

目录

目录.....	1
第 1 章 基本管理操作.....	1-1
1.1 交换机管理.....	1-1
1.1.1 管理方式.....	1-1
1.1.2 CLI 界面.....	1-10
1.2 交换机基本配置.....	1-14
1.2.1 基本配置.....	1-14
1.2.2 远程管理.....	1-16
1.2.3 配置交换机的 IP 地址.....	1-19
1.2.4 SNMP 配置.....	1-21
1.2.5 交换机升级.....	1-29
1.3 文件系统.....	1-42
1.3.1 文件存储设备介绍.....	1-42
1.3.2 文件系统操作任务配置序列.....	1-42
1.3.3 典型应用.....	1-44
1.3.4 排错帮助.....	1-44
1.4 集群网管.....	1-44
1.4.1 集群网管介绍.....	1-44
1.4.2 集群网管基本配置.....	1-45
1.4.3 集群网管举例.....	1-48
1.4.4 集群网管排错帮助.....	1-48
1.5 设备管理.....	1-49
1.5.1 设备管理简介.....	1-49
1.5.2 设备管理配置.....	1-49
1.5.3 设备管理排错帮助.....	1-51
1.5.4 板卡带电插拔操作.....	1-51
第 2 章 二层业务操作.....	2-1
2.1 端口配置.....	2-1
2.1.1 业务端口配置.....	2-1
2.1.2 端口配置举例.....	2-3
2.1.3 端口排错帮助.....	2-5

2.2 端口环路检测	2-5
2.2.1 端口环路检测功能简介	2-5
2.2.2 端口环路检测功能配置任务序列.....	2-5
2.2.3 端口环路检测典型案例	2-7
2.2.4 端口环路检测排错帮助	2-7
2.3 ULDP	2-8
2.3.1 ULDP 功能简介	2-8
2.3.2 ULDP 配置任务序列.....	2-9
2.3.3 ULDP 功能典型案例.....	2-11
2.3.4 ULDP 排错帮助	2-13
2.4 LLDP	2-14
2.4.1 LLDP 功能简介	2-14
2.4.2 LLDP 功能配置任务序列.....	2-14
2.4.3 LLDP 功能典型案例	2-17
2.4.4 LLDP 功能排错帮助	2-18
2.5 Port Channel	2-18
2.5.1 Port Channel 介绍.....	2-18
2.5.2 LACP 简介	2-20
2.5.3 Port Channel 配置任务序列	2-21
2.5.4 Port Channel 举例.....	2-22
2.5.5 Port Channel 排错帮助	2-25
2.6 MTU.....	2-25
2.6.1 MTU 介绍	2-25
2.6.2 MTU 配置任务序列.....	2-25
2.7 EFM OAM.....	2-26
2.7.1 EFM OAM 介绍	2-26
2.7.2 EFM OAM 基本配置.....	2-29
2.7.3 EFM OAM 举例	2-32
2.7.4 EFM OAM 排错帮助.....	2-33
2.8 DDM	2-33
2.8.1 DDM 介绍	2-33
2.8.2 DDM 配置任务序列	2-35
2.8.3 DDM 举例	2-37
2.8.4 DDM 排错帮助.....	2-41
2.9 LLDP-MED	2-41
2.9.1 LLDP-MED 介绍	2-41
2.9.2 LLDP-MED 配置任务序列.....	2-42
2.9.3 LLDP-MED 举例	2-44

2.9.4 LLDP-MED 排错帮助	2-47
2.10 bpdu-tunnel	2-47
2.10.1 bpdu-tunnel 介绍	2-47
2.10.2 bpdu-tunnel 配置任务序列	2-48
2.10.3 bpdu-tunnel 举例	2-49
2.10.4 bpdu-tunnel 排错帮助	2-50
2.11 PORT SECURITY	2-50
2.11.1 PORT SECURITY 介绍	2-50
2.11.2 PORT SECURITY 配置任务序列	2-50
2.11.3 PORT SECURITY 举例	2-51
2.11.4 PORT SECURITY 排错帮助	2-52
2.12 CFM-OAM	2-52
2.12.1 概述	2-52
2.12.2 CFM OAM 基本概念	2-53
2.12.3 CFM OAM 功能介绍	2-55
2.12.4 CFM OAM 基础功能配置	2-58
2.12.5 CFM OAM 故障确认	2-63
2.12.6 CFM OAM 故障定位	2-64
2.12.7 ULPP 联动（可选）	2-65
2.12.8 配置应用举例	2-68
2.12.9 CFM 排错帮助	2-71
2.13 QSFP+端口拆分与合并	2-72
2.13.1 QSFP+端口拆分与合并操作功能介绍	2-72
2.13.2 QSFP+端口配置	2-73
2.13.3 QSFP+端口配置典型案例	2-73
2.13.4 QSFP+端口配置排错帮助	2-74
2.14 VLAN	2-74
2.14.1 VLAN 介绍	2-74
2.14.2 VLAN 的配置任务序列	2-75
2.14.3 VLAN 典型应用	2-78
2.14.4 Hybrid 端口的典型应用	2-79
2.15 GVRP	2-81
2.15.1 GVRP 介绍	2-81
2.15.2 GVRP 配置任务序列	2-82
2.15.3 GVRP 举例	2-83
2.15.4 GVRP 排错帮助	2-85
2.16 Dot1q-tunnel	2-85
2.16.1 Dot1q-tunnel 介绍	2-85

2.16.2 Dot1q-tunnel 配置	2-86
2.16.3 Dot1q-tunnel 典型应用	2-87
2.16.4 Dot1q-tunnel 排错帮助	2-88
2.17 VLAN-translation	2-88
2.17.1 VLAN-translation 介绍	2-88
2.17.2 VLAN-translation 配置	2-88
2.17.3 VLAN-translation 典型应用	2-89
2.17.4 VLAN-translation 排错帮助	2-90
2.18 动态 VLAN	2-90
2.18.1 动态 VLAN 介绍	2-90
2.18.2 动态 VLAN 配置	2-91
2.18.3 动态 VLAN 典型应用	2-93
2.18.4 动态 VLAN 排错帮助	2-94
2.19 Voice VLAN	2-94
2.19.1 Voice VLAN 介绍	2-94
2.19.2 Voice VLAN 配置	2-95
2.19.3 Voice VLAN 典型应用	2-96
2.19.4 Voice VLAN 排错帮助	2-97
2.20 Super VLAN	2-97
2.20.1 Super VLAN 简介	2-97
2.20.2 Super VLAN 配置任务序列	2-99
2.20.3 Super VLAN 典型应用	2-100
2.20.4 Super VLAN 排错帮助	2-101
2.21 MAC 地址表	2-102
2.21.1 MAC 地址表介绍	2-102
2.21.2 MAC 地址表配置	2-104
2.21.3 典型配置举例	2-106
2.21.4 排错帮助	2-107
2.22 MAC Notification	2-107
2.22.1 MAC Notification 介绍	2-107
2.22.2 MAC Notification 配置	2-107
2.22.3 MAC Notification 举例	2-109
2.22.4 MAC Notification 排错帮助	2-109
第 3 章 IP 业务操作	3-1
3.1 三层接口	3-1
3.1.1 三层接口介绍	3-1
3.1.2 三层接口配置	3-1

3.2 网管端口	3-3
3.2.1 IP 网管端口介绍	3-3
3.2.2 IP 网管端口配置	3-3
3.3 IP 配置	3-4
3.3.1 IPv4、IPv6 介绍	3-4
3.3.2 IP 配置	3-5
3.3.3 IP 配置举例	3-11
3.3.4 IPv6 排错帮助	3-16
3.4 IP 转发	3-16
3.4.1 IP 转发介绍	3-16
3.4.2 IP 路由聚合配置	3-16
3.5 URPF	3-17
3.5.1 URPF 介绍	3-17
3.5.2 URPF 配置任务序列	3-18
3.5.3 URPF 典型案例	3-19
3.5.4 URPF 排错帮助	3-19
3.6 ARP	3-20
3.6.1 ARP 介绍	3-20
3.6.2 ARP 配置	3-20
3.6.3 ARP 转发排错帮助	3-21
3.7 I3 station movement	3-21
3.7.1 I3 station movement 介绍	3-21
3.7.2 I3 station movement 配置	3-21
3.8 防 ARP 扫描	3-22
3.8.1 防 ARP 扫描功能介绍	3-22
3.8.2 防 ARP 扫描配置任务序列	3-22
3.8.3 防 ARP 扫描典型案例	3-25
3.8.4 防 ARP 扫描排错帮助	3-26
3.9 ARP 绑定	3-26
3.9.1 概述	3-26
3.9.2 ARP 绑定配置	3-27
3.9.3 ARP 绑定举例	3-28
3.10 ARP GUARD	3-29
3.10.1 ARP GUARD 的介绍	3-29
3.10.2 ARP GUARD 配置任务序列	3-30
3.11 Arp local proxy	3-30

3.11.1 Arp local proxy 功能简介	3-30
3.11.2 Arp local proxy 功能配置任务序列	3-31
3.11.3 Arp local proxy 功能典型案例	3-32
3.11.4 Arp local proxy 功能排错帮助	3-32
3.12 免费 ARP 发送	3-33
3.12.1 免费 ARP 发送功能简介	3-33
3.12.2 免费 ARP 发送功能配置任务序列	3-33
3.12.3 免费 ARP 发送功能典型案例	3-34
3.12.4 免费 ARP 发送功能排错帮助	3-34
3.13 Keepalive gateway	3-35
3.13.1 keepalive gateway 介绍	3-35
3.13.2 keepalive gateway 功能配置任务序列	3-35
3.13.3 keepalive gateway 举例	3-36
3.13.4 keepalive gateway 排错帮助	3-36
3.14 UFT	3-37
3.14.1 UFT 功能介绍	3-37
3.14.2 UFT 配置任务序列	3-37
3.14.3 UFT 典型案例	3-38
3.14.4 UFT 排错帮助	3-39
3.15 DHCP	3-39
3.15.1 DHCP 介绍	3-39
3.15.2 DHCP 服务器配置	3-40
3.15.3 DHCP 中继配置	3-43
3.15.4 DHCP 配置举例	3-44
3.15.5 DHCP 排错帮助	3-46
3.16 DHCP option 82	3-47
3.16.1 DHCP option 82 介绍	3-47
3.16.2 DHCP option 82 的配置任务序列	3-49
3.16.3 DHCP option 82 应用举例	3-52
3.16.4 DHCP option 82 排错帮助	3-54
3.17 DHCP Snooping	3-54
3.17.1 DHCP Snooping 介绍	3-54
3.17.2 DHCP Snooping 的配置任务序列	3-55
3.17.3 DHCP Snooping 典型应用	3-59
3.17.4 DHCP Snooping 排错帮助	3-60
3.18 DHCP option 60 和 option 43	3-61
3.18.1 DHCP option 60 和 option 43 介绍	3-61
3.18.2 DHCP option 60 和 option 43 配置任务序列	3-61

3.18.3 DHCP option 60 和 option 43 举例	3-62
3.18.4 DHCP option 60 和 option 43 排错帮助	3-62
第 4 章 路由协议相关操作	4-1
4.1 路由协议概述	4-1
4.1.1 路由表	4-1
4.1.2 IP 路由策略	4-2
4.2 静态路由	4-8
4.2.1 静态路由介绍	4-8
4.2.2 缺省路由介绍	4-8
4.2.3 静态路由配置	4-9
4.2.4 配置案例	4-9
4.3 RIP	4-10
4.3.1 RIP 介绍	4-10
4.3.2 RIP 配置	4-12
4.3.3 RIP 案例	4-18
4.3.4 RIP 排错帮助	4-23
4.4 OSPF	4-23
4.4.1 OSPF 介绍	4-23
4.4.2 OSPF 配置	4-26
4.4.3 OSPF 案例	4-31
4.4.4 OSPF 排错帮助	4-40
4.5 BGP	4-41
4.5.1 BGP 介绍	4-41
4.5.2 BGP 配置	4-44
4.5.3 BGP 典型案例	4-55
4.5.4 BGP 排错帮助	4-68
4.6 IPv4 黑洞路由	4-68
4.6.1 黑洞路由介绍	4-68
4.6.2 IPv4 黑洞路由配置任务	4-68
4.6.3 黑洞路由举例	4-69
4.6.4 黑洞路由排错帮助	4-70
4.7 GRE	4-70
4.7.1 GRE 隧道介绍	4-70
4.7.2 GRE 隧道基本配置	4-70
4.7.3 GRE 隧道举例	4-72
4.7.4 GRE 隧道引用业务环回组举例	4-75
4.7.5 GRE 隧道排错帮助	4-79

4.8 ECMP	4-79
4.8.1 ECMP 简介	4-79
4.8.2 ECMP 配置任务序列	4-80
4.8.3 ECMP 典型案例	4-80
4.9 BFD	4-83
4.9.1 BFD 介绍	4-83
4.9.2 BFD 配置任务序列	4-84
4.9.3 BFD 举例	4-86
4.9.4 BFD 排错帮助	4-89
4.10 BGP GR	4-89
4.10.1 GR 简介	4-89
4.10.2 GR 配置任务序列	4-91
4.10.3 GR 典型案例	4-93
4.11 OSPF GR	4-94
4.11.1 OSPF GR 介绍	4-94
4.11.2 OSPF GR 基本配置	4-95
4.11.3 OSPF GR 举例	4-96
4.11.4 OSPF GR 排错帮助	4-97
 第 5 章 IPv4 组播协议	 5-1
5.1 组播	5-1
5.1.1 组播简介	5-1
5.1.2 组播地址	5-1
5.1.3 IP 组播报文转发	5-2
5.1.4 IP 组播应用	5-3
5.2 PIM-DM	5-3
5.2.1 PIM-DM 介绍	5-3
5.2.2 PIM-DM 配置任务序列	5-4
5.2.3 PIM-DM 典型案例	5-6
5.2.4 PIM-DM 排错帮助	5-7
5.3 PIM-SM	5-8
5.3.1 PIM-SM 介绍	5-8
5.3.2 PIM-SM 配置任务序列	5-9
5.3.3 PIM-SM 典型案例	5-12
5.3.4 PIM-SM 排错帮助	5-14
5.4 MSDP	5-15
5.4.1 MSDP 介绍	5-15

5.4.2 MSDP 配置任务简介	5-16
5.4.3 配置 MSDP 基本功能	5-16
5.4.4 配置 MSDP 对等体	5-17
5.4.5 配置报文收发	5-18
5.4.6 配置 SA-cache 参数	5-19
5.4.7 MSDP 举例	5-19
5.4.8 MSDP 排错帮助	5-25
5.5 ANYCAST RP	5-25
5.5.1 ANYCAST RP 介绍	5-25
5.5.2 ANYCAST RP 配置任务	5-26
5.5.3 ANYCAST RP 典型案例	5-28
5.5.4 ANYCAST RP 排错帮助	5-29
5.6 PIM-SSM	5-29
5.6.1 PIM-SSM 介绍	5-29
5.6.2 PIM-SSM 配置任务序列	5-30
5.6.3 PIM-SSM 典型案例	5-30
5.6.4 PIM-SSM 排错帮助	5-32
5.7 DVMRP	5-32
5.7.1 DVMRP 介绍	5-32
5.7.2 配置任务序列	5-33
5.7.3 DVMRP 典型案例	5-35
5.7.4 DVMRP 排错帮助	5-36
5.8 DCSCM	5-36
5.8.1 DCSCM 介绍	5-36
5.8.2 DCSCM 配置任务序列	5-37
5.8.3 DCSCM 典型案例	5-40
5.8.4 DCSCM 排错帮助	5-41
5.9 IGMP	5-42
5.9.1 IGMP 介绍	5-42
5.9.2 配置任务序列	5-43
5.9.3 IGMP 典型案例	5-45
5.9.4 IGMP 排错帮助	5-46
5.10 IGMP Snooping	5-46
5.10.1 IGMP Snooping 介绍	5-46
5.10.2 IGMP Snooping 配置任务	5-47
5.10.3 IGMP Snooping 典型案例	5-49
5.10.4 IGMP Snooping 排错帮助	5-52
5.11 IGMP Proxy	5-53

5.11.1 IGMP Proxy 介绍	5-53
5.11.2 IGMP Proxy 配置任务	5-53
5.11.3 IGMP Proxy 举例	5-54
5.11.4 IGMP Proxy 排错帮助	5-57
5.12 组播 VLAN	5-57
5.12.1 组播 VLAN 介绍	5-57
5.12.2 组播 VLAN 配置任务	5-57
5.12.3 组播 VLAN 举例	5-58
第 6 章 安全功能操作	6-1
6.1 ACL	6-1
6.1.1 ACL 介绍	6-1
6.1.2 ACL 配置	6-2
6.1.3 ACL 举例	6-19
6.1.4 ACL 排错帮助	6-23
6.2 802.1x	6-24
6.2.1 802.1x 介绍	6-24
6.2.2 802.1x 配置	6-35
6.2.3 802.1x 应用举例	6-37
6.2.4 802.1x 排错帮助	6-42
6.3 端口、VLAN 中 MAC、IP 数量限制	6-43
6.3.1 端口、VLAN 中 MAC、IP 数量限制功能简介	6-43
6.3.2 端口、VLAN 中 MAC、IP 数量限制功能配置任务序列	6-44
6.3.3 端口、VLAN 中 MAC、IP 数量限制功能典型案例	6-46
6.3.4 端口、VLAN 中 MAC、IP 数量限制功能排错帮助	6-47
6.4 AM	6-47
6.4.1 AM 功能简介	6-47
6.4.2 AM 功能配置任务序列	6-48
6.4.3 AM 功能典型案例	6-49
6.4.4 AM 功能排错帮助	6-50
6.5 安全特性	6-50
6.5.1 安全特性介绍	6-50
6.5.2 安全特性配置	6-50
6.5.3 安全特性举例	6-52
6.6 TACACS+	6-52
6.6.1 TACACS+简介	6-52
6.6.2 TACACS+配置	6-53
6.6.3 TACACS+典型案例	6-53

6.6.4 TACACS+排错帮助	6-54
6.7 RADIUS	6-54
6.7.1 RADIUS 简介	6-54
6.7.2 RADIUS 配置	6-57
6.7.3 RADIUS 典型案例	6-58
6.7.4 RADIUS 排错帮助	6-60
6.8 SSL.....	6-60
6.8.1 SSL 简介	6-60
6.8.2 SSL 配置任务序列.....	6-62
6.8.3 SSL 典型案例	6-63
6.8.4 SSL 排错帮助	6-63
6.9 VLAN-ACL	6-64
6.9.1 VLAN-ACL 功能简介	6-64
6.9.2 VLAN-ACL 功能配置任务序列.....	6-64
6.9.3 VLAN-ACL 功能配置案例	6-66
6.9.4 VLAN-ACL 排错帮助	6-67
6.10 PPPoE 中间代理.....	6-67
6.10.1 PPPoE Intermediate Agent 介绍	6-67
6.10.2 PPPoE Intermediate Agent 配置任务序列.....	6-72
6.10.3 PPPoE Intermediate Agent 典型应用	6-73
6.10.4 PPPoE Intermediate Agent 排错帮助	6-75
6.11 EIFP	6-75
6.11.1 EIFP 介绍	6-75
6.11.2 EIFP 配置	6-77
6.11.3 EIFP 举例	6-85
6.11.4 EIFP 排错帮助	6-89
6.12 Captive Portal 认证	6-89
6.12.1 Captive Portal 认证配置	6-89
6.12.2 计费功能配置.....	6-94
6.12.3 Free-resource 功能配置.....	6-97
6.12.4 认证白名单功能配置.....	6-98
6.12.5 认证成功后页面自动推送（暂不支持）	6-99
6.12.6 http-redirect-filter	6-102
6.12.7 Portal 无感知	6-104
6.12.8 Portal 逃生	6-107
6.13 QoS	6-113
6.13.1 QoS	6-113
6.13.2 PBR.....	6-127

