2
P(config-router)# exit
```

Step 5 Enable router rip

RIP configuration for PE1/PE2/PE3:

```
Switch(config)# router rip
Switch(config-router)# network 11.11.1.1/16
Switch(config-router)# exit
```

Step 6 Create a VPLS instance

Config PE1, PE2 and PE3 VPLS PW raw mode, and assign their vpls peers.

VPLS instance for PE1:

```
PE1(config)# mpls vpls v1 100
PE1(config-vpls)# vpls-peer 11.11.3.3 raw
PE1(config-vpls)# vpls-peer 11.11.4.4 raw
PE1(config-vpls)# exit
```

VPLS instance for PE2:

```
PE2(config)# mpls vpls v4 100
PE2(config-vpls)# vpls-peer 11.11.1.1 raw
PE2(config-vpls)# vpls-peer 11.11.3.3 raw
PE2(config-vpls)# exit
```

VPLS instance for PE3:

```
PE3(config)# mpls vpls v3 100
PE3(config-vpls)# vpls-peer 11.11.1.1 raw
PE3(config-vpls)# vpls-peer 11.11.4.4 raw
PE3(config-vpls)# exit
```

Step 7 bind the interface and the VPLS instance

Config AC of PE1, PE2 and PE3 ethernet access mode.

Interface configuration for PE1:

```
PE1(config)# interface eth-0-1
PE1(config-if)# mpls-vpls v1 ethernet
PE1(config-if)# l2protocol stp tunnel
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-1
PE2(config-if)# mpls-vpls v4 ethernet
PE2(config-if)# l2protocol stp tunnel
PE2(config-if)# exit
```

Interface configuration for PE3:

```
PE3(config)# interface eth-0-1
PE3(config-if)# mpls-vpls v3 ethernet
PE3(config-if)# l2protocol stp tunnel
PE3(config-if)# exit
```

Step 8 Exit the configure mode

```
Switch(config)# end
```

Configuring static MAC entries for VPLS

In a Virtual Switch Instance (VSI), if a PE receives a packet with an unknown destination MAC address, the PE will flood the packet. User can configure static MAC entries to specify the interface or peer node to which the received packets to be forwarded. The following example shows how to configure static MAC entries for a VSI. The following configuration is based on Figure VPLS model topology.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1, eth-0-9 need enable ldp and enable label switch:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 11.11.9.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 11.11.1.1/32
PE1(config-if)# exit
```

Interface configuration for PE2, eth-0-13 need enable ldp and enable label switch:

```
PE2(config)# interface eth-0-13
PE2(config-if)# no switchport
PE2(config-if)# ip add 11.11.13.4/24
PE2(config-if)# enable-ldp
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 11.11.4.4/32
PE2(config-if)# exit
```

Interface configuration for PE3, eth-0-17 need enable ldp and enable label switch:

```
PE3(config)# interface eth-0-17
PE3(config-if)# no switchport
PE3(config-if)# ip address 11.11.17.3/24
PE3(config-if)# enable-ldp
PE3(config-if)# label-switching
PE3(config-if)# exit
PE3(config)# interface loopback 0
PE3(config-if)# ip address 11.11.3.3/32
PE3(config-if)# exit
```

Interface configuration for P, interface need enable ldp and enable label switch:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 11.11.9.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-13
P(config-if)# no switchport
P(config-if)# ip address 11.11.13.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 11.11.17.2/24
P(config-if)# enable-ldp
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface loopback 0
P(config-if)# ip address 11.11.2.2/32
P(config-if)# exit
```

Step 3 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 11.11.1.1
PE1(config-router)# targeted-peer 11.11.3.3
PE1(config-router)# targeted-peer 11.11.4.4
PE1(config-router)# transport-address 11.11.1.1
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 11.11.4.4
PE2(config-router)# transport-address 11.11.4.4
PE2(config-router)# targeted-peer 11.11.1.1
PE2(config-router)# targeted-peer 11.11.3.3
PE2(config-router)# exit
```

LDP configuration for PE3:

```
PE3(config)# router ldp
PE3(config-router)# router-id 11.11.3.3
PE3(config-router)# transport-address 11.11.3.3
```

```
PE3(config-router)# targeted-peer 11.11.1.1
PE3(config-router)# targeted-peer 11.11.4.4
PE3(config-router)# exit
```

LDP configuration for P:

```
P(config)# router ldp
P(config-router)# router-id 11.11.2.2
P(config-router)# exit
```

Step 4 Enable router rip

```
Switch(config)# router rip
Switch(config-router)# network 11.11.1.1/16
Switch(config-router)# exit
```

Step 5 Create a VPLS instance

Config PE1, PE2 and PE3 VPLS PW raw mode, and assign their vpls peers.