6.13.3 IPv6 PBR	6-130
6.14 MPLS QoS	6-133
6.14.1 MPLS QoS 介绍.....	6-133
6.14.2 MPLS QoS 配置.....	6-134
6.14.3 MPLS QoS 举例.....	6-135
6.14.4 MPLS QoS 排错帮助	6-137
6.15 基于流的重定向.....	6-137
6.15.1 基于流的重定向介绍.....	6-137
6.15.2 基于流的重定向配置任务序列	6-138
6.15.3 基于流的重定向举例.....	6-138
6.15.4 基于流的重定向排错帮助	6-138
6.16 Egress QoS	6-139
6.16.1 Egress QoS 介绍.....	6-139
6.16.2 Egress QoS 配置.....	6-140
6.16.3 Egress QoS 举例.....	6-143
6.16.4 Egress QoS 排错帮助	6-144
6.17 灵活 QinQ 操作.....	6-145
6.17.1 灵活 QinQ 功能介绍	6-145
6.17.2 灵活 QinQ 配置	6-145
6.17.3 灵活 QinQ 举例	6-147
6.17.4 灵活 QinQ 排错帮助	6-149
6.18 Vlan-shaping	6-149
6.18.1 Vlan-shaping 介绍	6-149
6.18.2 Vlan-shaping 配置	6-152
6.18.3 Vlan-shaping 举例	6-155
6.18.4 Vlan-shaping 排错帮助	6-159
第 7 章 可靠性操作	7-1
7.1 MSTP.....	7-1
7.1.1 MSTP 介绍	7-1
7.1.2 MSTP 配置	7-3
7.1.3 MSTP 举例	7-7
7.1.4 MSTP 排错帮助	7-12
7.2 ERPS.....	7-12
7.2.1 ERPS 介绍	7-12
7.2.2 ERPS 配置	7-18
7.2.3 ERPS 举例	7-20
7.2.4 ERPS 排错帮助	7-26

7.3 VRRP	7-27
7.3.1 VRRP 简介	7-27
7.3.2 VRRP 配置	7-27
7.3.3 VRRP 典型案例	7-29
7.3.4 VRRP 排错帮助	7-30
7.4 MRPP	7-30
7.4.1 MRPP 简介	7-30
7.4.2 MRPP 配置任务序列	7-33
7.4.3 MRPP 典型案例	7-35
7.4.4 MRPP 排错帮助	7-37
7.5 ULPP	7-37
7.5.1 ULPP 简介	7-37
7.5.2 ULPP 配置任务序列	7-39
7.5.3 ULPP 典型案例	7-41
7.5.4 ULPP 排错帮助	7-44
7.6 ULSM	7-44
7.6.1 ULSM 简介	7-44
7.6.2 ULSM 配置任务序列	7-45
7.6.3 ULSM 典型案例	7-46
7.6.4 ULSM 排错帮助	7-47
7.7 HA	7-48
7.7.1 HA 简介	7-48
7.7.2 HA 配置任务序列	7-49
7.7.3 HA 典型案例	7-50
第 8 章 维护及调试操作	8-1
8.1 维护和调试	8-1
8.1.1 Ping	8-1
8.1.2 Ping6	8-1
8.1.3 Traceroute	8-1
8.1.4 Traceroute6	8-1
8.1.5 Show	8-2
8.1.6 Debug	8-3
8.2 Info-center	8-3
8.2.1 基本概念	8-3
8.2.2 信息中心功能配置任务序列	8-6
8.2.3 信息中心配置举例	8-9
8.2.4 信息中心功能排错帮助	8-11

8.3 定时重启交换机	8-12
8.3.1 定时重启交换机简介	8-12
8.3.2 定时重启交换机任务序列	8-12
8.4 CPU 收发报文调试	8-12
8.4.1 CPU 收发报文简介	8-12
8.4.2 CPU 收发报文任务序列	8-12
8.5 COPP	8-13
8.5.1 COPP 功能简介	8-13
8.5.2 COPP 功能任务序列	8-13
8.5.3 COPP 功能配置举例	8-16
8.5.4 COPP 功能排错帮助	8-18
8.6 镜像	8-18
8.6.1 镜像介绍	8-18
8.6.2 镜像配置任务序列	8-18
8.6.3 镜像举例	8-19
8.6.4 镜像排错帮助	8-20
8.7 RSPAN	8-20
8.7.1 RSPAN 简介	8-20
8.7.2 RSPAN 配置任务序列	8-22
8.7.3 RSPAN 典型案例	8-23
8.7.4 RSPAN 排错帮助	8-26
8.8 sFlow	8-26
8.8.1 sFlow 介绍	8-26
8.8.2 sFlow 配置任务序列	8-27
8.8.3 sFlow 举例	8-29
8.8.4 sFlow 排错帮助	8-29
8.9 IPFIX	8-30
8.9.1 IPFIX 介绍	8-30
8.9.2 IPFIX 基本配置	8-30
8.9.3 IPFIX 举例	8-34
8.9.4 IPFIX 排错帮助	8-36
8.10 ERSPAN	8-36
8.10.1 ERSPAN 简介	8-36
8.10.2 ERSPAN 配置任务序列	8-36
8.10.3 ERSPAN 典型案例	8-37
8.10.4 ERSPAN 排错帮助	8-39
第 9 章 网络时间及域名管理操作	9-1

9.1 SNTP	9-1
9.1.1 SNTP 简介	9-1
9.1.2 SNTP 典型配置举例	9-2
9.2 NTP	9-2
9.2.1 NTP 功能简介	9-2
9.2.2 NTP 功能配置任务序列	9-3
9.2.3 NTP 功能典型案例	9-5
9.2.4 NTP 功能排错帮助	9-6
9.3 DNSv4/v6	9-6
9.3.1 DNS 简介	9-6
9.3.2 DNSv4/v6 配置任务序列	9-7
9.3.3 DNS 典型案例	9-9
9.3.4 DNS 排错帮助	9-10
9.4 夏令时	9-11
9.4.1 夏令时介绍	9-11
9.4.2 夏令时配置任务序列	9-11
9.4.3 夏令时举例	9-12
9.4.4 夏令时排错帮助	9-12
第 10 章 vpn 操作	10-1
10.1 MPLS.....	10-1
10.1.1 MPLS 概述	10-1
10.2 LDP	10-7
10.2.1 LDP 简介	10-7
10.2.2 LDP 配置	10-14
10.2.3 LDP 典型案例	10-19
10.2.4 LDP 排错帮助	10-20
10.3 MPLS VPN	10-21
10.3.1 BGP/MPLS VPN 简介	10-21
10.3.2 BGP MPLS VPN 配置	10-30
10.3.3 BGP MPLS VPN 典型案例	10-36
10.3.4 MPLS BGP VPN 排错帮助	10-49
10.4 MPLS VPN 访问公网	10-50
10.4.1 访问公网简介	10-50
10.4.2 访问公网配置	10-53
10.4.3 访问公网典型案例	10-54
10.4.4 访问公网排错帮助	10-63

10.5 VPLS	10-64
10.5.1 VPLS 简介	10-64
10.5.2 VPLS 配置	10-72
10.5.3 VPLS 典型案例	10-74
10.5.4 VPLS 排错帮助	10-88
 第 11 章 虚拟化操作	 11-1
11.1 VSF	11-1
11.1.1 概述	11-1
11.1.2 VSF 相关配置	11-7
11.1.3 VSF 典型案例	11-11
11.1.4 VSF 排错帮助	11-13
 第 12 章 IPV6 业务操作	 12-1
12.1 DHCPv6	12-1
12.1.1 DHCPv6 简介	12-1
12.1.2 DHCPv6 服务器配置	12-2
12.1.3 DHCPv6 中继代理配置	12-3
12.1.4 DHCPv6 前缀代理服务器端配置	12-4
12.1.5 DHCPv6 客户端配置	12-6
12.1.6 DHCPv6 前缀代理客户端配置	12-6
12.1.7 DHCPv6 配置举例	12-7
12.1.8 DHCPv6 排错帮助	12-10
12.2 DHCPv6 option37, 38	12-11
12.2.1 DHCPv6 option37, 38 介绍	12-11
12.2.2 DHCPv6 option37, 38 配置任务序列	12-12
12.2.3 DHCPv6 option37, 38 举例	12-16
12.2.4 DHCPv6 option37, 38 排错帮助	12-19
12.3 ND 绑定	12-20
12.3.1 概述	12-20
12.3.2 ND 绑定配置	12-20
12.3.3 ND 绑定举例	12-20
12.4 RIPng	12-21
12.4.1 RIPng 介绍	12-21
12.4.2 RIPng 配置	12-22
12.4.3 RIPng 典型案例	12-26
12.4.4 RIPng 排错帮助	12-29
12.5 OSPFv3	12-29

12.5.1 OSPFv3 介绍	12-29
12.5.2 OSPFv3 配置	12-32
12.5.3 OSPFv3 案例	12-36
12.5.4 OSPFv3 排错帮助	12-39
12.6 MBGP4+	12-39
12.6.1 MBGP4+简介	12-39
12.6.2 MBGP4+配置	12-40
12.6.3 MBGP4+案例	12-41
12.6.4 MBGP4+排错帮助	12-43
12.7 IPv6 黑洞路由	12-43
12.7.1 黑洞路由介绍	12-43
12.7.2 IPv6 黑洞路由配置任务	12-43
12.7.3 黑洞路由举例	12-43
12.7.4 黑洞路由排错帮助	12-44
12.8 IPv6 组播协议	12-45
12.8.1 PIM-DM6	12-45
12.8.2 PIM-SM6	12-49
12.8.3 ANYCAST RP v6	12-57
12.8.4 PIM-SSM6	12-61
12.8.5 IPv6 DCSCM	12-64
12.8.6 MLD	12-68
12.8.7 MLD Snooping	12-72
12.9 IPv6 安全 RA	12-77
12.9.1 IPv6 安全 RA 简介	12-77
12.9.2 IPv6 安全 RA 配置任务序列	12-78
12.9.3 IPv6 安全 RA 典型案例	12-78
12.9.4 IPv6 安全 RA 排错帮助	12-79
12.10 SAVI	12-79
12.10.1 SAVI 介绍	12-79
12.10.2 SAVI 配置	12-80
12.10.3 SAVI 典型应用	12-83
12.10.4 SAVI 排错帮助	12-85
12.11 IPv6 VRRPv3	12-85
12.11.1 VRRPv3 简介	12-85
12.11.2 VRRPv3 配置	12-88
12.11.3 VRRPv3 典型案例	12-89
12.11.4 VRRPv3 排错帮助	12-90

第1章 基本管理操作

1.1 交换机管理

1.1.1 管理方式

用户购买到交换机设备后，需要对交换机进行配置，从而实现对网络的管理。交换机为用户提供了两种管理方式：带外管理和带内管理。

1.1.1.1 带外管理

带外管理即通过 Console 进行管理，通常情况下，在首次配置交换机或者无法进行带内管理时，用户会使用带外管理方式。例如：用户希望通过远程 Telnet 来访问交换机时，必须首先通过 Console 给交换机配置一个 IP 地址。

用户用 Console 管理的步骤如下：

第一、搭建环境：

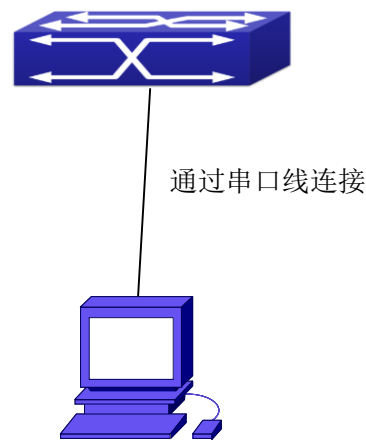


图 1-1 交换机 Console 管理配置环境

按照图 1-1 所示，将 PC 的串口（RS-232 接口）和交换机随机提供的串口线连接，下面是连接中用到的设备说明：

设备名称	说明
PC 机	有完好的键盘和 RS-232 串口，并且安装了终端仿真程序，如 Windows 系统自带超级终端等。
串口线	一端与 PC 机的 RS-232 串口相连；另一端与交换机的

	Console 相连。
交换机	有完好的 Console。

第二、进入超级终端：

连接成功后，打开 Windows 系统自带超级终端。下面是打开 Windows XP 自带超级终端的示例。

1) 点击超级终端：

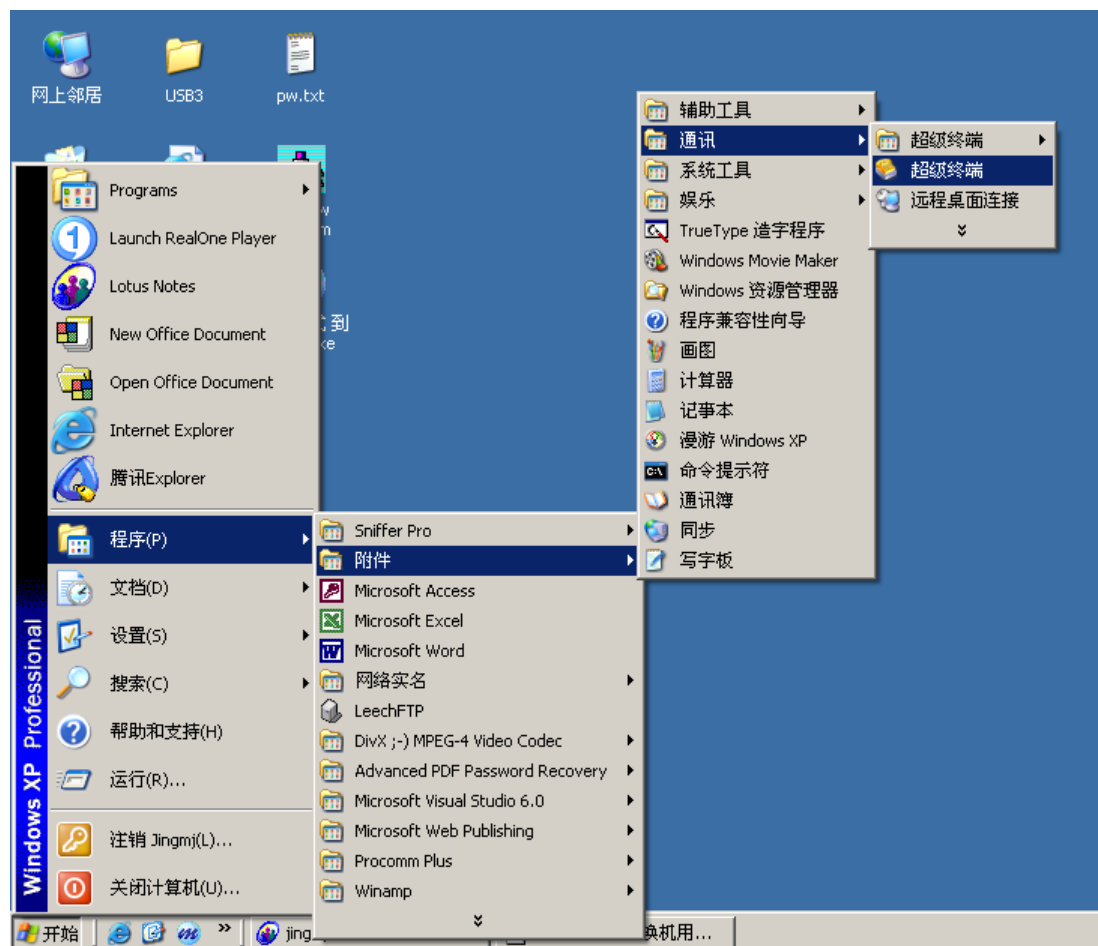


图 1-2 打开超级终端一

2) 在“名称”处填入打开超级终端的名称，例如把它定义为“Switch”：



图 1-3 打开超级终端二

3) 在“连接时使用”处，选择 PC 机使用的 RS-232 串口，如连接的是串口 1，则选择串口 1，点击确定按钮：



图 1-4 打开超级终端三

4) 出现 COM1 属性，波特率选择“9600”，数据位选择“8”，奇偶校验选择“无”，停止位选择“1”，数据流控制选择“无”；或者直接点击“还原默认值”后，点击确定按钮：

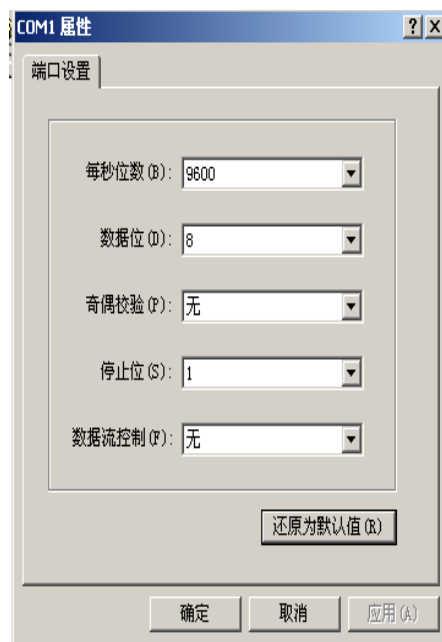


图 1-5 打开超级终端四

5) 出现超级终端的配置界面:

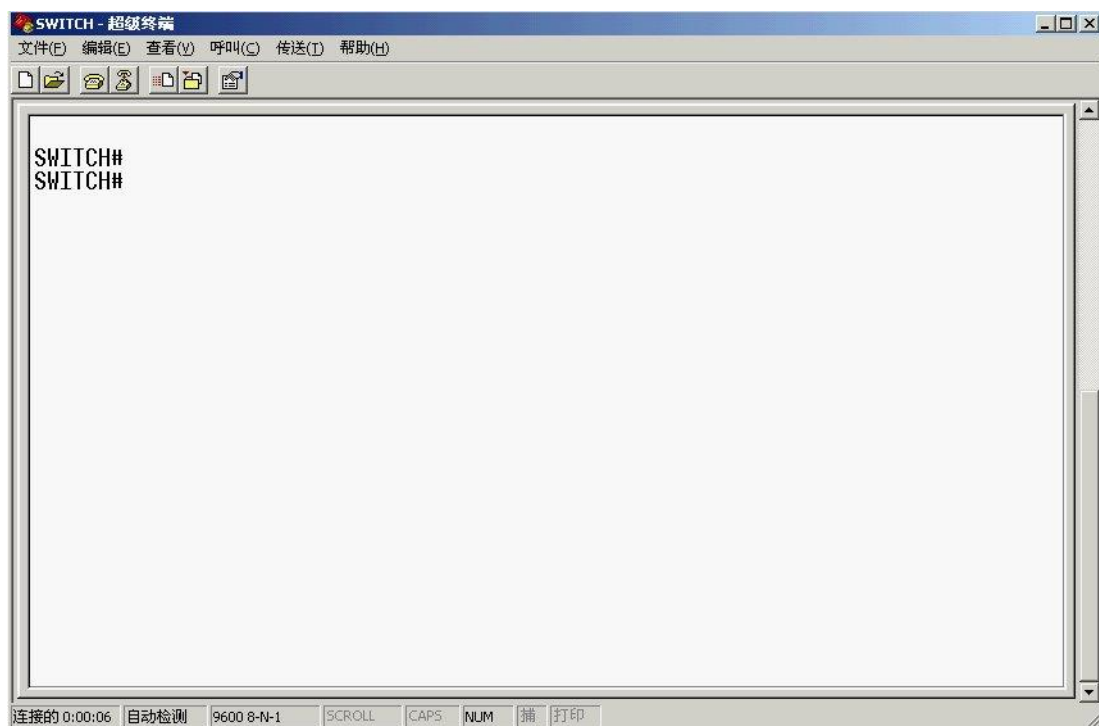


图 1-6 打开超级终端五

第三、进入交换机 CLI 界面:

打开交换机的电源开关。在超级终端的配置界面上出现了如下提示，进入到交换机的 CLI 配置方式:

switch>

接下来用户可以输入相关命令进行管理，命令详见具体章节。另外机箱式核心路由交换机还提供了一个管理网口（或称网管端口），PC 机通过交叉网线与该接口相连也可以进行带外管理，详见端口配置章节。

1.1.1.2 带内管理

所谓带内管理（In-band management），包括通过 Telnet 程序登录到交换机，或者通过 HTTP 协议访问交换机，或者通过神州数码网络（北京）有限公司自主研发的 snmp 网管软件对交换机进行配置管理。交换机提供带内管理方式可以使连接到交换机中的某些设备具备管理交换机的功能。当交换机的配置出现变更，导致带内管理失效时，可以使用带外管理对交换机进行配置管理。

1.1.1.2.1 通过 Telnet 管理交换机

通过 Telnet 管理交换机要具备的条件：

- 1) 交换机配置 IPv4/IPv6 地址；
- 2) 作为 Telnet 客户端的主机 IP 地址与其所属交换机的 VLAN 接口的 IPv4/IPv6 地址在相同网段；
- 3) 若不满足 2)，则 Telnet 客户端可以通过路由器等设备到达交换机某个 IPv4/IPv6 地址。

本交换机是三层交换机，可以配置多个 IPv4/IPv6 地址，配置方法见配置交换机的 IP 地址相关章节。下面以交换机出厂时的情况举例，系统只有 VLAN1 存在。

以下是 Telnet 客户端 Telnet 到交换机的 VLAN1 接口的步骤（以 IPv4 地址为例）：

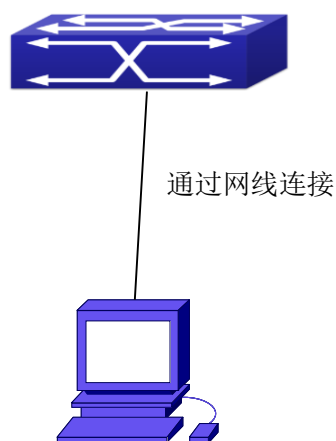


图 1-7 通过 Telnet 管理交换机

第一：配置交换机 IP 地址，并且启动交换机的 Telnet Server 功能。

首先配置主机的 IP 地址，要与交换机的 VLAN1 接口 IP 地址在同一个网段。如交换机的 VLAN1 接口 IP 地址为 10.1.128.251/24，则可以设置主机的 IP 地址为 10.1.128.252/24。

在主机上执行“ping 10.1.128.251”命令，显示是否 ping 通；若 ping 不通，则检查原因。

下面简单介绍交换机 VLAN1 接口的 IP 地址配置命令，在进行带内管理之前，必须通过带外管理即 Console 方式配置交换机的 IP 地址，配置命令如下（以后如不特殊说明，所有的交换机配置时的提示符均采用 Switch）：

```
Switch>
```

```
Switch>enable
```

```
Switch#config
```

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.128.251 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shutdown
```

然后在 Console 的全局配置模式下使用命令 **telnet-server enable** 启动 Telnet Server 功能。配置如下：

```
Switch>enable
```

```
Switch#config
```

```
Switch(config)# telnet-server enable
```

第二：运行 Telnet 客户端程序。

运行 Windows 自带的 Telnet 客户端程序，并且指定 Telnet 的目的地址。

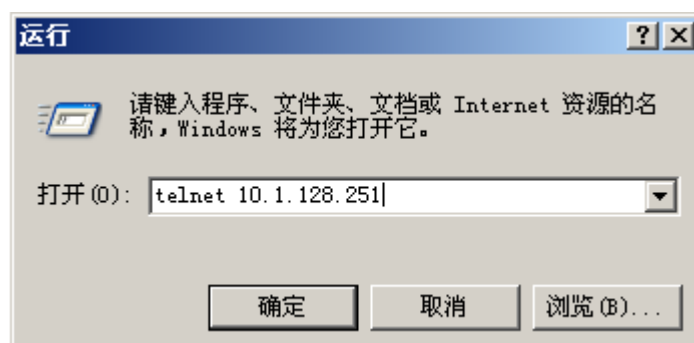


图 1-8 运行 Windows 自带的 telnet 客户端程序

第三：登录交换机。

登录到 Telnet 的配置界面，需要输入正确的用户名和口令，否则交换机将拒绝该 Telnet 用户的访问。该项措施是为了保护交换机免受非授权用户的非法操作。若交换机没有设置授权 Telnet 用户，则任何用户都无法进入交换机的 Telnet 配置界面。因此在允许 Telnet 方式配置管理交换机时，必须在 Console 的全局配置模式下使用命令 **username <username> privilege <privilege> [password (0 | 7) <password>]** 为交换机设置 Telnet 授权用户和口令并使用命令 **authentication line vty login local** 打开本地验证方式，其中 privilege 选项必须存在且为 15。如交换机的授权用户名为 test，口令为明文的 test，设置方式如下：

```
Switch>enable
```

```
Switch#config
```

```
Switch(config)#username test privilege 15 password 0 test
```

```
Switch(config)#authentication line vty login local
```

在 Telnet 配置界面上输入正确的用户名和口令，Telnet 用户就可成功的进入到交换机的 CLI 配置界面。Telnet 登录后与通过 Console 进入后使用的命令完全一致。

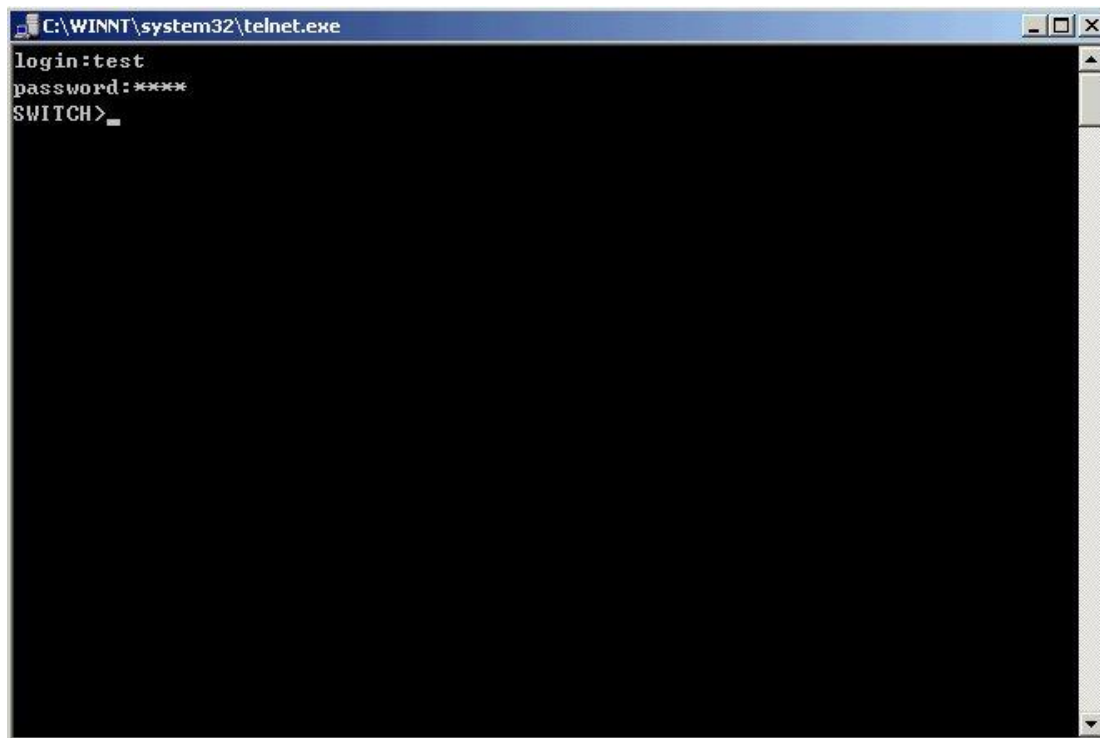


图 1-9 Telnet 配置界面

1.1.1.2.2 通过 HTTP 管理交换机

通过 HTTP 管理交换机要具备的条件：

- 1) 交换机配置 IPv4/IPv6 地址；
- 2) 作为客户端的主机 IPv4/IPv6 地址与其所属交换机的 VLAN 接口的 IPv4/IPv6 地址在相同网段；
- 3) 若不满足 2)，则客户端可以通过路由器等设备到达交换机某个 IPv4/IPv6 地址。

与 Telnet 用户登录交换机类似，只要主机能够 ping/ping6 通交换机的 IPv4/IPv6 地址，并且输入正确的登录口令，该主机就可通过 HTTP 访问交换机。步骤如下：

第一：配置交换机的 IP 地址，并且启动交换机的 HTTP Server 功能。

通过带外管理配置交换机 IP 地址，参见通过 telnet 管理交换机一节。

在 Console 的全局配置模式下使用命令 **ip http server** 启动 HTTP Server 功能，使能 Web 配置。配置如下：

```
Switch>enable
Switch#config
Switch(config)#ip http server
```

第二：在配置主机上运行 HTTP 协议。

在主机上运行 Web 浏览器，在地址栏处输入交换机的 IP 地址，或直接运行 Windows 的 HTTP 协议，比如交换机的 IP 地址为“10.1.128.251”；

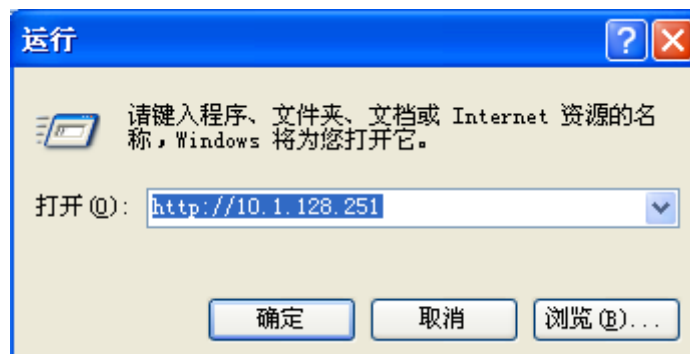


图 1-10 执行 HTTP 协议

在使用 IPv6 地址访问交换机时，推荐使用浏览器 Firefox，版本在 1.5 以上，比如交换机的 IPv6 地址为 3ffe:506:1:2::3，在地址处输入交换机的 IPv6 地址 `http://[3ffe:506:1:2::3]`，地址需要用方括号括起来。

第三：Web 访问交换机。

登录到 Web 的配置界面，需要输入正确的用户名和口令，否则交换机将拒绝该 HTTP 访问。该项措施是为了保护交换机免受非授权用户的非法操作。若交换机没有设置授权 Web 用户，则任何用户都无法进入交换机的 Web 配置界面。因此在允许 Web 方式管理交换机时，必须在 Console 方式下的全局配置模式下使用命令 **username <username> privilege <privilege> [password (0 | 7) <password>]** 为交换机设置 Web 授权用户和口令并使用命令 **authentication line web login local** 打开本地验证方式，其中 privilege 选项必须存在且为 15。如交换机的授权用户名为 admin，口令为明文的 admin，设置方式如下：

```
Switch>enable
```

```
Switch#config
```

```
Switch(config)#username admin privilege 15 password 0 admin
```

```
Switch(config)#authentication line web login local
```

下面是 ES710 的 Web 配置的登录界面：



图 1-11 Web 登录界面

输入正确的用户名和密码，进入交换机的 Web 配置主界面。

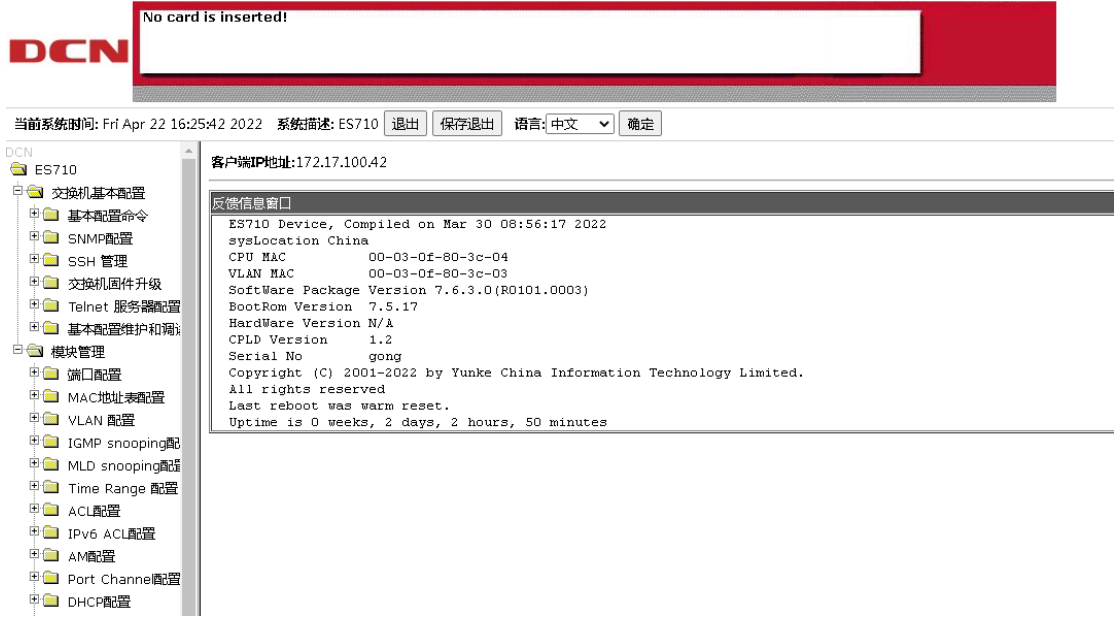


图 1-12 Web 配置界面

注意：通过 Web 配置交换机名称时，交换机名称由英文字母构成。

1.1.1.2.3 通过 snmp 网管软件管理交换机

通过 snmp 网管软件管理交换机要具备的条件：

- 1) 交换机配置 IP 地址;
- 2) 作为客户端的主机 IP 地址与其所属交换机的 VLAN 接口的 IP 地址在相同网段;
- 3) 若不满足 2), 则客户端可以通过路由器等设备到达交换机某个 IP 地址;
- 4) 开启了 snmp 功能。

安装 snmp 网管软件的主机必须能 ping 通交换机的 IP 地址, 这样在运行 Snmp 网管软件时, Snmp 网管软件就能找到交换机设备, 并且对其进行可读写的操作。具体如何通过 Snmp 网管软件管理交换机在本手册中不做介绍, 请参看《Snmp 网管软件用户手册》。

1.1.2 CLI 界面

交换机为用户提供了三种管理界面: CLI (Command Line Interface 命令行) 界面、Web 界面和 Snmp 网管软件。我们将对 CLI 界面做详细的介绍, Web 界面与 CLI 界面功能相似, 就略去不介绍了, Snmp 网管软件请参看《Snmp 网管软件用户手册》。

用户对 CLI 界面并不陌生, 前面我们提到的 Console 管理方式、Telnet 登录到交换机都是通过 CLI 界面对交换机进行配置管理的。

CLI 界面由 Shell 程序提供, 它是由一系列的配置命令组成的, 根据这些命令在配置管理交换机时所起的作用不同, Shell 将这些命令分类, 不同类别的命令对应着不同的配置模式。下面将介绍交换机的 Shell 的特点:

- ☞ 配置模式
- ☞ 配置语法
- ☞ 快捷键
- ☞ 帮助功能
- ☞ 对输入的检查
- ☞ 支持不完全匹配

1.1.2.1 配置模式介绍

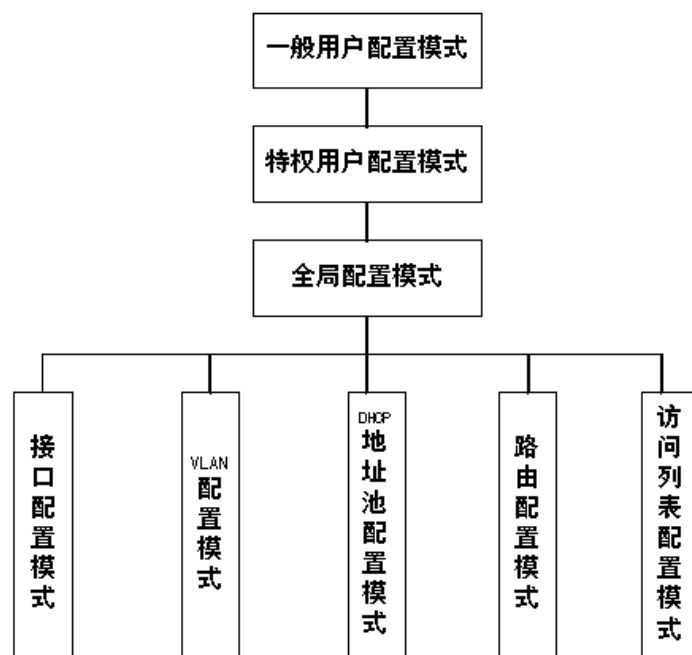


图 1-13 交换机的 Shell 配置模式

1.1.2.1.1 一般用户模式

用户进入 CLI 界面，首先进入的就是一般用户模式，提示符为“Switch>”，符号“>”为一般用户模式的提示符。当用户从特权用户模式使用命令 **exit** 退出时，可以回到一般用户模式。

1.1.2.1.2 特权用户模式

在一般用户模式使用 **enable** 命令，如果已经配置了进入特权用户的口令，则输入相应的特权用户口令，即可进入特权用户模式“Switch#”。当用户从全局配置模式使用 **exit** 退出时，也可以回到特权用户模式。另外交换机提供“Ctrl+z”的快捷键，使得交换机在任何配置模式（一般用户模式除外），都可以退回到特权用户模式。

在特权用户模式下，用户可以查询交换机配置信息、各个端口的连接情况、收发数据统计等。而且进入特权用户模式后，可以进入到全局模式对交换机的各项配置进行修改，因此进入特权用户模式后必须要设置特权用户口令，防止非特权用户的非法使用，对交换机配置进行恶意修改，造成不必要的损失。

1.1.2.1.3 全局配置模式

进入特权用户模式后，只需使用命令 **config**，即可进入全局配置模式“Switch(config)#”。当用户在其他配置模式，如接口配置模式、VLAN 配置模式时，可以使用命令 **exit** 退回到全局配置模式。

在全局配置模式，用户可以对交换机进行全局性的配置，如对 MAC 地址表、端口镜像、创建 VLAN、启动 IGMP Snooping、STP 等。用户在全局配置模式还可通过命令进入到接

口配置模式对各个接口进行配置。

接口配置模式

在全局配置模式，使用命令 **interface** 就可以进入到相应的接口配置模式。交换机操作系统提供了三种端口类型：1.VLAN 接口；2.以太网端口；3.port-channel，因此就有三种接口的配置模式。

接口类型	进入方式	可执行操作	退出方式
VLAN 接口	在全局配置模式下，输入命令 interface vlan <Vlan-id> 。	配置交换机的 IP 等。	使用 exit 命令即可退回全局配置模式。
以太网端口	在全局配置模式下，输入命令 interface ethernet <interface-list> 。	配置交换机提供的以太网接口的双工模式、速率等。	使用 exit 命令即可退回全局配置模式。
port-channel	在全局配置模式下，输入命令 interface port-channel <port-channel-number> 。	配置 port-channel 有关的双工模式、速率等。	使用 exit 命令即可退回全局配置模式。

VLAN 配置模式

在全局配置模式，使用命令 **vlan <vlan-id>** 就可以进入到相应的 VLAN 配置模式。在 VLAN 配置模式，用户可以配置属于本 VLAN 的成员端口。执行 **exit** 命令即可从 VLAN 配置模式退回到全局配置模式。

DHCP 地址池配置模式

在全局配置模式下用 **ip dhcp pool <name>** 命令进入到 DHCP 地址池配置模式 “Switch(Config-<name>-dhcp)#”。在 DHCP 地址池配置模式下可以配置 DHCP 地址池的属性。执行 **exit** 命令即可从 DHCP 地址池配置模式退回到全局配置模式。

路由配置模式

路由协议类型	进入方式	可执行操作	退出方式
RIP 路由协议	在全局配置模式下输入 router rip 命令。	配置 RIP 协议参数。	执行 exit 命令即可退回全局配置模式。
OSPF 路由协议	在全局配置模式下输入 router ospf 命令。	配置 OSPF 协议参数。	执行 exit 命令即可退回全局配置模式。
BGP 路由协议	在全局配置模式下输入 router bgp <AS number> 命令。	配置 BGP 协议参数。	执行 exit 命令即可退回全局配置模式。

访问列表配置模式

访问列表类型	进入方式	可执行操作	退出方式
命名标准 IP 访问列表配置模式	在全局配置模式下输入 ip access-list standard 命令。	配置标准访问列表配置模式。	执行 exit 命令即可退回全局配置模式。

命名扩展 IP 访问列表配置模式	在全局配置模式下输入 ip access-list extended 命令。	配置扩展访问列表的参数。	执行 exit 命令即可退回全局配置模式。

1.1.2.2 配置语法

交换机为用户提供的配置命令形式不完全一样，但它们都遵循交换机配置命令的语法。以下是交换机提供的通用命令格式：

cmdtxt <variable> {enum1 | ... | enumN} [option1 | ... | optionN]

语法说明：黑体字 **cmdtxt** 表示命令关键字；**<variable>** 表示参数为变量；**{enum1 | ... | enumN}** 表示在参数集 **enum1~enumN** 中必须选一个参数；**[option1 | ... | optionN]** 表示该参数可选择一个或者不选。在各种命令中还会出现“<>”，“{ }”，“[]”符号的组合使用，如：**[<variable>]**，**{enum1 <variable> | enum2}**，**[option1 [option2]]** 等等。

下面是几种配置命令语法的具体分析：

- ☞ **show version**，没有任何参数，属于只有关键字没有参数的命令，直接输入命令即可；
- ☞ **vlan <vlan-id>**，输入关键字后，还需要输入相应的参数值；
- ☞ **firewall {enable | disable}**，此类命令用户可以输入 **firewall enable** 或者 **firewall disable**；
- ☞ **snmp-server community {ro | rw} <string>**，出现以下几种输入情况：
snmp-server community ro <string>
snmp-server community rw <string>

1.1.2.3 支持快捷键

交换机为方便用户的配置，特别提供了多个快捷键，如上、下、左、右键及删除键 **BackSpace** 等。如果超级终端不支持上下光标键的识别，可以使用 **ctrl+p** 和 **ctrl+n** 来替代。

按键	功能	
删除键 BackSpace	删除光标所在位置的前一个字符，光标前移	
上光标键“↑”	显示上一个输入命令。最多可显示最近输入的二十个命令	
下光标键“↓”	显示下一个输入命令。当使用上光标键回溯到以前输入的命令时，也可以使用下光标键退回到相对于前一个命令的下一个命令	
左光标键“←”	光标向左移动一个位置	左右键的配合使用， 可对已输入的命令做覆盖修改
右光标键“→”	光标向右移动一个位置	
Ctrl+p	相当于上光标键“↑”的作用	
Ctrl+n	相当于下光标键“↓”的作用	
Ctrl+b	相当于左光标键“←”的作用	
Ctrl+f	相当于右光标键“→”的作用	
Ctrl+z	从其他配置模式（一般用户配置模式除外）直接退回到特权用户模式	
Ctrl+c	打断交换机 ping 或其它正在执行的命令进程	
Tab 键	当输入的字符串可以无冲突的表示命令或关键字时，可以使用 Tab 键将其补充成完整的命令或关键字	

1.1.2.4 帮助功能

交换机为用户提供了两种方式获取帮助信息，其中一种方式为使用“help”命令，另一种为“?”方式。

帮助	使用方法及功能
Help	在任一命令模式下，输入“help”命令均可获取有关帮助系统的简单描述。
“?”	1. 在任一命令模式下，输入“?”获取该命令模式下的所有命令及其简单描述； 2. 在命令的关键字后，输入以空格分隔的“?”，若该位置是参数，会输出该参数类型、范围等描述；若该位置是关键字，则列出关键字的集合及其简单描述；若输出“<cr>”，则此命令已输入完整，在该处键入回车即可； 3. 在字符串后紧接着输入“?”，会列出以该字符串开头的所有命令。

1.1.2.5 对输入的检查

1.1.2.5.1 成功返回信息

通过键盘输入的所有命令都要经过 Shell 的语法检查。当用户正确输入相应模式下的命令后，且命令执行成功，不会显示信息。

1.1.2.5.2 错误返回信息

当用户输入的命令错误，或者参数范围，类型，格式有错误，交换机会提示错误。

1.1.2.6 支持不完全匹配

交换机的 Shell 支持不完全匹配的搜索命令和关键字，当输入无冲突的命令或关键字时，Shell 就会正确解析。

例如：

1. 对特权用户配置命令“show interface ethernet 1/1”，只要输入“sh in e 1/1”即可。
2. 对特权用户配置命令 “show running-config”，如果仅输入“sh r”，系统会报“> Ambiguous command!”，因为 Shell 无法区分“show r”是“show radius”命令还是“show running-config”命令，因此必须输入“sh ru”，Shell 才会正确的解析。

1.2 交换机基本配置

1.2.1 基本配置

交换机的基本配置包括进入和退出特权用户模式的命令、进入和退出接口配置模式的命令、设置及显示交换机的时钟、显示交换机的系统版本信息等。

命令	解释
一般用户配置模式/特权用户配置模式	
enable [<1-15>] disable	用户使用 enable 命令，从一般用户配置模式进入特权用户配置模式或修改当前用户权限级别， disable 为退出特权用户模式命令。
特权用户配置模式	
config [terminal]	从特权用户配置模式进入到全局配置模式。
各种配置模式	
exit	从当前模式退出，进入上一个模式，如在全局配置模式使用本命令退回到特权用户配置模式，在特权用户配置模式使用本命令退回到一般用户配置模式等。
show privilege	显示当前用户权限。
一般用户配置模式和特权用户配置模式之外的其它模式	
end	当前处于非一般用户配置模式或特权用户配置模式时，使用该命令可以直接退回到特权用户配置模式。
特权用户配置模式	
clock set <HH:MM:SS> [YYYY.MM.DD]	设置系统日期和时钟。
show version	显示交换机版本信息。
set default	恢复交换机的出厂设置。
write	将当前运行时配置参数保存到 Flash Memory。
reload	热启动交换机。
show cpu usage	显示 CPU 使用率。
show cpu utilization	显示当前 CPU 的使用率。
show memory usage	显示内存使用率。
全局配置模式	
banner <login motd><LINE> no banner <login motd>	配置使用 telnet 或通过 console 等方式登陆交换机认证成功时显示的欢迎信息。
set boot password	设置下次进入 bootrom 时的密码。 注意：本交换机不支持此命令。

web-auth privilege <1-15> no web-auth privilege	设置 web 登录交换机的级别。

1.2.2 远程管理

1.2.2.1 Telnet

1.2.2.1.1 Telnet 简介

Telnet 远程登录是一个简单的远程终端协议。用户用 Telnet 就可在其所在地通告 TCP 连接注册（即登录）到远程的另一个主机上（使用 IP 地址或主机名）。Telnet 能把用户的击键传到远程的主机，同时也能把远程主机的输出通过 TCP 连接返回到用户屏幕。这种服务是透明的，因为用户的感觉是键盘和显示器是直接连接在远程主机上。

Telnet 使用客户—服务器模式，在本地的系统为 Telnet 客户端，远程主机为 Telnet 服务器。交换机既可以作为 Telnet 服务器也可以作为 Telnet 客户端。

当交换机作为 Telnet 服务器时，用户可以通过 Windows 或其他操作系统自带的 Telnet 客户端软件，Telnet 登录到交换机，就如前面介绍带内管理章节中介绍的一样。交换机作为 Telnet 服务器时最多可以同时与 5 个 Telnet 客户端建立 TCP 连接。

当作为 Telnet 客户端时，在交换机的特权用户配置模式下使用 telnet 命令即可登录到其他远端主机。交换机作为 Telnet 客户端时只能与一个远程主机建立 TCP 连接，如果想与另一个远程主机建立连接，则必须先断开与上一个远程主机的 TCP 连接。

1.2.2.1.2 Telnet 任务序列

- (1) 配置 Telnet 服务器
- (2) 交换机 Telnet 登录到远程主机

1. 配置 Telnet 服务器

命令	解释
全局配置模式	
telnet-server enable no telnet-server enable	打开交换机的 Telnet 服务器功能；本命令的 no 操作为关闭 Telnet 服务器功能。
username <username> [privilege <privilege>] [password [0 7] <password>] no username <username>	配置 Telnet 登录到交换机的本地用户名和口令；本命令的 no 操作为删除本地授权 Telnet 用户。本地 Telnet 用户的执行命令权限必须设为 15。

aaa authorization config-commands no aaa authorization config-commands	对 VTY（即指 Telnet 和 ssh 登录方式）登录用户开启命令授权功能，相应 no 命令关闭命令授权功能。只有开启了此功能并且配置了命令授权方法，执行某个命令时才会要求进行授权。
authentication securityip <ip-addr> no authentication securityip <ip-addr>	配置 Telnet 登录到交换机的安全 IP 地址；本命令的 no 操作为删除授权 Telnet 安全地址。
authentication securityipv6 <ipv6-addr> no authentication securityipv6 <ipv6-addr>	配置 Telnet 登录到交换机的安全 IPv6 地址；本命令的 no 操作为删除授权 Telnet 安全地址。
authentication ip access-class {<num-std> <name>} no authentication ip access-class	为 Telnet/SSH/Web 登录方式绑定标准的 IP ACL 规则；本命令的 no 操作为取消绑定 ACL。
authentication ipv6 access-class {<num-std> <name>} in no authentication ipv6 access-class	为 Telnet/SSH/Web 登录方式绑定标准的 IPv6 ACL 规则；本命令的 no 操作为取消绑定 ACL。
authentication line {console vty web} login method1 [method2 ...] no authentication line {console vty web} login	配置远程登录认证方法列表。
authentication enable method1 [method2 ...] no authentication enable	配置 enable 认证方法列表
authorization line {console vty web} exec method1 [method2 ...] no authorization line {console vty web} exec	配置远程登录授权方法列表。
authorization line vty login <1-15> {local radius tacacs} (none) no authorization line vty login <1-15>	配置 VTY（即指 Telnet 和 ssh 登录方式）登录用户的命令授权方式和授权选择优先级。
accounting line {console vty} command <1-15> {start-stop stop-only none} method1 [method2...] no accounting line {console vty} command <1-15>	配置登录记账方法列表
特权模式	

terminal monitor terminal no monitor	使登录到交换机的 Telnet 客户端显示调试信息；本命令的 no 操作为关闭 Telnet 客户端显示调试信息的功能。
show users	显示通过 telnet 和 ssh 登陆的用户信息，包括线路号，登陆的用户名及用户的 IP。
clear line vty <0-31>	删除指定线路上登陆的用户，强制 telnet 或 ssh 登陆的用户下线。

2. 交换机 Telnet 登录到远程主机

命令	解释
特权模式	
telnet [vrf <vrf-name>] {<ip-addr> <ipv6-addr> host <hostname>} [<port>]	使用交换机提供的 Telnet 客户端登录到远程主机。

1.2.2.2 SSH

1.2.2.2.1 SSH 简介

SSH 是 Secure Shell 安全外壳的简写，它建立在可靠的 TCP/IP 等通信协议之上，通过 SSH 服务器和 SSH 客户端软件之间的密钥交换，身份认证，加密等算法的协商，在它们之间建立一条安全的传输信息的通道，保证双方交互的信息不能被任何第三方截取和破译，从而最大限度的保证了对实现 SSH 服务器功能的交换机的配置和管理。目前交换机上实现的是符合 SSH2.0 协议规定的 SSH 服务器。支持 SSH2.0 标准的终端软件如 SSH Secure Client、putty 等可以利用 SSH2.0 协议，远程登陆交换机上的 SSH 服务器，实现对该交换机的安全的配置和管理。

目前 SSH 服务器支持客户端软件对主机的 RSA 公钥认证，支持协议规定的 3DES 加密算法，以及 SSH 服务器对 SSH 用户的密码认证等。

1.2.2.2.2 SSH 服务器配置任务序列

命令	解释
全局配置模式	
ssh-server enable no ssh-server enable	打开交换机的 SSH 服务器功能；本命令的 no 操作为关闭 SSH 服务器功能。

username <username> [privilege <privilege>] [password [0 7] <password>] no username <username>	配置 SSH 客户端软件登录到交换机的用户名和口令；本命令的 no 操作为删除授权 SSH 用户。
ssh-server timeout <timeout> no ssh-server timeout	设置 SSH 认证超时时间；本命令的 no 操作恢复缺省的 SSH 认证超时时间。
ssh-server authentication-retries <authentication-retries> no ssh-server authentication-retries	设置 SSH 认证重试次数；本命令的 no 操作恢复缺省的 SSH 认证重试次数。
ssh-server host-key create rsa modulus <modulus>	生成新的 SSH 服务器的 RSA 主机密钥对。
特权模式	
terminal monitor terminal no monitor	使登录到交换机的 SSH 客户端显示调试信息；本命令的 no 操作为关闭 SSH 客户端显示调试信息的功能。
show crypto key	显示 ssh 加密密钥。
crypto key clear rsa	清除 ssh 加密密钥。

1.2.2.2.3 SSH 服务器配置举例

案例 1:

组网需求：交换机端运行 SSH 服务器，终端运行支持 SSH2.0 标准的客户端软件如 Secure shell client 或 putty 等。客户端使用用户名和密码方式登录交换机。

交换机首先配置本地的地址，然后增加 SSH 用户，启动 SSH 服务，这样 SSH2.0 客户端软件就可以利用交换机配置的用户名和密码远程登录配置 SSH 服务的交换机了。

```
Switch(config)#ssh-server enable
```

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 100.100.100.200 255.255.255.0
```

```
Switch(Config-if-Vlan1)#exit
```

```
Switch(config)#username test privilege 15 password 0 test
```

如果是 IPv6 网络，终端需要运行支持 IPv6 功能的 SSH 客户端软件，例如 putty6，交换机上除配置本地的地址为 IPv6 地址外，其它配置不变。

1.2.3 配置交换机的 IP 地址

交换机所有以太网接口都默认为二层（DataLink Layer）端口，进行二层转发。VLAN 接口（Interface VLAN）代表了某一个 VLAN 的三层接口功能，可以配置 IP 地址，该 IP 地址也是交换机的 IP 地址。VLAN 接口相关的配置命令都可以在 VLAN 接口模式下配置。交

交换机为用户提供了三种配置 IP 地址的方式：

- ☞ 手工配置方式
- ☞ BootP 方式
- ☞ DHCP 方式

手工配置 IP 地址，即用户为交换机指定一个 IP 地址。

BootP/DHCP 方式是交换机作为 BootP/DHCP Client，向 BootP/DHCP Server 发出 BootPRequest（申请获取地址的请求包）广播包，BootP/DHCP Server 接收到请求后将地址分配给交换机。另外交换机还具有了 DHCP Server 的功能，能够动态的为 DHCP Client 分配 IP 地址、网关地址及 DNS 服务器地址等网络参数。具体 DHCP Server 的配置参见网络协议操作部分。

1.2.3.1 配置交换机的 IP 地址任务序列

1. 进入 VLAN 接口配置模式或网管端口配置模式
2. 手工配置方式
3. BootP 方式
4. DHCP 方式

1.进入 VLAN 接口配置模式或网管端口配置模式

命令	解释
全局配置模式	
interface vlan <vlan-id> no interface vlan <vlan-id>	创建一个 VLAN 接口（VLAN 接口属于三层接口）；本命令的 no 操作为删除交换机创建的 VLAN 接口（三层接口）。
interface ethernet <interface-name>	进入网管端口配置模式。

2.手工配置方式

命令	解释
VLAN 接口配置模式	
ip address <ip_address> <mask> [secondary] no ip address <ip_address> <mask> [secondary]	配置交换机的 VLAN 接口的 IP 地址；本命令的 no 操作为删除交换机的 VLAN 接口的 IP 地址。
ipv6 address <ipv6-address / prefix-length> [eui-64] no ipv6 address <ipv6-address /	配置 IPv6 地址，包括可聚合全球单播地址，本地站点地址，本地链路地址。本命令的 no 操作为删除 IPv6 地址。

prefix-length>	

3.BootP 方式

命令	解释
VLAN 接口配置模式	
ip bootp-client enable no ip bootp-client enable	使能交换机为 BootP Client，通过 BootP 协商方式获取 IP 地址及网关地址；本命令的 no 操作为关闭 BootP Client 功能。

4.DHCP

命令	解释
VLAN 接口配置模式	
ip dhcp-client enable no ip dhcp-client enable	使能交换机为 DHCP Client，通过 DHCP 协商方式获取 IP 地址及网关地址；本命令的 no 操作为关闭 DHCP Client 功能。