VPLS instance for PE1:

```
PE1(config)# mpls vpls v1 100
PE1(config-vpls)# vpls-peer 11.11.3.3 raw
PE1(config-vpls)# vpls-peer 11.11.4.4 raw
PE1(config-vpls)# mac-address-table 0000.0000.0001 forward eth-0-1
PE1(config-vpls)# mac-address-table 0000.0000.0003 forward peer 11.11.3.3
PE1(config-vpls)# mac-address-table 0000.0000.0004 forward peer 11.11.4.4
PE1(config-vpls)# exit
```

VPLS instance for PE2:

```
PE2(config)# mpls vpls v4 100
PE2(config-vpls)# vpls-peer 11.11.1.1 raw
PE2(config-vpls)# vpls-peer 11.11.3.3 raw
PE2(config-vpls)# mac-address-table 0000.0000.0004 forward eth-0-1
PE2(config-vpls)# mac-address-table 0000.0000.0001 forward peer 11.11.1.1
PE2(config-vpls)# mac-address-table 0000.0000.0003 forward peer 11.11.3.3
PE2(config-vpls)# exit
```

VPLS instance for PE3:

```
PE3(config)# mpls vpls v3 100
PE3(config-vpls)# vpls-peer 11.11.1.1 raw
PE3(config-vpls)# vpls-peer 11.11.4.4 raw
PE3(config-vpls)# mac-address-table 0000.0000.0003 forward eth-0-1
PE3(config-vpls)# mac-address-table 0000.0000.0001 forward peer 11.11.1.1
PE3(config-vpls)# mac-address-table 0000.0000.0004 forward peer 11.11.4.4
PE3(config-vpls)# exit
```

Step 6 bind the interface and the VPLS instance

Config AC of PE1, PE2 and PE3 ethernet access mode.

Interface configuration for PE1:

```
PE1(config)# interface eth-0-1
PE1(config-if)# mpls-vpls v1 ethernet
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-1
PE2(config-if)# mpls-vpls v4 ethernet
PE2(config-if)# exit
```

Interface configuration for PE3:

```
PE3(config)# interface eth-0-1
PE3(config-if)# mpls-vpls v3 ethernet
PE3(config-if)# exit
```

Step 7 Exit the configure mode

```
Switch(config)# end
```

Step 8 Validation

Use the show mac address-table vpls to display complete information about the VPLS MAC entries. The following are the sample outputs for the show command.

Display the result on PE1:

```
PE1# show mac address-table vpls
vpls      peer          mac          static
v1        eth-0-1       0000.0000.0001 1
v1        11.11.3.3     0000.0000.0003 1
v1        11.11.4.4     0000.0000.0004 1
```

Display the result on PE2:

```
PE2# show mac address-table vpls
vpls      peer          mac          static
v1        eth-0-1       0000.0000.0004 1
v1        11.11.1.1     0000.0000.0001 1
v1        11.11.3.3     0000.0000.0003 1
```

Display the result on PE3:

```
PE3# show mac address-table vpls
vpls      peer          mac          static
v1        eth-0-1       0000.0000.0003 1
v1        11.11.1.1     0000.0000.0001 1
v1        11.11.4.4     0000.0000.0004 1
```

21.4 Configuring VPWS

21.4.1 Overview

Brief Introduction

This chapter describes how to configure VPWS. The MPLS L2CIRCUIT is a point-to-point Layer 2 connection transported by means of Multiprotocol Label Switching (MPLS) on the service provider's network. The Layer 2 circuit is transported over a single Label Switched Path (LSP) tunnel between two Provider Edge (PE) routers.

21.4.2 Configuration Examples

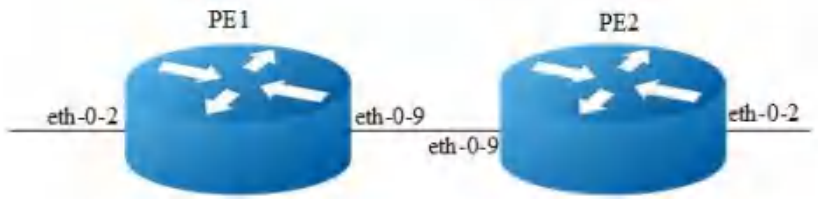


Figure 21-4 Topology of vpws configuration

Configuring VPWS using LDP

The Virtual Circuit module is a part of the LDP module. It is based on the IETF drafts proposed by Martini, et al [L2TRANS]. The Virtual Circuits module sets up virtual circuits for transporting Layer 2 protocols across an MPLS network. This chapter includes a step-by-step configuration of VPWS.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-2
PE1(config-if)# mpls-l2-circuit t1 ethernet
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 192.168.10.10/32
PE1(config-if)# exit
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 8.8.8.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-2
PE2(config-if)# mpls-l2-circuit t1 ethernet
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 192.168.11.10/32
PE2(config-if)# exit
PE2(config)# interface eth-0-9
PE2(config-if)# no switchport
PE2(config-if)# ip address 8.8.8.2/24
PE2(config-if)# enable-ldp
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 3 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 192.168.10.10
PE1(config-router)# targeted-peer 192.168.11.10
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 192.168.11.10
PE2(config-router)# targeted-peer 192.168.10.10
PE2(config-router)# exit
```

Step 4 Configure VPWS VC ID

VC ID configuration for PE1:

```
PE1(config)# mpls l2-circuit t1 200 192.168.11.10 raw
```

VC ID configuration for PE2:

```
PE2(config)# mpls l2-circuit t1 200 192.168.10.10 raw
```

Step 5 Enable router rip

```
Switch(config)# router rip
Switch(config-router)# network 0.0.0.0/0
Switch(config-router)# exit
```

Step 6 Exit the configure mode

```
Switch(config)# end
```

Step 7 Validation

Use the show mpls l2-circuit and the show mpls vc-table commands respectively to display complete information about the Layer-2 Virtual Circuit. The following are the sample outputs for the show commands displaying Layer-2 virtual circuit information.

Display the result on PE1:

```
PE1# show mpls l2-circuit
VC-Name          VC-ID    Interface   AC-type  VLAN  PW-mode  Manual
t1               200      eth-0-2     Ethernet N/A   Raw     No
PE1# show mpls vc-table
VC-ID     PW Intf  AC Intf  L/R Label     EndPoint      Status  Manual
200       eth-0-9  eth-0-2  32768/32768   192.168.11.10 Active  No
```

Display the result on PE2:

```
PE2# show mpls l2-circuit
VC-Name          VC-ID    Interface   AC-type  VLAN  PW-mode  Manual
t1               200      eth-0-2     Ethernet N/A   Raw     No
PE2# show mpls vc-table
VC-ID     PW Intf  AC Intf  L/R Label     EndPoint      Status  Manual
200       eth-0-9  eth-0-2  32768/32768   192.168.10.10 Active  No
```

VC configuration using static command

The following example will describe how to configure static VPWS

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-2
PE1(config-if)# mpls-l2-circuit t2 ethernet
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 192.168.10.10/32
PE1(config-if)# exit
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 8.8.8.1/24
```

```
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-2
PE2(config-if)# mpls-l2-circuit t2 ethernet
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 192.168.11.10/32
PE2(config-if)# exit
PE2(config)# interface eth-0-9
PE2(config-if)# no switchport
PE2(config-if)# ip address 8.8.8.2/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 3 Configure ftn entry

FTN entry for PE1:

```
PE1(config)# mpls ftn-entry 192.168.11.1/24 111 8.8.8.2
```

FTN entry for PE2:

```
PE2(config)# mpls ftn-entry 192.168.10.1/24 222 8.8.8.1
```

Step 4 Configure static ilm

Static ilm for PE1:

```
PE1(config)# mpls ilm-entry pop 212
```

Static ilm for PE2:

```
PE2(config)# mpls ilm-entry pop 111
```

Step 5 Configure VPWS VC ID

VC ID configuration for PE1:

```
PE1(config)# mpls l2-circuit t2 201 192.168.11.10 raw manual
PE1(config)# mpls l2-circuit-fib-entry t2 44 33
```

VC ID configuration for PE2:

```
PE2(config)# mpls l2-circuit t2 201 192.168.10.10 raw manual
PE2(config)# mpls l2-circuit-fib-entry t2 33 44
```

Step 6 Exit the configure mode

```
Switch(config)# end
```

Step 7 Validation

Use the show mpls l2circuitand the show mpls vc-table commands respectively to display complete information about the Layer-2 Virtual Circuit. The following are the sample outputs for the show commands displaying Layer-2 virtual circuit information.

Display the result on PE1:

```
PE1# show mpls l2-circuit
VC-Name      VC-ID    Interface   AC-type      VLAN  PW-mode  Manual
t2           201      eth-0-2     Ethernet     N/A   Raw      Yes
PE1# show mpls vc-table
VC-ID     PW Intf  AC Intf  L/R Label     EndPoint      Status  Manual
201       eth-0-9  eth-0-2  44/33         192.168.11.10 Active  Yes
```

Display the result on PE2:

```
PE2# show mpls l2-circuit
VC-Name      VC-ID    Interface   AC-type      VLAN  PW-mode   Manual
t2           201      eth-0-2     Ethernet     N/A   Raw       Yes
PE2# show mpls vc-table
VC-ID     PW Intf  AC Intf  L/R Label     EndPoint      Status   Manual
201       eth-0-9  eth-0-2  33/44         192.168.10.10 Active   Yes
```

Configuring Tunnel L2 protocol packets by VPWS

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. The following example will display how to tunnel STP protocol packets by vpws. Users can configure other L2 protocol packets like that. The following configuration is also based on chart 1.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Enable L2 protocol globally

```
Switch(config)# l2protocol enable
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-2
PE1(config-if)# mpls-l2-circuit t1 ethernet
PE1(config-if)# l2protocol stp tunnel
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 192.168.10.10/32
PE1(config-if)# exit
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 8.8.8.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# l2protocol enable
PE2(config)# interface eth-0-2
PE2(config-if)# mpls-l2-circuit t1 ethernet
PE2(config-if)# l2protocol stp tunnel
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 192.168.11.10/32
PE2(config-if)# exit
PE2(config)# interface eth-0-9
PE2(config-if)# no switchport
PE2(config-if)# ip address 8.8.8.2/24
PE2(config-if)# enable-ldp
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 4 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 192.168.10.10
PE1(config-router)# targeted-peer 192.168.11.10
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 192.168.11.10
PE2(config-router)# targeted-peer 192.168.10.10
PE2(config-router)# exit
```

Step 5 Configure VPWS VC ID

VC ID configuration for PE1:

```
PE1(config)# mpls l2-circuit t1 200 192.168.11.10 raw
```

VC ID configuration for PE2:

```
PE2(config)# mpls l2-circuit t1 200 192.168.10.10 raw
```

Step 6 Enable router rip

```
switch(config)# router rip
switch(config-router)# network 0.0.0.0/0
switch(config-router)# exit
```

Step 7 Exit the configure mode

```
Switch(config)# end
```

21.5 Configuring MPLS QoS

21.5.1 Overview

Brief Introduction

MPLS QoS is the important part of QoS network, which is usually implemented by DiffServ model.

MPLS use labels to take the place of routes, which is powerful, flexible and can satisfy all kinds of requirements.

Principle Description

MPLS LSP modelcontain three models：Uniform/Pipe/Short Pipe.

Uniform model: The packets on IP network and MPLS network have the same priority, which means the priority is take effect golbally. On the ingress device, the packets will be added labels and the exp will be mapped from dscp. On the egress device, the dscp of the packets will be mapped from exp.

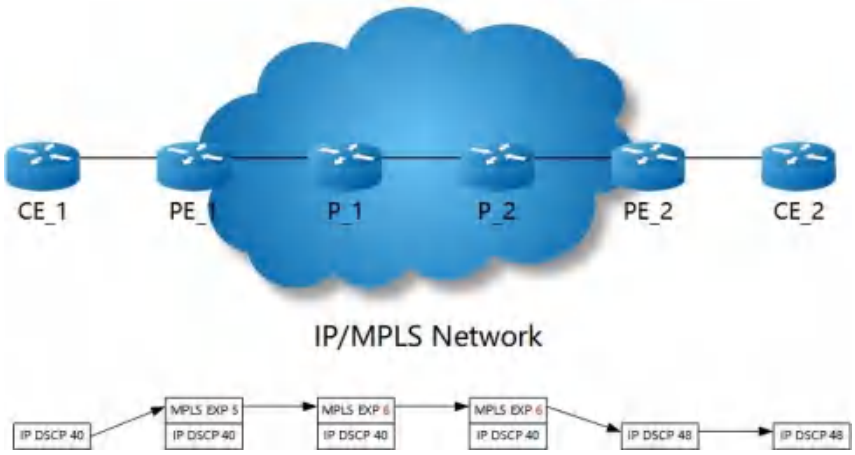


Figure 21-5 Uniform model

Pipe model: On the ingress device, the packets will be added labels and the exp will be assigned by the users. On the egress device, the phb will be mapped from exp and the output packetswill carry the original dscp.

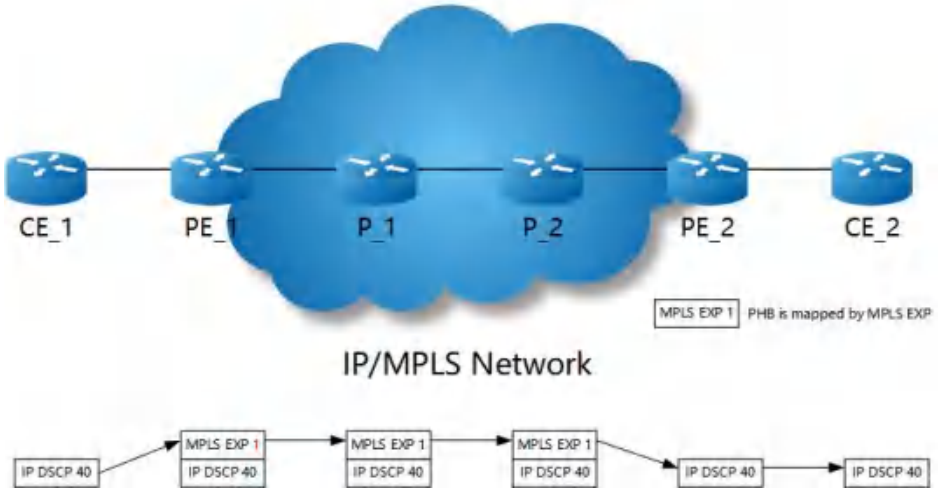


Figure 21-6 Pipe model

Pipe model: On the ingress device, the packets will be added labels and the exp will be assigned by the users. On the egress device, the phb will be mapped from dscp and the output packetswill carry the original dscp.

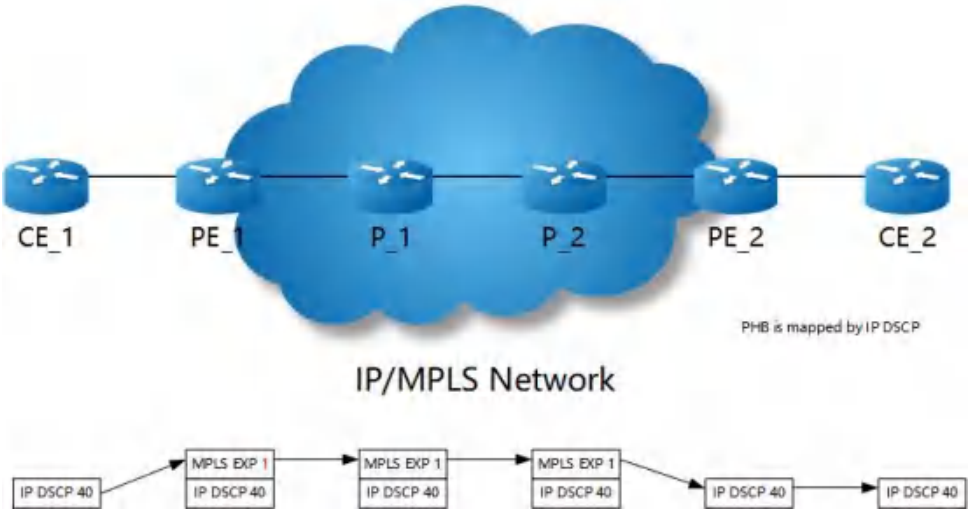


Figure 21-7 Short pipe model

21.5.2 Configuration Examples

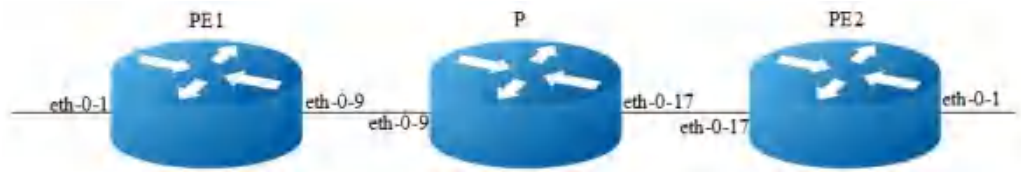


Figure 21-8 MPLS QoS LSP model

MPLS QoS Uniform Configuration

The following example will describe how to configure MPLS QoS Uniform model.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Set MPLS LSP Model