1.2.4 SNMP 配置

1.2.4.1 SNMP 介绍

SNMP（Simple Network Management Protocol，简单网络管理协议）是目前使用最广泛的网络管理协议，大量应用于以 TCP/IP 为基础的计算机网络。SNMP 是一个不断发展着的协议：SNMP v1 简单、易实现，被众多厂家支持；SNMPv2c 对 v1 的功能进行了部分修正和丰富，开始支持分层式的网管；SNMPv3 在安全性上做了极大的扩充，添加了 USM（基于用户的安全模型）和 VACM（基于视图的访问控制模型）子系统。

SNMP 协议提供了一个在网络中两点之间交换管理信息的直接的、基本的方法。SNMP 采用轮询的消息查询机制，采用被广泛使用的面向无连接的传输层协议 UDP 来传输消息，因此能够很好的被现有的计算机网络支持。

SNMP 协议采用管理站/代理模式，因此 SNMP 结构包括两个部分：NMS 网络管理站（Network Management Station），是运行支持 SNMP 协议的网管软件客户端程序的工作站，在 SNMP 的网络管理中起核心作用。Agent 代理，是运行在被管理网络设备上的服务器端软件，直接管理被管对象。NMS 利用通信手段，通过 Agent 代理来管理各被管对象。（本款及后续系列交换机都具有 Agent 代理功能）。

SNMP 的 NMS 和 Agent 之间通过标准消息进行通信，采取客户/服务器模式，由 NMS 发出请求，Agent 做出应答。SNMP 共有 7 种消息类型：

- ☞ Get-Request
- ☞ Get-Response

- ☞ Get-Next-Request
- ☞ Get-Bulk-Request
- ☞ Set-Request
- ☞ Trap
- ☞ Inform-Request

NMS 通过 Get-Request、Get-Next-Request, Get-Bulk-Request 和 Set-Request 消息来对 Agent 发出查询和设置管理变量的请求, Agent 在收到请求后, 用 Get-Response 消息来对请求进行回复。在某些特殊情况下, 如网络设备端口 UP/DOWN、网络拓扑变化时候, Agent 会主动的向 NMS 发送 Trap 消息来通知 NMS 发生了异常事件。(当然, NMS 还可以通过设置 RMON 功能, 自行对一些情况设置警报, 当设定的警报事件触发后, Agent 将根据设置发送 Trap 或记录 Log, 关于 RMON, 下面有具体介绍)。Inform-Request 主要用于 NMS 之间进行通信, 一般应用于分层式的网络管理。

USM 提供了完善的加密和鉴别服务, 保证了传输的安全性。根据用户输入的密码, USM 对报文进行加密, 保证报文在传输过程中不被察看, 对报文进行鉴别, 保证报文在传输过程中不被修改。USM 的标准加密算法为 DES-CBC 算法, 鉴别算法为 HMAC-MD5 和 HMAC-SHA 算法。同时 USM 提供了时间窗的检查, 防止网络延迟和重播的干扰。

VACM 提供了优秀的用户鉴权服务, 保证了访问的安全性。VACM 将访问权限相同的用户放入同一组中, 并为组划分了读写和通告操作的安全范围, 有效地防止了用户的越权行为。

1.2.4.2 MIB 介绍

NMS 可以访问的网管信息是经过精确的定义的, 被完备的组织在一个管理信息数据库中, 即 MIB。所谓 **MIB 管理信息库** (Management Information Base), 是对于通过网络管理协议可以访问信息的精确定义。它使用一个层次化、结构化的形式, 定义了可以从被监测网络设备上获得的管理信息。ISO ASN.1 为 MIB 定义了一个树型结构, 每个 MIB 都用这种树型结构来组织所有的可用信息, 树的每个节点都包含一个 OID 对象标识符(Object Identifier)和一个关于该节点的简短文本描述。OID 是有句点隔开的一组整数, 它命名节点并且可以由它指示该节点在 MIB 树型结构中的位置, 如下图所示:

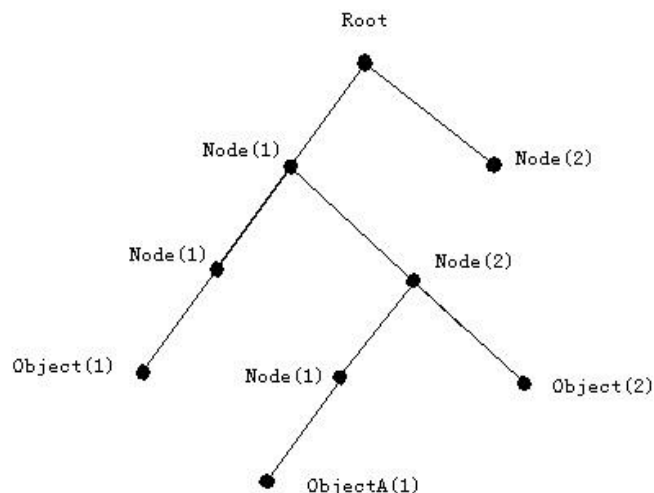


图 1-14 ASN.1 树实例

在该图中，对象 A 的 OID 为 1.2.1.1，NMS 通过这个独一无二的 OID，可以没有歧义的访问到这个对象，从而获取这个对象包含的标准变量。MIB 就按照这个结构定义了一个所监控网络设备的标准变量的集合。

如果您需要浏览 Agent 的 MIB 中的变量信息，需要在 NMS 上运行 MIB 浏览软件。Agent 上的 MIB 一般包括公有 MIB 和私有 MIB 两部分，公有 MIB 内定义的是公开的，所有 NMS 都可以访问的网络管理信息，私有 MIB 内定义的是各设备特有的属性信息，需要设备厂商的支持才可以浏览和控制。

MIB-I [RFC1156] 是 SNMP 公有 MIB 库的第一个版本，后来经过扩充，被 MIB-II [RFC1213] 所取代，MIB-II 保留了 MIB-I 在原来的 MIB 树中的对象标识符。MIB-II 下面包含很多子树，我们将之称为组，这些组里的对象覆盖了网络管理的各个功能域，NMS 通过访问 SNMP Agent 的 MIB 库，就可以得到相应的网络管理信息。

本交换机可以做为 SNMP Agent，支持 SNMPv1/v2c 和 SNMPv3。本交换机支持基本的 MIB-II、RMON 公有 MIB，还支持 BRIDGE MIB 等相关公有 MIB，同时支持自定义的私有 MIB，基本涵盖了本款交换机所涉及的各个相关参数，为网管软件管理提供全面的支持。

1.2.4.3 RMON 介绍

RMON 是对 SNMP 标准基本体系的最重要的扩充。RMON 是一套 MIB 定义，其作用是定义标准的网络监视功能和接口，使基于 SNMP 的管理终端和远程监视器之间能够通信。RMON 提供了一种有效并且高效的方法来监视子网范围内的行为。

RMON 的 MIB 分为 10 组，该交换机支持其中最常用的 1、2、3、9 组，即：

统计组(statistics): 维护代理监视的每一个子网的基本使用和错误统计。

历史组(history): 记录从统计组可得到的信息的周期性统计样本。

警报组(alarm): 允许管理控制台人员为 RMON 代理记录的任何计数或整数设置采样间隔和报警阈值。

事件组(event): 一个关于由 RMON 代理产生的所有事件的表。

其中，警报组需要事件组的实现。统计组和历史组是显示现在或以前的一些子网统计数据。警报组和事件组则提供了一种可以监视网络上任意整数型数据变化的方法，并在数据异常时提供一些警告动作（发送 Trap 或者记录 Log）。

1.2.4.4 SNMP 配置

1.2.4.4.1 SNMP 配置任务序列

1. 打开或关闭 SNMP 代理服务器功能
2. 配置 SNMP 团体字符串及代理设备的属性
3. 配置 SNMP 管理站地址
4. 配置引擎号
5. 配置用户
6. 配置组
7. 配置视图
8. 配置 TRAP
9. 启动/关闭 RMON

1. 打开或关闭 SNMP 代理服务器功能

命令	解释
全局配置模式	
snmp-server enable no snmp-server enable	打开交换机作为 SNMP 代理服务器功能；本命令的 no 操作为关闭 SNMP 代理服务器功能。

2. 配置 SNMP 团体字符串

命令	解释
全局配置模式	
snmp-server community {ro rw} {0 7} <string> [access {<num-std> <name>}] [ipv6-access {<ipv6-num-std> <ipv6-name>}] [read <read-view-name>] [write <write-view-name>] no snmp-server community <string> [access {<num-std> <name>}] [ipv6-access	设置交换机的团体字符串；本命令 no 操作为删除配置的团体字符串。

{<ipv6-num-std> <ipv6-name>}]	

3.配置 SNMP 管理站地址

命令	解释
全局配置模式	
snmp-server securityip {<ipv4-address> <ipv6-address> } no snmp-server securityip {<ipv4-address> <ipv6-address> }	设置允许访问本交换机的NMS管理站的安全IPv4或者IPv6地址；本命令的no操作为删除配置的安全IPv4或者IPv6地址。
snmp-server securityip enable snmp-server securityip disable	打开/关闭 NMS 管理站的安全 IP 地址检查功能。

4.配置引擎号

命令	解释
全局配置模式	
snmp-server engineid <engine-string> no snmp-server engineid	设置交换机的本地引擎号，用于 v3。

5.配置用户

命令	解释
全局配置模式	
snmp-server user <use-string> <group-string> [{authPriv authNoPriv} auth {md5 sha} <word>] [access {<num-std> <name>}] [ipv6-access {<ipv6-num-std> <ipv6-name>}] no snmp-server user <user-string> [access {<num-std> <name>}] [ipv6-access {<ipv6-num-std> <ipv6-name>}]	为一个 SNMP 的组添加一个新用户，完成 USM 配置，用于 v3。

6.配置组

命令	解释
全局配置模式	
snmp-server group <group-string> {noauthnopriv authnopriv authpriv} [[read <read-string>] [write	设置交换机的组信息，完成 VACM 配置，用于 v3。

<pre> <write-string>] [notify <notify-string>]] [access {<num-std> <name>}] [ipv6-access {<ipv6-num-std> <ipv6-name>}] no snmp-server group <group-string> {noauthnopriv authnopriv authpriv} [access {<num-std> <name>}] [ipv6-access {<ipv6-num-std> <ipv6-name>}] </pre>	

7.配置视图

命令	解释
全局配置模式	
<pre> snmp-server view <view-string> <oid-string> {include exclude} no snmp-server view <view-string> [<oid-string>] </pre>	设置交换机的视图信息，用于 v3。

8.配置 TRAP

命令	解释
全局配置模式	
<pre> snmp-server enable traps no snmp-server enable traps </pre>	设置允许设备发送 Trap 消息，用于 v1/v2c/v3。
<pre> snmp-server host {<host-ipv4-address> <host-ipv6-address>} {v1 v2c {v3 {Noauthnopriv Authnopriv Authpriv}}} <user-string> no snmp-server host {<host-ipv4-address> <host-ipv6-address>} {v1 v2c {v3 {Noauthnopriv Authnopriv Authpriv}}} <user-string> </pre>	设置接收 SNMP 的 Trap 消息的网络管理站 IPv4 或者 IPv6 地址，v1/v2c 中的 Trap 团体字符串，v3 中的用户名和安全级别。本命令的 no 操作为删除指定的接收 Trap 消息的网络管理站的 IPv4 或者 IPv6 地址。
<pre> snmp-server trap-source {<ipv4-address> <ipv6-address>} no snmp-server trap-source {<ipv4-address> <ipv6-address>} </pre>	设置本交换机发送 trap 时所使用源 IPv4 或 IPv6 地址；本命令的 no 操作为删除配置的发送 trap 使用的源 IPv4 或 IPv6 地址。
端口配置模式	

[no] switchport updown notification enable	开启或者关闭端口 UP/DOWN 事件发送 trap 信息的功能。

9.启动/关闭 RMON

命令	解释
全局配置模式	
rmon enable no rmon enable	打开/关闭 RMON。

1.2.4.5 SNMP 典型配置举例

管理站（NMS）的 IP 地址是 1.1.1.5；交换机（Agent）的 IP 地址是 1.1.1.9。

案例 1：管理站的网管软件使用 SNMP 协议从交换机获取数据。

配置步骤如下：

```
Switch(config)#snmp-server enable
Switch(config)#snmp-server community rw private
Switch(config)#snmp-server community ro public
Switch(config)#snmp-server securityip 1.1.1.5
```

这样，管理站就可以使用 **private** 作为团体字符串对交换机进行可读写的访问，也可以使用 **public** 作为团体字符串对交换机进行只读的访问。

案例 2：管理站要接收来自交换机的 v1 Trap 消息（注：管理站可能设置了对 Trap 的团体字符串的验证，那么此例假设管理站的 Trap 验证团体字符串为 **usertrap**）。

配置步骤如下：

```
Switch(config)#snmp-server enable
Switch(config)#snmp-server host 1.1.1.5 v1 usertrap
Switch(config)#snmp-server enable traps
```

案例 3：管理站的网管软件使用 SNMP v3 协议从交换机获取数据。

配置步骤如下：

```
Switch(config)#snmp-server enable
Switch(config)#snmp-server user tester UserGroup authPriv auth md5 hellotst
Switch(config)#snmp-server group UserGroup AuthPriv read max write max notify max
Switch(config)#snmp-server view max 1 include
```

案例 4：管理站要接收来自交换机的 v3Trap 消息。

配置步骤如下：


```
Switch(config)#snmp-server enable
Switch(config)#snmp-server host 10.1.1.2 v3 authpriv tester
Switch(config)#snmp-server enable traps
```

案例 5： 管理站（NMS）的 IPv6 地址是 2004:1:2:3::2, 交换机（Agent）的 IPv6 地址是 2004:1:2:3::1。管理站的网管软件使用 SNMP 协议从交换机获取数据。

交换机端的配置如下：

```
Switch(config)#snmp-server enable
Switch(config)#snmp-server community rw private
Switch(config)#snmp-server community ro public
Switch(config)#snmp-server securityip 2004:1:2:3::2
```

这样，管理站就可以使用 **private** 作为团体字符串对交换机进行可读写的访问，也可以使用 **public** 作为团体字符串对交换机进行只读的访问。

案例 6： 管理站要接收来自交换机的 Trap 消息（注：管理站可能设置了对 Trap 的团体字符串的验证，那么此例假设管理站的 Trap 验证团体字符串为 **usertrap**）。

交换机端的配置如下：

```
Switch(config)#snmp-server host 2004:1:2:3::2 v1 usertrap
Switch(config)#snmp-server enable traps
```

1.2.4.6 SNMP 排错帮助

在配置、使用 SNMP 时，可能会由于物理连接、配置错误等原因导致 SNMP 未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 **show interface** 命令），保证交换机和主机能互相 ping 通（使用 **ping** 命令）；
- ☞ 然后，务必确认交换机打开了 SNMP Agent 服务器功能（使用 **snmp-server enable** 命令）；
- ☞ 接着，保证为 NMS 配置安全 IP（使用 **snmp-server securityip** 命令）和团体字符串（使用 **snmp-server community** 命令）正确，因为只要有一个错误 SNMP 将不能正确和 NMS 通信；
- ☞ 如果需要使用 Trap 功能，务必打开 Trap 功能（使用 **snmp-server enable traps** 命令），为了保证 Trap 能发送到指定的主机，请记住正确配置 Trap 的目标主机的 IP 和团体字符串（使用 **snmp-server host** 命令）；
- ☞ 如果需要使用 RMON 功能，必须首先打开 RMON（使用 **rmon enable** 命令）；
- ☞ 在 SNMP 运行过程中，如果用户还有疑惑，可以使用 **show snmp** 命令查看 SNMP

收发包的统计信息；可以使用 `show snmp status` 命令查看 SNMP 的配置信息；可以使用 `debug snmp packet` 命令打开 SNMP 的调试开关，查看调试信息。如果上述查看方式的尝试仍无法解决 SNMP 的问题，那么请与技术服务中心联系。

1.2.5 交换机升级

1.2.5.1 交换机系统文件

系统文件包括系统映像文件和引导文件。交换机的升级就是对这两个文件更新，方法就是用新的文件覆盖旧的文件。

系统映像文件是指交换机硬件驱动和软件支持程序等的压缩文件，即我们通常说的 IMG 升级文件。交换机系统映像文件只允许保存在 FLASH 中，文件名固定为 `nos.img`。

引导文件是指引导交换机初始化等的文件，即我们通常说的 ROM 升级文件（如果文件较大，可以压缩为 IMG 文件）。引导文件只允许保存在 ROM 中。交换机规定引导文件的文件名固定为 `boot.rom`。

系统映像文件和引导文件的升级方法是一样的。交换机为用户提供了两种模式下的交换机升级：一、BootROM 模式；二、Shell 模式下的 TFTP 升级、FTP 升级。下面两节详细介绍这两种模式的升级方法。

1.2.5.2 BootROM 模式升级

在 BootROM 模式下升级交换机只有一种升级方式：TFTP，可通过 BootROM 下的命令设置。

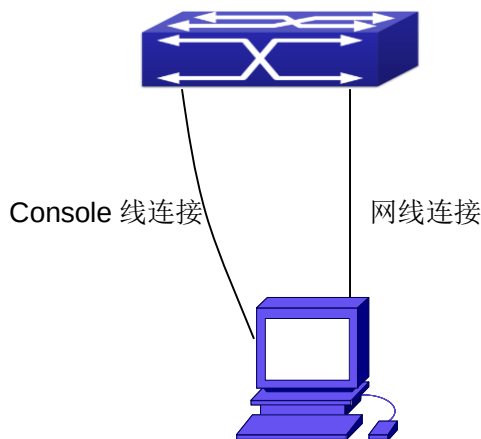


图 1-15 在 BootROM 模式下交换机升级的典型拓扑

升级步骤如下：

第一步：

如图将一台 PC 作为交换机的控制台，并且该控制台的以太网口与交换机的网管端口相连，在 PC 上安装有 TFTP 的服务器软件，及需要升级的 `img` 文件。

第二步:

在交换机启动的过程中, 按住“ctrl+b”键, 直到交换机进入 BootROM 监控模式。显示如下:

[Boot]:

第三步:

在 BootROM 模式下, 执行命令 setconfig, 设置本机在 BootROM 模式下的 IP 地址、服务器的 IP 地址。如设置本机地址为 192.168.1.2, PC 地址为 192.168.1.66, 配置如下:

[Boot]: setconfig

Host IP Address: [10.1.1.1] 192.168.1.2

Server IP Address: [10.1.1.2] 192.168.1.66

[Boot]:

第四步:

打开 PC 中 TFTP 服务器, 运行 TFTP Server 程序。在往交换机下传升级版本时, 请先检查服务器与被升级交换机之间的连接状态, 在交换机端使用 ping 命令, ping 通后, 在交换机的 BootROM 模式下执行 load 命令; 若 ping 不通, 则检查原因。

以下是更新引导文件 boot.rom。

[Boot]: load boot.rom

TFTP from server 192.168.1.66; our IP address is 192.168.1.2

Filename 'boot.rom'.

Load address: 0x300000

Loading:

#####

#####

done

Bytes transferred = 496240 (79270 hex)

[Boot]:

第五步:

在 BootROM 模式下, 执行命令 write boot.rom。以下是对更新引导文件的保存:

[Boot]: write boot.rom

File exists, overwrite? (Y/N)[N] y

Writing flash:/boot.rom.....

Write flash:/boot.rom OK.

[Boot]:

第六步:

以下是更新系统映像文件的配置:

[Boot]: load nos.img

TFTP from server 192.168.1.66; our IP address is 192.168.1.2

Filename 'nos.img'.

Load address: 0x300000

Loading: #####

done

Bytes transferred = 51635 (c9b3 hex)

[Boot]:

第七步:

在 BootROM 模式下, 执行命令 write nos.img。以下是对更新系统映像文件的保存:

[Boot]: write nos.img

File exists, overwrite? (Y/N)[N] y

Writing

flash:/nos.img.....

Write flash:/nos.img OK.

[Boot]

第八步:

升级交换机成功, 在 BootROM 模式下, 执行命令 run 或 reboot, 回到 CLI 配置界面。

[Boot]:run (或者 reboot)

BootROM 模式下的其他命令

1. DIR 命令

用于显示 FLASH 中现存的文件。

[Boot]: dir

256 boot.conf

2270 startup.cfg

2270 1.cfg

255 bootip.conf

```
11838890    a.img
11838890    nos.img
```

7 file(s), 0 dir(s)

Total size:27838464 bytes , used size:23810048 bytes, free size:4028416 bytes

[Boot]:

1.2.5.3 FTP/TFTP 升级

1.2.5.3.1 FTP/TFTP 简介

FTP（File Transfer Protocol）/TFTP（Trivial File Transfer Protocol）都是文件传输协议，在TCP/IP协议族中处于第四层，即属于应用层协议，主要用于主机之间、主机与交换机之间传输文件。它们都采用客户机—服务器模式进行文件传输。下面是它们的不同之处。

FTP承载于TCP之上，提供可靠的面向连接数据流的传输服务，但它不提供文件存取授权，以及简单的认证机制（通过明文传输用户名和密码来实现认证）。FTP在进行文件传输时，客户机和服务器之间要建立两个连接：控制连接和数据连接。首先由FTP客户机发出传送请求，与服务器的21号端口建立控制连接，通过控制连接来协商数据连接。

数据连接有两种方式：

一种为主动连接。客户端是主动告诉服务器自己用于数据传输的地址和端口号，控制连接将一直保留到数据传输完成。接着服务器在20号端口没有被使用的条件下采用20号端口与客户机提供的地址和端口号建立数据连接，并传输数据；如果20号端口正在被使用，通过设置20号端口可以重用，服务器通过系统自动生成另外的端口号建立数据连接。

另外一种方式是被动连接。客户端通过控制连接告诉服务器端建立被动连接，服务器端就建立自己的数据监听端口，将这个端口通过控制连接告诉客户端，由客户端主动与指定地址的端口建立数据连接。

由于数据连接是通过指定的地址和端口号，还存在通过第三方来提供数据连接服务。

TFTP 承载在 UDP 之上，提供不可靠的数据流传输服务，同时也不提供用户认证机制以及根据用户权限提供对文件操作授权；它是通过发送报文，应答方式，加上超时重传方式来保证数据的正确传输。TFTP 相对于 FTP 的优点是提供简单的、开销不大的文件传输服务。

交换机实现 FTP/TFTP 客户机和服务器的功能。当交换机作为 FTP/TFTP 客户机时，在不影响交换机正常工作的情况下，能从远端 FTP/TFTP 服务器（可以是主机和其它交换机）下载配置文件或系统文件，（FTP 客户端可以查看服务器上的文件列表），也可以将交换机当前配置文件或系统文件上载到远端 FTP/TFTP 服务器上（可以是主机和其它交换机）的功能；当交换机作为 FTP/TFTP 服务器时，同样它能为它授权的 FTP/TFTP 客户机提供上传和下载文件的服务，（FTP 服务器还提供传输服务器上文件列表功能）。

下面介绍 FTP/TFTP 中经常用到的术语：

ROM：EPROM 的简称，可擦写只读寄存器。交换机采用 FLASH 闪存代替 EPROM。

SDRAM: 交换机内存, 用于系统软件的运行和配置序列的存放。

FLASH: 闪存, 用于保存系统文件和配置文件。

系统文件: 包括系统映像文件和引导文件。

系统映像文件: 是指交换机硬件驱动和软件支持程序等的压缩文件, 即我们通常说的 **IMG** 升级文件。交换机系统映像文件只允许保存在 **FLASH** 中。交换机规定, 在全局配置模式下通过 **FTP** 上载系统映像文件的文件名固定为 **nos.img**, 拒绝其它系统 **IMG** 文件的上载。

引导文件: 是指引导交换机初始化等的文件, 即我们通常说的 **ROM** 升级文件 (如果文件较大, 可以压缩为 **IMG** 文件)。引导文件只允许保存在 **ROM** 中。交换机规定引导文件的文件名固定为 **boot.rom**。

配置文件: 包括启动配置文件和运行配置文件。区分启动配置文件和运行配置文件便于实现配置的备份和更新。

启动配置文件: 是指交换机启动时采用的配置序列。启动配置文件存放在非易失存储介质中, 即对应着所谓的配置保留。对于不支持 **CF** 卡的设备, 交换机启动配置文件只存放在 **FLASH** 中。对于支持 **CF** 卡的设备, 启动配置文件可以存放在 **FLASH** 中或者 **CF** 卡中。对于支持配置多启动配置文件的设备, 启动配置文件名为扩展名为 **.cfg** 的文件, 默认情况下为 **startup.cfg**。对于不支持多配置文件的设备, 启动配置文件名固定为 **startup-config**。

运行配置文件: 是指交换机当前运行的配置序列。交换机运行配置文件存放在内存中。目前, 使用命令 **write** 或命令 **copy running-config startup-config**, 可将运行配置序列 **running-config** 从内存中保存到 **FLASH** 中, 即实现运行配置序列到启动配置文件的转变, 形成配置保留。为了禁止非法文件的上载和方便配置, 交换机规定, 运行配置文件名固定为 **running-config**。

出厂配置文件: 即 **factory-config** 文件, 是交换机出厂时的配置文件, 使用命令 **set default** 和命令 **write**, 重启后可将配置文件恢复成出厂配置文件。

1.2.5.3.2 FTP/TFTP 配置

交换机作为 **FTP** 或者 **TFTP** 客户机时, 在配置上极其相似, 因此本手册把 **FTP** 和 **TFTP** 作为客户机时的配置结合在一起说明。

1.2.5.3.2.1 FTP/TFTP 配置任务序列

1. FTP/TFTP 客户机配置

- (1) 上载/下载配置文件或系统文件
- (2) **FTP** 客户机查看服务器上文件列表

2. FTP 服务器配置

- (1) 启动 **FTP Server**
- (2) 配置 **FTP** 登录用户名和口令
- (3) 修改 **FTP Server** 连接空闲时限
- (4) 关闭 **FTP Server**

3. TFTP 服务器配置

- (1) 启动 TFTP Server
- (2) 配置 TFTP Server 连接空闲时限
- (3) 配置超时范围内没有收到应答报文的重传次数
- (4) 关闭 TFTP Server

1. FTP/TFTP 客户机配置

(1) FTP/TFTP 客户机端上下载文件

命令	解释
特权用户配置模式	
copy <source-url> <destination-url> [ascii binary]	FTP/TFTP 客户机上下载文件。

(2) FTP 客户机查看服务器上文件列表

特权用户配置模式	
ftp-dir <ftpServerUrl>	FTP 客户机查看服务器上文件列表 FtpServerUrl 为 ftp://user:password@IPv4 IPv6 Address 格式。

2. FTP 服务器配置

(1) 启动 FTP 服务器

命令	解释
全局配置模式	
ftp-server enable no ftp-server enable	启动 FTP Server; 本命令的 no 操作为关闭 FTP Server 服务, 禁止 FTP 用户登录。

(2) 配置 FTP 登录用户名和口令

命令	解释
全局配置模式	
ip ftp username <username> password [0 7] <password> no ip ftp username <username>	配置 FTP 登录用户名和登录口令; 本命令的 no 操作为删除配置的用户名, 同时也删除密码。

(3) 修改 FTP Server 连接空闲时限

命令	解释
全局配置模式	
ftp-server timeout <seconds>	设置连接空闲时限。

3. TFTP 服务器配置

(1) 启动 TFTP 服务器

命令	解释

全局配置模式	
tftp-server enable no tftp-server enable	启动 TFTP Server；本命令的 no 操作为关闭 TFTP Server 服务，禁止 TFTP 用户登录。

(2) 修改 TFTP Server 连接空闲时限

命令	解释
全局配置模式	
tftp-server transmission-timeout <seconds>	设置超时的时间间隔。

(3) 修改 TFTP Server 连接重传次数

命令	解释
全局配置模式	
tftp-server retransmission-number <number>	设置超时时间内的最大重传次数。

1.2.5.3.3 FTP/TFTP 配置举例

无论地址为 IPv4 还是 IPv6，命令设置方式相同，下面的举例中仅以 IPv4 地址为例。

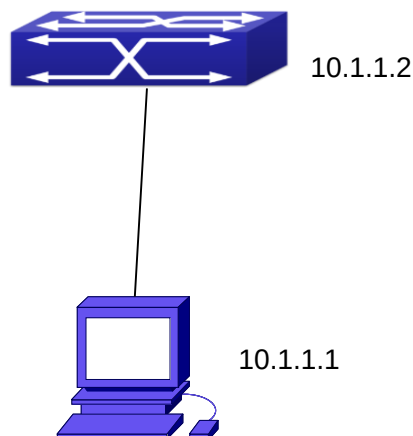


图 1-16 作为 FTP/TFTP client 下载 nos.img 文件

实例 1：交换机作为 FTP/TFTP 客户端来使用。交换机通过某个端口与计算机相连接，计算机为 FTP/TFTP 服务器，IP 地址为 10.1.1.1，交换机作为 FTP/TFTP 客户端，交换机管理 VLAN 的 IP 地址为 10.1.1.2。从计算机上下载交换机的“nos.img”文件。

☞ FTP 配置:

计算机端配置:

在计算机上启动 FTP Server 软件，并且设置用户名为“Switch”，密码为“superuser”。并将“12_30_nos.img”文件放到计算机上对应的 FTP Server 目录下。

交换机的配置步骤如下:

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch#copy ftp://Switch:superuser@10.1.1.1/12_30_nos.img nos.img
```

执行完上述命令，交换机就可将计算机上的“nos.img”文件下载到 FLASH 中了。

☞ TFTP 配置:

计算机端配置:

在计算机机上启动 TFTP Server 软件，并将“12_30_nos.img”文件放到计算机上对应的 TFTP Server 目录下。

交换机的配置步骤如下:

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch#copy tftp://10.1.1.1/12_30_nos.img nos.img
```

实例 2: 交换机作为 FTP 服务器来使用。交换机通过某个端口与计算机相连接，交换机作为 FTP 服务器，计算机作为 FTP 客户端，将交换机上的“nos.img”文件传送到计算机上，保存为 12_25_nos.img。

交换机的配置步骤如下:

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch(config)#ip ftp username Switch password 0 superuser
```

计算机端配置:

通过 FTP 客户端软件，登录到交换机，用户名为“Switch”，密码为“superuser”，通过“get

nos.img 12_25_nos.img”命令将交换机上的“nos.img”文件下载到计算机上。

实例 3: 交换机作为 TFTP 服务器来使用。交换机通过某个端口与计算机相连接，交换机作为 TFTP 服务器，计算机作为 TFTP 客户端，将交换机上的“nos.img”文件传送到计算机上。

交换机的配置步骤如下：

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

```
Switch(config)#tftp-server enable
```

计算机端配置：

通过 TFTP 客户端软件，登录到交换机，通过“tftp”命令将交换机上的“nos.img”文件下载到计算机上。

实例 4: 机架式交换机通过 FTP/TFTP 方式升级指定板卡上的 boot.rom 文件。

交换机作为 FTP/TFTP 客户端来使用。交换机通过某个端口与计算机相连接，计算机为 FTP/TFTP 服务器，IP 地址为 10.1.1.1，交换机作为 FTP/TFTP 客户端，交换机管理 VLAN 的 IP 地址为 10.1.1.2。从计算机上下载交换机上槽号为 4 的线卡的 boot.rom 文件。

☞ FTP 配置：

计算机端配置：

在计算机上启动 FTP Server 软件，并且设置用户名为 Switch，密码为 superuser。并将 boot.rom 文件放到计算机上对应的 FTP Server 目录下。

交换机的配置步骤如下：

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch#copy ftp://Switch:superuser@10.1.1.1/boot.rom slot 4 boot.rom
```

执行完上述命令，就可将计算机上的 boot.rom 文件下载到交换机上槽号为 4 的线卡的 FLASH 中了。

☞ TFTP 配置：

计算机端配置：

在计算机机上启动 TFTP Server 软件，并将 boot.rom 文件放到计算机上对应的 TFTP Server 目录下。

交换机的配置步骤如下：

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch#copy tftp://10.1.1.1/boot.rom slot 4 boot.rom
```

执行完上述命令，就可将计算机上的 boot.rom 文件下载到交换机上槽号为 4 的线卡的 FLASH 中了。

实例 5： 交换机作为 FTP 客户机查看 FTP 服务器上文件列表。同步情况：交换机通过以太口和计算机相连，计算机为 FTP 服务器，IP 地址为 10.1.1.1，交换机作为 FTP 客户端，交换机 VLAN1 接口 IP 地址为 10.1.1.2。

FTP 配置：

计算机端：

在计算机上启动 FTP Server 软件，并且设置用户 Switch，密码为 superuser。

交换机：

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
```

```
Switch(Config-if-Vlan1)#no shut
```

```
Switch(Config-if-Vlan1)#exit
```

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

```
Switch#ftp-dir ftp://Switch:superuser@10.1.1.1
```

```
220 Serv-U FTP-Server v2.5 build 6 for WinSock ready...
```

```
331 User name okay, need password.
```

```
230 User logged in, proceed.
```

```
200 PORT Command successful.
```

```
150 Opening ASCII mode data connection for /bin/ls.
```

```
recv total = 480
```

```
nos.img
```

```
nos.rom
```

```
parsecommandline.cpp
```

```
position.doc
```

```
qmdict.zip
```

```
...(省略部分显示)
```

```
show.txt
```

```
snmp.TXT
```

```
226 Transfer complete.
```

1.2.5.3.4 FTP/TFTP 排错帮助

1.2.5.3.4.1 FTP 排错帮助

在使用 FTP 协议上下载系统文件时，务必保证链路的可达，即在使用 FTP 之前，使用 Ping 命令检验 FTP 客户机和服务器的是否可达。如 Ping 不通，则需查看相应排错信息，恢复链路的可达性。

- ✎ 下面是发送文件时的正确信息，如果不正确，请检查链路的可达性并重新执行 copy 命令。

```
220 Serv-U FTP-Server v2.5 build 6 for WinSock ready...
331 User name okay, need password.
230 User logged in, proceed.
200 PORT Command successful.
nos.img file length = 1526021
read file ok
send file
150 Opening ASCII mode data connection for nos.img.
226 Transfer complete.
close ftp client.
```

- ✎ 下面是接收文件时的正确信息，如果不正确，请检查链路的可达性并重新执行 copy 命令。

```
220 Serv-U FTP-Server v2.5 build 6 for WinSock ready...
331 User name okay, need password.
230 User logged in, proceed.
200 PORT Command successful.
recv total = 1526037
*****
write ok
150 Opening ASCII mode data connection for nos.img (1526037 bytes).
226 Transfer complete.
```

- ✎ 如果交换机通过 FTP 升级系统文件或系统启动文件时，必须等待系统提示出现表示升级成功的“close ftp client.”或“226 Transfer complete.”的字样，此时交换机可以进行重启的操作，否则将会导致交换机无法启动。若出现通过 FTP 升级系统文件及系统启动文件失败时，请重新升级或进入到 BootROM 模式升级。

1.2.5.3.4.2 TFTP 排错帮助

在使用 TFTP 协议上下载系统文件时，务必保证链路的可达，即在使用 TFTP 之前，使

用 Ping 命令检验 TFTP 客户机和服务器的是否可达。如 Ping 不通,则需查看相应排错信息,恢复链路的可达性。

☞ 下面是发送文件时的正确信息,如果不正确,请检查链路的可达性并重新执行 copy 命令。

```
nos.img file length = 1526021
read file ok
begin to send file,wait...
file transfers complete.
close tftp client.
```

☞ 下面是接收文件时的正确信息,如果不正确,请检查链路的可达性并重新执行 copy 命令。

```
begin to receive file,wait...
recv 1526037
*****

write ok
transfer complete
close tftp client.
```

如果交换机通过 TFTP 升级系统文件或系统启动文件时,必须等待系统提示出现表示升级成功的“close tftp client.”的字样,此时交换机可以进行重启的操作,否则将会导致交换机无法启动。若出现通过 TFTP 升级系统文件及系统启动文件失败时,请重新升级或进入到 BootROM 模式升级。

1.2.5.4 分布式加载

1.2.5.4.1 分布式加载简介

机架式交换机分为主控和线卡两部分,以往版本的机架式交换机升级,主控和线卡均要升级 bootrom 和 img 文件。分布式加载允许线卡内没有 img 文件,启动时线卡从主控获取 img 文件进行加载启动,免除为线卡升级的步骤。

1.2.5.4.2 分布式加载过程

线卡若本身存在实体 img,则启动时先加载自身 img,启动完成之后,线卡来判断版本是否一致,如果一致则继续运行,如果不一致,由线卡向活动主控发消息,活动主控向线卡下发空文件,使线卡用这个文件覆盖掉自己原来的 img,从而在下次启动时重新从主控获取自己的 img。

☞ 目前只有线卡支持启动之后 flash 无 img 时从主控获取 img,备份主控不支持。

1.2.5.4.3 分布式加载配置举例

1.2.5.4.3.1 活动主控升级

将活动主控升级成 7.0 的 bootrom 和 img 有两种方法：

- 1) BootROM 模式下的 TFTP 升级。
- 2) Shell 模式下的 TFTP 升级、FTP 升级。

1.2.5.4.3.2 线卡升级

☞ 确保活动主控已经升级成新的 7.0 版本的 bootrom 和 img 文件。

线卡升级方法：

- 1) 用 BootROM 模式下的 TFTP 或者 Shell 模式下的 TFTP 升级、FTP 升级将线卡升级成 7.0 版本的 bootrom，
 - 2) 将线卡热插拔一下或者在主控上执行重新启动线卡的命令（**reset slot <slot-id>**）。
- ☞ 若想保留线卡之前的 img 文件，只将本地文件改名保存，线卡只会启动以 nos.img 命名的 img 文件。
- ☞ 可用命令 copy 命令将 img 强制写入线卡内（只有 flash 空间为 16M 的线卡才能写进实体 img），这样每次线卡启动的时候不用从主控获取 img 加载，缩短整机启动时间。