```
Switch(config)# mpls lsp-model uniform
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 10.0.9.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface eth-0-1
PE1(config-if)# no switchport
PE1(config-if)# qos domain type dscp 1
PE1(config-if)# trust dscp
PE1(config-if)# replace dscp
PE1(config-if)# ip address 1.1.1.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for P:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 10.0.9.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 10.0.17.2/24
P(config-if)# label-switching
P(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-17
PE2(config-if)# no switchport
PE2(config-if)# ip address 10.0.17.1/24
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface eth-0-1
PE2(config-if)# no switchport
PE2(config-if)# qos domain type dscp 1
PE2(config-if)# trust dscp
PE2(config-if)# replace dscp
PE2(config-if)# ip address 2.2.2.2/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 4 Configure static arp

Interface configuration for PE1:

```
PE1(config)# arp 1.1.1.2 0001.0001.0002
```

Interface configuration for PE2:

```
PE2(config)# arp 2.2.2.1 0002.0002.0001
```

Step 5 Configure static ftn/ilm

Static ftn for PE1:

```
PE1(config)# mpls ftn-entry 2.2.2.0/24 102 10.0.9.2
PE1(config)# mpls ilm-entry pop 201
```

Static ilm for P:

```
P(config)# mpls ilm-entry swap 102 10.0.17.1 203
P(config)# mpls ilm-entry swap 302 10.0.9.1 201
```

Static ilm for PE2:

```
PE2(config)# mpls ftn-entry 1.1.1.0/24 302 10.0.17.2
PE2(config)# mpls ilm-entry pop 203
```

Step 6 Validation

Display the result on PE1:

```
PE1# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN
```

```
Code   FEC            Out-Label   Nexthop       Out-Intf
K>     2.2.2.0/24       102       10.0.9.2      eth-0-9

PE1# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC            I/O Label        Nexthop       Out-Intf
K>     0.0.0.0/0       201/-           0.0.0.0       N/A
```

Display the result on P:

```
P# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC            I/O Label        Nexthop       Out-Intf
K>     0.0.0.0/0       102/203          10.0.17.1     eth-0-17
K>     0.0.0.0/0       302/201          10.0.9.1      eth-0-9
```

Display the result on PE2:

```
PE2# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN

Code   FEC            Out-Label   Nexthop       Out-Intf
K>     1.1.1.0/24       302       10.0.17.2     eth-0-17

PE2# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC            I/O Label        Nexthop       Out-Intf
K>     0.0.0.0/0       203/-           0.0.0.0       N/A
```

MPLS QoS Pipe Configuration

The following example will describe how to configure MPLS QoS Pipe model.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Set MPLS LSP Model