命令	解释
特权配置模式	
copy <source-url> <destination-url> [ascii binary]	将主控 flash 里的版本写入线卡中，不支持将主控的 bootrom 拷贝到线卡上； 或者以 ftp/tftp 方式升级线卡的 bootrom 和 img 文件。

1.2.5.4.3.3 备份主控升级

活动主控和备份主控默认在 7.0 版本内不同步，为了保证交换机的正常工作，请保持活动主控和备份主控的版本一致，避免运行中活动主控故障导致业务中断。

备份主控升级方法如下：

- ☞ 确保将活动主控升级成新的 7.0 版本的 bootrom 和 img 文件
- 1) 用 BootROM 模式下的 TFTP 或者 Shell 模式下的 TFTP 升级、FTP 升级将备份主控升级至 7.0 版本的 bootrom。
 - 2) 用 BootROM 模式下的 TFTP 或者 Shell 模式下的 TFTP 升级、FTP 升级将备份主控升级至 7.0 版本的 img 文件。
 - 3) 或者在活动主控上配置强制同步命令，然后重启或者热插拔备份主控。

命令	解释
全局配置模式	
force sync software-version {disable enable}	配置活动主控与线卡之间强制版本同步，适用独立和堆叠模式。

force intra sync software-version {disable enable}	堆叠模式下，配置机架内活动主控与线卡之间强制版本同步。
force sm sync software-version {disable enable}	配置活动主控与备份主控之间强制版本同步，适用独立模式和堆叠模式。
force intra sm sync software-version {disable enable}	堆叠模式下，配置机架内活动主控与备份主控之间强制版本同步。

1.3 文件系统

1.3.1 文件存储设备介绍

交换机的文件存储设备主要包括 FLASH 和 CF 卡。FLASH 是交换机上最常用的存储设备，可以用来存放系统映像文件（IMG 文件）、系统引导文件（ROM 文件）和系统配置文件（CFG 文件）。

CF 卡是交换机上新增的一种存储设备，可以看做是 FLASH 设备的一种扩展存储介质。在 CF 卡上可以存储 IMG 文件、ROM 文件、CFG 文件以及上电控制器文件（MCU 文件）。当 CF 卡插到交换机上的 Compact Flash 插槽后，可以在 Shell 模式下或 Bootrom 模式下进行文件的复制、删除、重命名以及目录的创建、删除等操作，还可以利用 CF 卡进行 IMG 文件、ROM 文件以及 MCU 文件的升级操作。

1.3.2 文件系统操作任务配置序列

1. CF 卡的挂载与卸载操作
2. 存储设备格式化操作
3. 子目录的创建操作
4. 子目录的删除操作
5. 修改当前存储设备的工作路径
6. 显示当前工作路径
7. 显示存储设备中的指定文件或目录信息
8. 删除文件系统上的指定文件
9. 文件重命名操作
10. 文件复制操作

1. CF 卡的挂载与卸载操作

命令	解释
特权用户配置模式	
mount <device>	挂载与卸载 CF 卡。

umount <device>	

2. 存储设备格式化操作

命令	解释
特权用户配置模式	
format <device>	格式化存储设备。

3. 子目录的创建操作

命令	解释
特权用户配置模式	
mkdir <directory>	在指定设备的指定目录下创建子目录。

4. 子目录的删除操作

命令	解释
特权用户配置模式	
rmdir <directory>	在指定设备的指定目录下删除子目录。

5. 修改当前存储设备的工作路径

命令	解释
特权用户配置模式	
cd <directory>	修改当前存储设备的工作路径。

6. 显示当前工作路径

命令	解释
特权用户配置模式	
pwd	显示当前工作路径。

7. 显示存储设备中的指定文件或目录信息

命令	解释
特权用户配置模式	
dir [WORD]	显示存储设备中的指定文件或目录信息。

8. 删除文件系统上的指定文件

命令	解释
特权用户配置模式	
delete <file-url>	删除文件系统上的指定文件。

9.文件重命名

命令	解释
特权用户配置模式	
rename <source-file-url> <dest-file >	把交换机上的某个指定的文件重命名为新的文件名。

10.文件复制

命令	解释
特权用户配置模式	
copy <source-file-url > <dest-file-url>	把交换机上的某个指定文件复制为新的文件。

1.3.3 典型应用

将 CF 卡插到交换机的 AM 板卡上,并将当前 AM 板卡上 FLASH 中的 IMG 文件 flash:/nos.img 复制一份到 CF 卡上的 cf:/nos-6.1.11.0.img 文件,然后卸载 CF 卡。

交换机配置如下:

```
Switch#copy flash:/nos.img cf:/nos-6.1.11.0.img
Copy flash:/nos.img to cf:/nos-6.1.11.0.img? [Y:N] y
Copied file flash:/nos.img to cf:/nos-6.1.11.0.img.
Switch#unmount cf
```

1.3.4 排错帮助

当用户进行文件系统操作出现错误时,请检查是否是以下原因所引起的:

- ☞ 文件名或文件路径输入是否正确;CF 卡是否成功挂载;
- ☞ 重命名文件时,该文件是否正在被使用,或目标文件名是否与已有的文件或目录相同。

1.4 集群网管

1.4.1 集群网管介绍

集群网管是一种带内配置管理手段,与常规的 CLI、SNMP、Web Config 不同的是,它借助某台中间交换机(命令交换机)对目标交换机(成员交换机)进行间接配置管理,而不是在管理工作站上对目标交换机进行直接配置管理。命令交换机和成员交换机之间是一对多的关系,在一台命令交换机上可以配置和管理多台成员交换机。只要在命令交换机上配置

公网 IP，集群中所有成员交换机均可不配置公网 IP 也可以被远程配置管理，在 IP 资源日益紧张的今天，集群网管的配置手段节省了 IP 地址资源。而且集群网管做到了动态地发现具有集群功能的交换机（候选交换机），管理员可以根据需要手动或自动把发现的候选交换机加入已建立的集群，成为集群的成员交换机，从而可以通过命令交换机对成员交换机进行所有的配置和管理操作。当多台成员交换机分布在不同的地理位置（典型情况如不同的楼层）时，集群网管的方便性是很明显的。集群网管还有一个优点就是带内管理，即不需要为配置管理成员交换机而建立专门的通信联接，而是直接利用命令和成员之间的中继连线，因而是非常节省网络资源的。

基于集群网管以上特征，集群网管主要有以下优点：

- ☞ 节省 IP 地址
- ☞ 简化配置管理任务
- ☞ 不受网络拓扑结构和距离的限制
- ☞ 自动发现和自动建立功能
- ☞ 在出厂默认配置下就可以通过集群网管管理多台交换机
- ☞ 支持对集群内任意一台交换机的升级和配置管理

1.4.2 集群网管基本配置

集群网管配置任务序列如下：

1. 启动或停止集群功能
2. 建立集群
 - 1) 配置集群中成员设备使用的私有 IP 地址池
 - 2) 创建或删除集群
 - 3) 在集群中加入或删除一个成员
3. 在命令交换机上配置集群相关属性
 - 1) 使能或禁止集群自动加入功能
 - 2) 设置自动加入的 member 成为手动加入的 member
 - 3) 设置或修改集群中的交换机发送保活报文的时间间隔
 - 4) 设置或修改集群内部的保活报文最大允许丢失的个数
 - 5) 清除命令交换机维护的候选交换机列表
4. 在候选交换机上配置集群相关参数
 - 1) 设置集群发送保活报文的时间间隔
 - 2) 设置集群内部保活报文最大允许丢失的个数
5. 利用集群网管进行远程管理
 - 1) 远程配置管理
 - 2) 对成员交换机进行远程升级
 - 3) 重启成员交换机
6. 利用 web 管理集群网管
 - 1) 启动 http
7. 利用 snmp 管理集群网管

1) 启动 snmp server

1. 启动或停止集群功能

命令	解释
全局配置模式	
cluster run [key <WORD>] [vid <VID>] no cluster run	启动或停止交换机本设备的集群功能。

2. 建立集群

命令	解释
全局配置模式	
cluster ip-pool <commander-ip> no cluster ip-pool	配置集群中成员设备使用的私有 IP 地址池。
cluster commander [<cluster_name>] no cluster commander	创建或者删除集群。
cluster member {candidate-sn <candidate-sn> mac-address <mac-addr> [id <member-id>]} no cluster member {id <member-id> mac-address <mac-addr>}	添加或删除集群成员。

3. 在命令交换机上配置集群相关属性

命令	解释
全局配置模式	
cluster auto-add no cluster auto-add	使能或禁止把新发现的候选交换机自动加入集群。
cluster member auto-to-user	把自动加入的成员变成手动加入的成员。
cluster keepalive interval <second> no cluster keepalive interval	设置集群的保活时间。
cluster keepalive loss-count <int> no cluster keepalive loss-count	设置集群中保活报文最大允许丢失个数。
特权模式	
clear cluster nodes [nodes-sn <candidate-sn-list> mac-address <mac-addr>]	清除命令交换机维护的候选交换机列表中的节点。

4. 在候选交换机上配置集群相关参数

命令	解释
全局配置模式	
cluster keepalive interval <second> no cluster keepalive interval	设置集群的保活时间。
cluster keepalive loss-count <int> no cluster keepalive loss-count	设置集群中保活报文最大允许丢失个数。

5. 利用集群网管进行远程管理

命令	解释
特权模式	
rcommand member <member-id>	在命令交换机上使用该命令对集群的成员交换机进行配置管理。
rcommand commander	在成员交换机上使用该命令对命令交换机进行配置管理。
cluster reset member [id <member-id> mac-address <mac-addr>]	在命令交换机上使用该命令重启成员交换机。
cluster update member <member-id> <src-url> <dst-filename> [ascii binary]	在命令交换机上对成员交换机进行远程升级。

6. 利用 web 管理集群网管

命令	解释
全局配置模式	
ip http server	在命令交换机和成员交换机上启动 http 功能。 注：通过 web 从 commander 访问 member 时必须确保 member 上面开启了 http 功能。commander 通过点击其集群拓扑图上 member 节点来访问 member。

7. 利用 snmp 管理集群网管

命令	解释
全局配置模式	

snmp-server enable	在命令交换机和成员交换机上启动 snmp server 功能。 注：通过 snmp 从 commander 访问 member 时必须确保 member 上面开启了 snmp server 功能。 commander 通过设置团体字符串 <commander-community>@sw<member id>的方式访问指定的 member。

1.4.3 集群网管举例

案例：
四台交换机 SW1-SW4，其中指定 SW1 为命令交换机，其余交换机为成员交换机，其中 SW2 和 SW4 和命令交换机直接相连，交换机 SW3 通过交换机 SW2 连接到命令交换机。

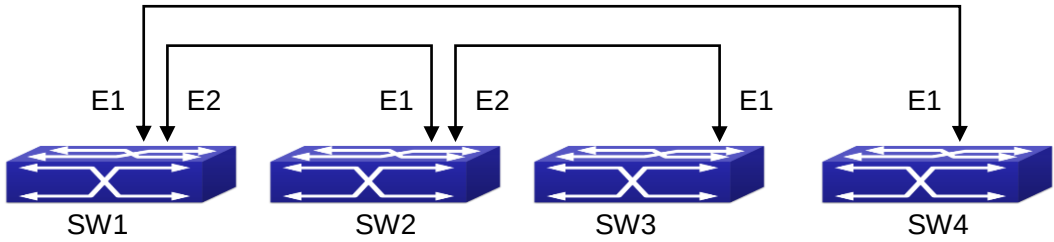


图 4-1 集群网管配置

配置步骤：

1. 配置命令交换机

SW1 的配置：

```
Switch(config)#cluster run
Switch(config)#cluster ip-pool 10.2.3.4
Switch(config)#cluster commander 5526
Switch(config)#cluster auto-add
```

2. 配置成员交换机

SW2-SW4 的配置：

```
Switch(config)#cluster run
```

1.4.4 集群网管排错帮助

当在使用集群网管过程中遇到问题，请检查是否是如下原因：

- 是否 正确配置了命令交换机，并启动了自动加入功能(cluster auto-add)，并且是否命令

交换机和成员交换机相连的端口属于配置的 cluster vlan。

- ☞ 在命令交换机的 VLAN1 启动 cluster commander 后，请不要在该 Vlan 下启动路由协议(rip,ospf,bgp)等，以避免该路由协议将该 Vlan 的私有集群网管地址广播到其它交换机，造成路由环路。
- ☞ 命令交换机和成员交换机之间的连接是否正常，可通过相关的 debug 命令观察交换机能否正确接收和处理相关的集群网管报文。

1.5 设备管理

1.5.1 设备管理简介

交换机的设备管理功能能够向用户提供板卡状态信息、板卡运行调试信息、机架的电源及风扇状态信息。实现对物理设备的维护和管理，并支持整机、单板重新启动，单板的带电插拔。交换机支持双主控，如果在系统中同时存在两块主控板，则所在槽位编号较小的主控板将成为Active Master，另外一块主控板将成为Standby Master。

1.5.2 设备管理配置

1.5.2.1 设备管理配置任务

1. 复位指定板卡
2. 强制主备切换
3. 强制 running-config 主备同步
4. 设置 running-config 同步间隔时长

1.复位指定板卡

命令	解释
特权模式	
reset slot <slotno>	复位指定槽位的板卡，包括线卡和 Standby Master 主控卡。

2.强制主备切换

命令	解释
特权模式	
force switchover	强制进行主备切换，即 Active Master 成为 Standby Master，Standby Master 成为 Active Master。

3.强制 running-config 主备同步

命令	解释
特权模式	
force runcfg-sync	强制将 running-config 从 Active Master 同步到 Standby Master。

4.设置 running-config 同步间隔时长

命令	解释
全局配置模式	
runcfg-sync [<interval>]	配置 running-config 同步间隔时长。

1.5.2.2 监测和调试任务

1. 显示板卡信息
2. 显示风扇状态信息
3. 显示电源状态信息
4. 显示设备管理报文的收发情况和板卡状态转换

1.显示板卡信息

命令	解释
特权模式	
show slot [<slotno>]	显示各板卡基本信息。

2.显示风扇状态信息

命令	解释
特权模式	
show fan	显示风扇板在位状态。

3.显示电源状态信息

命令	解释
特权模式	
show power	显示各电源状态信息。

4.显示设备管理报文的收发情况和板卡状态转换

命令	解释
特权模式	
debug devsm {send receive state}	显示设备管理收发报文的情况和板卡状态转

系统进入正常运行状态之后，用户热插一块板卡，如果此板卡与以前插入到此槽位的板卡类型不一致，则系统不进行配置恢复；如果板卡类型一致，那么系统尝试恢复以前保存的板卡配置，如果以前保存的板卡配置为空，则恢复 StartUp-Config 配置文件中记录的此槽位配置信息。

1.5.4.4 主备切换

交换机支持主备切换功能，即当交换机的两个主控板槽位上插满了主控板，且均处于正常工作状态，交换机支持用户带电更换 active master。用户可以通过 show slot 命令查看两块主控板具体哪一块是 active master，非 active master 的主控卡为 standby master。当用户要进行主备切换时，用户可以按下 active master 的 swap 按钮后将板卡拔出或者直接将 active master 拔出，这样此时的 standby master 就会成为 active master。主备切换之后，交换机的配置将恢复为 startup-config 保存的配置信息。

第2章 二层业务操作

2.1 端口配置

交换机提供线卡和主控卡，线卡提供各种类型的业务端口；主控卡提供 Console 口和网管端口，用于带外管理交换机。现在重点介绍业务端口和网管端口。

业务端口由线卡提供，在交换机各线卡的面板上均标注有各个端口的端口号。为了区分不同线卡上的端口，交换机操作系统提供的端口号（软件意义上的端口号）为 **ethernet X/Y**，其中 X 表示端口所在板卡的插槽号，Y 表示端口在板卡面板上的标号。例如一块 MS700-4XS16GX8GB 线卡插入第 1 槽，它上面的 3 号端口对应 **ethernet 1/3**。如果用户要对某些业务端口进行配置，可以使用命令 **interface ethernet <interface-list>** 进入到相应的以太网接口配置模式，参数 <interface-list> 为一个或多个端口，<interface-list> 包含多个端口时，可以使用 ‘;’ 和 ‘-’ 等特殊字符进行连接，‘;’ 连接不连续的端口号，‘-’ 连接连续的端口号。例如要同时对 1 号插槽板卡的 2,3,4,5 端口和 3 号插槽的 8,9,10 端口进行操作，可以使用命令 **interface ethernet 1/2-5;3/8-10**。在以太网接口配置模式下可对端口的速率、双工模式、流量控制等进行配置，相应业务端口的表现也随之改变。

网管端口位于主控卡上，标注为 **Ethernet**，软件配置名称为 **Ethernet0**，用户可以使用命令 **interface Ethernet 0** 进入到网管端口配置模式。通过以太网线连接到网管端口的用户可以以 Telnet、Web 等方式对交换机进行配置管理。网管端口不能像业务端口进行数据转发，不能配置二层协议（如 RSTP）和三层路由协议，也不支持自动识别连线类型，如与 PC 机直接连接需使用交叉线。网管端口支持 10/100Mb 连接速度，除了可以配置速率、双工等物理属性外，还可以配置 IP 地址。如果交换机有两块主控卡均工作在正常状态，只有 **Active Master** 的以太网接口可以作为网管端口，当发生主备切换，网管端口也相应发生变化。

2.1.1 业务端口配置

业务端口配置任务序列如下：

1. 进入业务端口配置模式
2. 配置业务端口的属性
 - (1) 配置组合端口的组合模式
 - (2) 打开或关闭端口
 - (3) 配置端口名字
 - (4) 配置端口连线类型
 - (5) 配置端口速率和双工模式
 - (6) 配置带宽控制
 - (7) 配置流量控制

- (8) 打开或关闭端口环回功能
- (9) 配置交换机广播风暴抑制功能
- (10) 设置机架式交换机线卡端口流量统计间隔时间
- (11) 配置扫描端口方式
- (12) 配置端口速率违背控制
- (13) 配置端口速率设置统计间隔时间