Interface configuration for PE1:

```
PE1(config)# mpls lsp-model pipe exp 7
```

Interface configuration for P:

```
P(config)# mpls lsp-model pipe
```

Interface configuration for PE2:

```
PE2(config)# mpls lsp-model pipe exp 7
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 10.0.9.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface eth-0-1
PE1(config-if)# no switchport
PE1(config-if)# qos domain type dscp 1
PE1(config-if)# trust dscp
PE1(config-if)# ip address 1.1.1.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for P:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 10.0.9.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 10.0.17.2/24
P(config-if)# label-switching
P(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-17
PE2(config-if)# no switchport
PE2(config-if)# ip address 10.0.17.1/24
PE2(config-if)# label-switching
PE2(config-if)# exit
PE2(config)# interface eth-0-1
PE2(config-if)# no switchport
PE2(config-if)# qos domain type dscp 1
PE2(config-if)# trust dscp
PE2(config-if)# ip address 2.2.2.2/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 4 Configure static arp

Interface configuration for PE1:

```
PE1(config)# arp 1.1.1.2 0001.0001.0002
```

Interface configuration for PE2:

```
PE2(config)# arp 2.2.2.1 0002.0002.0001
```

Step 5 Configure static ftn/ilm

Static ftn for PE1:

```
PE1(config)# mpls ftn-entry 2.2.2.0/24 102 10.0.9.2
PE1(config)# mpls ilm-entry pop 201
```

Static ilm for P:

```
P(config)# mpls ilm-entry swap 102 10.0.17.1 203
P(config)# mpls ilm-entry swap 302 10.0.9.1 201
```

Static ilm for PE2:

```
PE2(config)# mpls ftn-entry 1.1.1.0/24 302 10.0.17.2
PE2(config)# mpls ilm-entry pop 203
```

Step 6 Validation

Display the result on PE1:

```
PE1# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN

Code   FEC             Out-Label   Nexthop       Out-Intf
K>     2.2.2.0/24       102        10.0.9.2      eth-0-9

PE1# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code    FEC            I/O Label        Nexthop       Out-Intf
K>      0.0.0.0/0      201/-            0.0.0.0       N/A
```

Display the result on P:

```
P# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code    FEC            I/O Label        Nexthop       Out-Intf
K>      0.0.0.0/0      102/203          10.0.17.1     eth-0-17
K>      0.0.0.0/0      302/201          10.0.9.1      eth-0-9
```

Display the result on PE2:

```
PE2# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN

Code   FEC             Out-Label   Nexthop       Out-Intf
K>     1.1.1.0/24       302        10.0.17.2     eth-0-17

PE2# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM
```

```
Code   FEC          I/O Label      Nexthop      Out-Intf
K>     0.0.0.0/0    203/-          0.0.0.0      N/A
```

MPLS QoS Short Pipe Configuration

The following example will describe how to configure MPLS QoS Short Pipe model.

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Set MPLS LSP Model

Interface configuration for PE1:

```
PE1(config)# mpls lsp-model short-pipe exp 7
```

Interface configuration for P:

```
P(config)# mpls lsp-model short-pipe
```

Interface configuration for PE2:

```
PE2(config)# mpls lsp-model short-pipe exp 7
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for PE1:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip address 10.0.9.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
PE1(config)# interface eth-0-1
PE1(config-if)# no switchport
PE1(config-if)# qos domain type dscp 1
PE1(config-if)# trust dscp
PE1(config-if)# ip address 1.1.1.1/24
PE1(config-if)# label-switching
PE1(config-if)# exit
```

Interface configuration for P:

```
P(config)# interface eth-0-9
P(config-if)# no switchport
P(config-if)# ip address 10.0.9.2/24
P(config-if)# label-switching
P(config-if)# exit
P(config)# interface eth-0-17
P(config-if)# no switchport
P(config-if)# ip address 10.0.17.2/24
P(config-if)# label-switching
P(config-if)# exit
```

Interface configuration for PE2:

```
PE2(config)# interface eth-0-17
PE2(config-if)# no switchport
PE2(config-if)# ip address 10.0.17.1/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

```
PE2(config)# interface eth-0-1
PE2(config-if)# no switchport
PE2(config-if)# qos domain type dscp 1
PE2(config-if)# trust dscp
PE2(config-if)# ip address 2.2.2.2/24
PE2(config-if)# label-switching
PE2(config-if)# exit
```

Step 4 Configure static arp

Interface configuration for PE1:

```
PE1(config)# arp 1.1.1.2 0001.0001.0002
```

Interface configuration for PE2:

```
PE2(config)# arp 2.2.2.1 0002.0002.0001
```

Step 5 Configure static ftn/ilm

Static ftn for PE1:

```
PE1(config)# mpls ftn-entry 2.2.2.0/24 102 10.0.9.2
PE1(config)# mpls ilm-entry pop 201
```

Static ilm for P:

```
P(config)# mpls ilm-entry swap 102 10.0.17.1 203
P(config)# mpls ilm-entry swap 302 10.0.9.1 201
```

Static ilm for PE2:

```
PE2(config)# mpls ftn-entry 1.1.1.0/24 302 10.0.17.2
PE2(config)# mpls ilm-entry pop 203
```

Step 6 Validation

Display the result on PE1:

```
PE1# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN

Code   FEC            Out-Label   Nexthop       Out-Intf
K>     2.2.2.0/24       102       10.0.9.2      eth-0-9

PE1# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC            I/O Label        Nexthop       Out-Intf
K>     0.0.0.0/0       201/-           0.0.0.0       N/A
```

Display the result on P:

```
P# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC            I/O Label        Nexthop       Out-Intf
```

```
K>     0.0.0.0/0       102/203          10.0.17.1     eth-0-17
K>     0.0.0.0/0       302/201          10.0.9.1      eth-0-9
```

Display the result on PE2:

```
PE2# show mpls ftn-database
Codes: > - selected FTN, p - stale FTN, B - BGP FTN, K - CLI FTN,
  L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
  * -bypass FTN, U - unknown FTN

Code   FEC             Out-Label   Nexthop       Out-Intf
K>     1.1.1.0/24       302        10.0.17.2     eth-0-17

PE2# show mpls ilm-database
Codes: > - selected ILM, * - LSP ILM, p - stale ILM, B - BGP ILM, K - CLI ILM,
  L - LDP ILM, R - RSVP-TE ILM, S - SNMP ILM, I - IGP-Shortcut
  U - unknown ILM

Code   FEC             I/O Label        Nexthop       Out-Intf
K>     0.0.0.0/0       203/-            0.0.0.0       N/A
```

21.6 Configuring L3VPN

21.6.1 Overview

Brief Introduction

This chapter describes how to configure L3VPN. It uses Route Target's community to control route sending and receiving. RD is used to distinguish which VPN the route from. The inner label is uesd to map the different vrf, then through the vrf to guide packet forwarding.

21.6.2 Configuration Examples

Topology

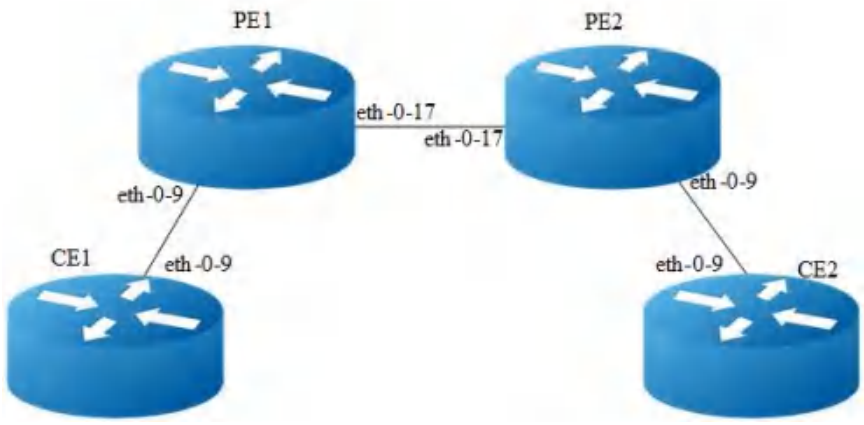


Figure 21-9 L3VPN model

Configuration Steps

The following example will describe how to configure L3VPN:

The following configuration should be operated on all switches if the switch ID is not specified.

Step 1 Enter the configure mode

```
Switch# configure terminal
```

Step 2 Set vrf

Vrf configuration for PE1:

```
PE1(config)# ip vrf vpn1
PE1(config-vrf)# rd 1:1
PE1(config-vrf)# route-target both 1:1
PE1(config-vrf)# exit
```

Vrf configuration for PE2:

```
PE2(config)# ip vrf vpn1
PE2(config-vrf)# rd 1:1
PE2(config-vrf)# route-target both 1:1
PE2(config-vrf)# exit
```

Step 3 Enter the interface configure mode and set the attributes of the interface

Interface configuration for CE1:

```
CE1(config)# interface eth-0-9
CE1(config-if)# no switchport
CE1(config-if)# ip address 2.2.2.1/24
CE1(config-if)# exit
CE1(config)# interface loopback 0
CE1(config-if)# ip address 4.4.4.4/32
CE1(config-if)# exit
```

Interface configuration for PE1, eth-0-9 need enable ldp and join vrf:

```
PE1(config)# interface eth-0-9
PE1(config-if)# no switchport
PE1(config-if)# ip vrf forwarding vpn1
PE1(config-if)# ip address 2.2.2.2/24
PE1(config-if)# exit
PE1(config)# interface eth-0-17
PE1(config-if)# no switchport
PE1(config-if)# label-switching
PE1(config-if)# ip address 1.1.1.1/24
PE1(config-if)# enable-ldp
PE1(config-if)# exit
PE1(config)# interface loopback 0
PE1(config-if)# ip address 5.5.5.5/32
PE1(config-if)# exit
```

Interface configuration for PE2, eth-0-9 need enable ldp and join vrf:

```
PE2(config)# interface eth-0-9
PE2(config-if)# no switchport
PE2(config-if)# ip vrf forwarding vpn1
PE2(config-if)# ip address 3.3.3.3/24
PE2(config-if)# exit
PE2(config)# interface eth-0-17
PE2(config-if)# no switchport
PE2(config-if)# label-switching
PE2(config-if)# ip address 1.1.1.2/24
PE2(config-if)# enable-ldp
PE2(config-if)# exit
PE2(config)# interface loopback 0
PE2(config-if)# ip address 6.6.6.6/32
PE2(config-if)# exit
```

Interface configuration for CE2:

```
CE2(config)# interface eth-0-9
CE2(config-if)# no switchport
CE2(config-if)# ip address 3.3.3.4/24
CE2(config-if)# exit
CE2(config)# interface loopback0
CE2(config-if)# ip address 7.7.7.7/32
CE2(config-if)# exit
```

Step 4 Enable router ldp

LDP configuration for PE1:

```
PE1(config)# router ldp
PE1(config-router)# router-id 5.5.5.5
PE1(config-router)# exit
```

LDP configuration for PE2:

```
PE2(config)# router ldp
PE2(config-router)# router-id 6.6.6.6
PE2(config-router)# exit
```

Step 5 Enable router rip

RIP configuration for CE1:

```
CE1(config)# router rip
CE1(config-router)# network 2.2.2.2/24
CE1(config-router)# redistribute connected
CE1(config-router)# exit
```

RIP configuration for PE1:

```
PE1(config)# router rip
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# network 2.2.2.0/24
PE1(config-router-af)# redistribute bgp
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

RIP configuration for PE2:

```
PE2(config)# router rip
PE2(config-router)# address-family ipv4 vrf vpn1
PE2(config-router-af)# network 3.3.3.3/24
PE2(config-router-af)# redistribute bgp
PE2(config-router-af)# exit-address-family
PE2(config-router)# exit
```

RIP configuration for CE2:

```
CE2(config)# router rip
CE2(config-router)# network 3.3.3.0/24
CE2(config-router)# redistribute connected
CE2(config-router)# exit
```

Step 6 Enable router ospf

OSPF configuration for PE1:

```
PE1(config)#router ospf
PE1(config-router)# redistribute connected
```

```
PE1(config-router)# network 1.1.1.0/24 area 0
PE1(config-router)# exit
```

OSPF configuration for PE2:

```
PE2(config)# router ospf
PE2(config-router)# redistribute connected
PE2(config-router)# network 1.1.1.0/24 area 0
PE2(config-router)# exit
```

Step 7 Enable router bgp

BGP configuration for PE1:

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 6.6.6.6 remote-as 1
PE1(config-router)# neighbor 6.6.6.6 update-source loopback0
PE1(config-router)# address-family ipv4
PE1(config-router-af)# no synchronization
PE1(config-router-af)# neighbor 6.6.6.6 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family vpnv4 unicast
PE1(config-router-af)# no synchronization
PE1(config-router-af)# neighbor 6.6.6.6 activate
PE1(config-router-af)# neighbor 6.6.6.6 send-community both
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# redistribute connected
PE1(config-router-af)# redistribute rip
PE1(config-router-af)# no synchronization
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

BGP configuration for PE2:

```
PE2(config)# router bgp 1
PE2(config-router)# neighbor 5.5.5.5 remote-as 1
PE2(config-router)# neighbor 5.5.5.5 update-source loopback0
PE2(config-router)# address-family ipv4
PE2(config-router-af)# no synchronization
PE2(config-router-af)# neighbor 5.5.5.5 activate
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family vpnv4 unicast
PE2(config-router-af)# no synchronization
PE2(config-router-af)# neighbor 5.5.5.5 activate
PE2(config-router-af)# neighbor 5.5.5.5 send-community both
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family ipv4 vrf vpn1
PE2(config-router-af)# redistribute connected
PE2(config-router-af)# redistribute rip
PE2(config-router-af)# no synchronization
PE2(config-router-af)# exit-address-family
PE2(config-router)# exit
```

Step 8 Validation

Use show ip route command and ping CE2 loopback address to validate the l3vpn is worked.

Display the result on PE1:

```
PE1# show ip route
s: K - kernel, C - connected, S - static, R - RIP, B - BGP
  O - OSPF, IA - OSPF inter area
  N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
  E1 - OSPF external type 1, E2 - OSPF external type 2
  i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
  Dc - DHCP Client
  [*] - [AD/Metric]
  * - candidate default
C       2.2.2.0/24 is directly connected, eth-0-9
C       2.2.2.1/32 is in local loopback, eth-0-9
R       3.3.3.0/24 [120/2] via 2.2.2.2, eth-0-9, 00:00:04
C       4.4.4.4/32 is directly connected, loopback0
R       7.7.7.7/32 [120/2] via 2.2.2.2, eth-0-9, 00:00:02

PE1# ping 7.7.7.7
PING 7.7.7.7 (7.7.7.7) 56(84) bytes of data.
64 bytes from 7.7.7.7: icmp_seq=0 ttl=62 time=1828 ms
64 bytes from 7.7.7.7: icmp_seq=1 ttl=62 time=1801 ms
64 bytes from 7.7.7.7: icmp_seq=2 ttl=62 time=1775 ms
64 bytes from 7.7.7.7: icmp_seq=3 ttl=62 time=1775 ms
64 bytes from 7.7.7.7: icmp_seq=4 ttl=62 time=1705 ms

--- 7.7.7.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4018ms
rtt min/avg/max/mdev = 1705.600/1777.267/1828.148/40.840 ms, pipe 3
```