3. 虚拟线路检测

1. 进入以太网端口配置模式

命令	解释
全局配置模式	
interface ethernet <interface-list>	进入业务端口配置模式。

2. 配置以太网端口属性

命令	解释
端口配置模式	
media-type { copper copper-preferred-auto fiber sfp-preferred-auto }	设定组合端口模式（仅限组合端口）。
shutdown no shutdown	关闭或打开指定端口。
description <string> no description	设定或取消指定端口的名字。
speed-duplex {auto [10 [100 [1000]] [auto full half]] force10-half force10-full force100-half force100-full force100-fx [module-type {auto-detected no-phy-integrated phy-integrated}} {force1g-half force1g-full} force10g-full} no speed-duplex	设置 1000Base-TX、100Base-TX 或 100Base-FX 端口的速率和双工模式。本命令的 no 操作为恢复默认设置，即自动协商速度双工。
negotiation {on off}	打开或关闭 1000Base-FX 端口的自动协商。
bandwidth control <bandwidth> [both receive transmit] no bandwidth control	设置或取消指定端口收、发数据占用的带宽。

flow control no flow control	打开或关闭指定端口的流量控制功能。
loopback no loopback	打开或关闭指定端口的环回测试功能。
storm-control {unicast broadcast multicast} <packets>	打开交换机的广播风暴抑制（或组播、未知单播，下同）功能，并设置每秒允许通过的广播包数量；本命令的 no 操作为取消广播风暴抑制功能。
remote-statistics interval <seconds>	设置机架式交换机线卡端口流量统计间隔时间。
port-scan-mode {interrupt poll} no port-scan-mode	配置扫描端口方式为中断或查询方式。本命令的 no 操作恢复默认的端口扫描方式。
rate-violation <200-2000000> [recovery <0-86400>] no rate-violation	设置交换机端口的报文最大收包速率，如果端口的接收报文速率违背了收包速率则 shutdown 端口，可以配置 shutdown 端口后的恢复时间，默认是 300s。本命令的 no 操作为关闭端口的速率违背功能。
Interface mode <auto-detect cr cr4 sfi sr sr4 xfi >	配置 40G 端口的模式，默认为 auto-detect 模式 注：本交换机不支持此命令。
全局配置模式	
port-rate-statistics interval <interval -value>	配置 port-rate-statistics 统计间隔时间。

3. 虚拟线路检测

命令	解释
特权模式	
virtual-cable-test interface (ethernet)IFNAME	进行端口的虚拟线路检测。
端口模式	
virtual-cable-test trap enable no virtual-cable-test trap enable	端口 link 状态由 up 变为 down 时，进行虚拟线路检测，将检测结果发送 snmp trap 到服务器，默认不检测。本命令的 no 操作取消检测。

2.1.2 端口配置举例

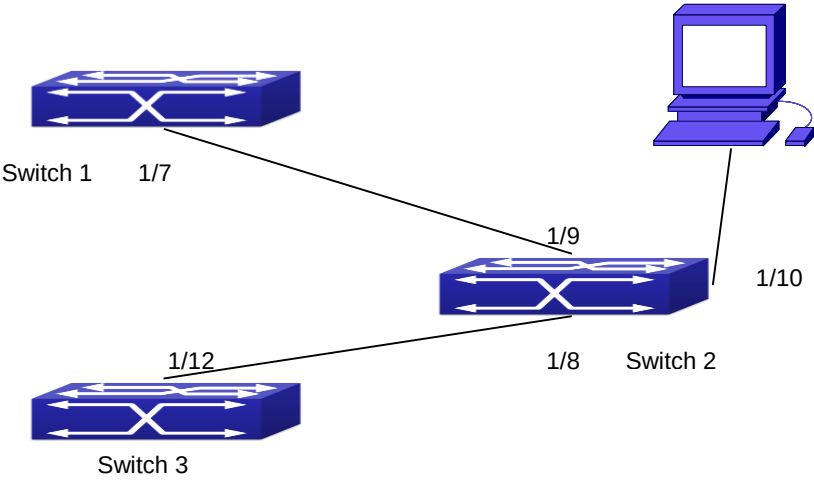


图 2-1 端口配置举例

交换机中均未配置 VLAN，因此使用缺省 VLAN1。

交换机	端口	属性
SW1	1/7	入口带宽限制，50M
SW2	1/8	端口镜像源端口
	1/9	100M/full、端口镜像源端口
	1/10	1000M/full、端口镜像目的端口
SW3	1/12	100M/full

配置如下：

```
SW1:
Switch1(config)#interface ethernet 1/7
Switch1(Config-If-Ethernet1/7)#bandwidth control 50000 both
SW2:
Switch2(config)#interface ethernet 1/9
Switch2(Config-If-Ethernet1/9)#speed-duplex force100-full
Switch2(Config-If-Ethernet1/9)#exit
Switch2(config)#interface ethernet 1/10
Switch2(Config-If-Ethernet1/10)#speed-duplex force1g-full
Switch2(Config-If-Ethernet1/10)#exit
Switch2(config)#monitor session 1 source interface ethernet 1/8;1/9
Switch2(config)#monitor session 1 destination interface ethernet 1/10
SW3:
Switch3(config)#interface ethernet 1/12
Switch3(Config-If-Ethernet1/12)#speed-duplex force100-full
Switch3(Config-If-Ethernet1/12)#exit
```

2.1.3 端口排错帮助

用户在进行端口配置时通常会遇到的情况及解决建议如下：

- ☞ 两个光口互相连接时，如果一端设置为自动协商，另一端设置强制速率/双工，则光口不会 Link up。这是由 IEEE 802.3 协议决定的。
- ☞ 建议用户不要进行以下的组合设置：打开某端口流控，同时设置该端口组播抑制；设置某端口的广播、组播或未知单播抑制，同时设置端口带宽限制。如果在端口进行这些组合设置，可能导致端口流量低于期望值。

2.2 端口环路检测

2.2.1 端口环路检测功能简介

随着交换机的发展，用户通过以太网交换机接入网络越来越多。在企业网中，用户通过二层以太网交换机接入网络，他们不仅有上 internet 的需求，同时内部二层互通的需求也相当迫切。当用户需要二层互通时，报文的转发直接通过 MAC 寻址，MAC 地址学习的正确与否决定着用户之间是否能够正确的互通。在二层交换中，通过 MAC 地址寻址来进行报文转发。二层设备的 MAC 地址学习都是通过源 MAC 地址学习来进行的。即：当端口收到一个未知源 MAC 地址的报文，会将这个 MAC 添加到接收端口上，以便后续以该 MAC 地址为目的的报文能够直接转发，即一次学习，多次转发。

当新来源 MAC 如果发现该 MAC 已经学习到了二层设备上，而源端口不一样，会修改原来 MAC 地址的源端口，也就是将原来的 MAC 地址移动到新的端口上来。因此当链路上存在环回情况时，最后会发现整个二层网络中的所有的 MAC 地址都移动到了存在环回的端口上了（大多数的情况是 MAC 地址频繁在不同端口间切换），导致二层网络瘫痪。在网络中进行端口环路检测非常必要，具有重要的意义。当设备通过环路检测发现了网络存在环回情况时，可以通过发送告警信息到网管系统，使网络管理人员能够及时发现网络中存在的问题，从而及时定位和解决。避免长时间的用户断网现象。

因为环路检测可以动态的发现链路上是否存在环回以及链路上的环回是否消失，因此，对于支持端口受控（比如端口隔离，端口 MAC 地址学习受控）的设备，可以实现自动维护。这样不仅减轻网管人员的工作负担，同时反应更加及时，能迅速将环回对网络的影响减小至最小。

2.2.2 端口环路检测功能配置任务序列

1. 配置环路检测的时间间隔
2. 启动端口环路检测功能
3. 配置端口环路控制方式

4. 显示和调试端口环路检测相关信息
5. 配置环路检测受控方式是否自动恢复

1. 配置环路检测的时间间隔

命令	解释
全局配置模式	
loopback-detection interval-time <loopback> <no-loopback> no loopback-detection interval-time	设置或恢复环路检测的时间间隔。

2. 启动端口环路检测功能

命令	解释
端口配置模式	
loopback-detection specified-vlan <vlan-list> no loopback-detection specified-vlan <vlan-list>	启动和关闭端口环路检测功能。

3. 配置端口环路控制方式

命令	解释
端口配置模式	
loopback-detection control {shutdown block } no loopback-detection control	打开和关闭端口的环路检测受控功能。

4. 显示和调试端口环路检测相关信息

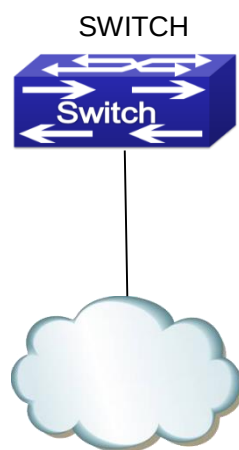
命令	解释
特权配置模式	
debug loopback-detection no debug loopback-detection	打开端口环路检测功能模块的调试信息，本命令的 NO 操作关闭调试信息输出。
show loopback-detection [interface <interface-list>]	不输入参数则显示所有端口的环路检测状态和检测结果，输入参数则显示相应端口的状态和结果。

5. 配置环路检测受控方式是否自动恢复

命令	解释

全局配置模式	
loopback-detection control-recovery timeout <0-3600>	配置环路检测受控方式是否自动恢复或者恢复的时间间隔。

2.2.3 端口环路检测典型案例



网络拓扑

图 2-2 端口环路检测典型配置案例

上述配置中，交换机检测网络拓扑中是否存在环路情况。在交换机与外部网路连接的端口上启动端口环路检测功能，如果外部网络中存在环路的情况，交换机就会提示下连的网络存在环路，并将交换机上的这个端口控制，以免影响整个网络的正常工作。

SWITCH 配置任务序列：

```
Switch (config)#loopback-detection interval-time 35 15
```

```
Switch (config)#int ethernet 1/1
```

```
Switch (Config-If-Ethernet1/1)#loopback-detection special-vlan 1-3
```

```
Switch (Config-If-Ethernet1/1)#loopback-detection control block
```

如果使用 block 控制，必须全局启动 mstp，并且配置生成树实例与 vlan 的对应关系

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

```
Switch(config)#spanning-tree mst configuration
```

```
Switch(Config-Mstp-Region)#instance 1 vlan 1
```

```
Switch(Config-Mstp-Region)#instance 2 vlan 2
```

```
Switch(Config-Mstp-Region)#
```

2.2.4 端口环路检测排错帮助

端口环路检测功能默认情况下是关闭的，如果需要检测环路可以打开该功能。

2.3 ULDP

2.3.1 ULDP 功能简介

单向链路是网络中常见的一种错误链路状态，在光纤连接的链路中尤为常见。所谓单向链路，是指链路两端的端口之一可以收到对端发送的数据链路层报文，而对端不能收到本端发送的报文。此时链路物理层处于是连通状态，能正常工作，因而物理层的检测机制无法发现设备间通信存在问题。如下图所示，图中所示的光纤连接的错误就无法通过物理层的自动协商等机制发现。

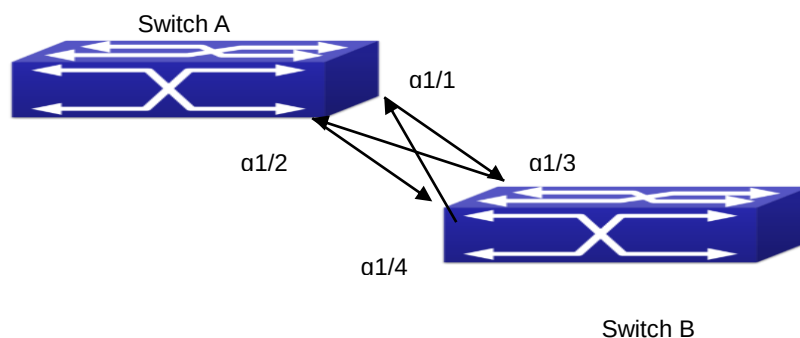


图 2-3 光纤交叉连接

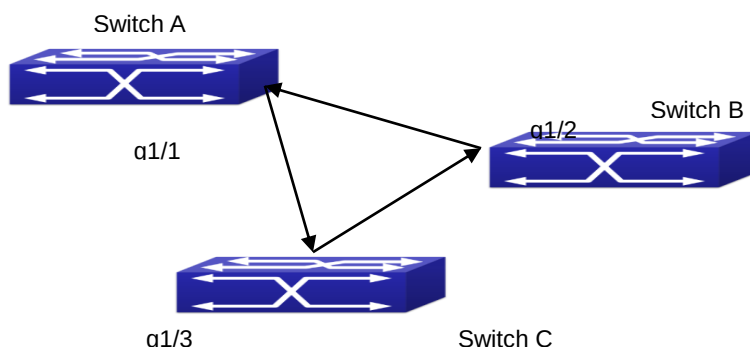


图 2-4 光纤一端未连接

这种情形通常会出现在以下几种情况之中：GBIC（Giga Bitrate Interface Converter，吉比特接口转换器）或接口出现故障、软件故障、硬件失效或其他异常表现。单向链路会引

起一系列问题，比如生成树拓扑环路，广播黑洞等。

ULDP（UniDirectional Link Detection Protocol）有助于防止在上述几种失效中发生灾难性的事件，在通过光纤或铜质以太网线（例如超五类双绞线）连接的交换机上，它可以监控物理线路的链路状态，当发现单向链路后，向用户发送告警信息，并根据用户配置，自动或者手动地关闭相关端口。

神州数码系列交换机 ULDP 是通过交互 ULDP 报文识别对端设备，检测链路链接的正确性。端口使能了 ULDP 后，将启动协议状态机，在状态机的不同状态下发送不同类型的报文，与对端交互信息来检测链路的连接状况。ULDP 可以动态的学习对端的通告报文时间间隔，并以此来调整对端信息在本端的存活时间。此外，ULDP 还提供了重启机制，当端口被 ULDP 关闭后，可以通过重启机制来重新检测。其中，ULDP 发送通告报文的时间间隔和重启时间间隔可由用户配置，以便 ULDP 在不同的网络环境对链路的连接错误做出更快的响应。

ULDP 能正确工作的前提是链路工作在全双工方式，链路两端都使能了 ULDP 功能，认证方式和密码相同。

2.3.2 ULDP 配置任务序列

1. 启动全局 ULDP 功能
2. 启用端口 ULDP 功能
3. 配置全局积极模式
4. 配置端口积极模式
5. 配置关闭单向链路方式
6. 配置 Hello 报文时间间隔
7. 配置 Recovery 时间间隔
8. 重启被 ULDP 关闭的端口
9. 显示和调试 ULDP 相关信息

1. 启动全局 ULDP 功能

命令	解释
全局配置模式	
uldp enable uldp disable	全局启动或关闭 ULDP 功能。

2. 启动端口 ULDP 功能

命令	解释
端口配置模式	

uldp enable uldp disable	端口启动或关闭 ULDP 功能。

3. 配置全局积极模式

命令	解释
全局配置模式	
uldp aggressive-mode no uldp aggressive-mode	设置全局工作模式。

4. 配置端口积极模式

命令	解释
端口配置模式	
uldp aggressive-mode no uldp aggressive-mode	设置端口工作模式。

5. 配置关闭单向链路方式

命令	解释
全局配置模式	
uldp manual-shutdown no uldp manual-shutdown	配置关闭单向链路方式。

6. 配置 Hello 报文时间间隔

命令	解释
全局配置模式	
uldp hello-interval <integer> no uldp hello-interval	配置 Hello 报文时间间隔范围为 5—100 秒，默认为 10 秒。

7. 配置 Recover 时间间隔

命令	解释
全局配置模式	
uldp recovery-time <integer> no uldp recovery-time <integer>	配置端口重启时间间隔范围为 30—86400 秒，默认为 0 秒。

8. 重启被 ULDP 关闭的端口

命令	解释
全局配置模式或端口配置模式	

uldp reset	全局模式下重启所有端口； 端口模式下重启指定端口。

9. 显示和调试 ULDP 相关信息

命令	解释
特权模式	
show uldp [interface ethernet IFNAME]	显示 ULDP 信息。不带端口参数显示全局 ULDP 信息。带端口参数显示全局信息和该端口邻居信息。
debug uldp fsm interface ethernet <IFname> no debug uldp fsm interface ethernet <IFname>	打开或关闭指定端口的状态机迁移信息调试项。
debug uldp error no debug uldp error	打开或关闭错误信息调试项。
debug uldp event no debug uldp event	打开或关闭事件信息调试项。
debug uldp packet {receive send} no debug uldp packet {receive send}	打开或关闭所有端口收发报文类型。
debug uldp {hello probe echo unidir all} [receive send] interface ethernet <IFname> no debug uldp {hello probe echo unidir all} [receive send] interface ethernet <IFname>	打开或关闭特定端口收发的某种类型报文的详细内容。

2.3.3 ULDP 功能典型案例

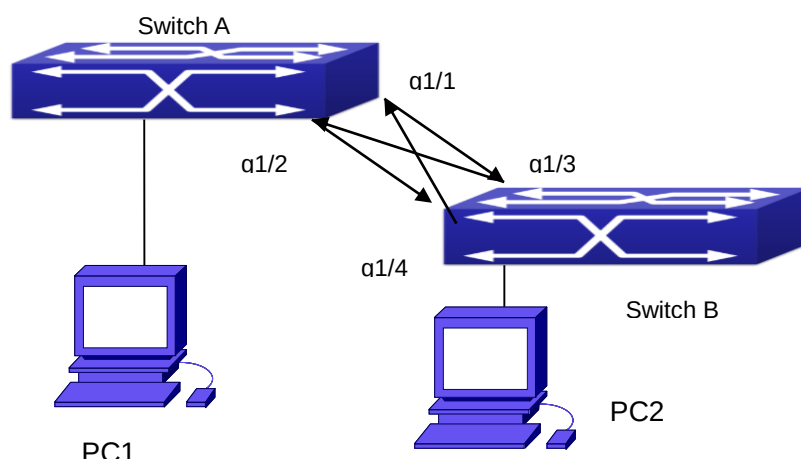


图 2-5 光纤交叉连接

在上述网络拓扑图中，SWITCH A 的端口 g1/1, g1/2 和 SWITCH B 的端口 g1/3, g1/4 均为光口，该连接为交叉连接。此时物理层连通，能工作正常，但是数据链路层异常。ULDP 可以检测并且关闭这种错误链路状态。最终的结果是 SWITCH A 的端口 g1/1, g1/2 和 SWITCH B 的端口 g1/3, g1/4 均被 ULDP 关闭。只有连接正确才能正常工作（不会被关闭）。

Switch A配置序列：

```
SwitchA(config)#uldp enable
SwitchA(config)#interface ethernet 1/1
SwitchA(Config-If-Ethernet1/1)#uldp enable
SwitchA(Config-If-Ethernet1/1)#exit
SwitchA(config)#interface ethernet 1/2
SwitchA(Config-If-Ethernet1/2)#uldp enable
```

Switch B配置序列：

```
SwitchB(config)#uldp enable
SwitchB(config)#interface ethernet 1/3
SwitchB(Config-If-Ethernet1/3)#uldp enable
SwitchB(Config-If-Ethernet1/3)#exit
SwitchB(config)#interface ethernet 1/4
SwitchB(Config-If-Ethernet1/4)#uldp enable
```

最终，SWITCH A的端口g1/1, g1/2均被ULDP关闭，在PC1的CRT终端均有提示信息：

```
%Oct 29 11:09:50 2007 A unidirectional link is detected! Port Ethernet1/1 need to
be shutted down!
```

```
%Oct 29 11:09:50 2007 Unidirectional port Ethernet1/1 shutted down!
```

```
%Oct 29 11:09:50 2007 A unidirectional link is detected! Port Ethernet1/2 need to
```

be shutted down!

%Oct 29 11:09:50 2007 Unidirectional port Ethernet1/2 shutted down!

SWITCH B的端口g1/3, g1/4均被ULDP关闭, 在PC2的CRT终端均有提示信息:

%Oct 29 11:09:50 2007 A unidirectional link is detected! Port Ethernet1/3 need to be shutted down!

%Oct 29 11:09:50 2007 Unidirectional port Ethernet1/3 shutted down!

%Oct 29 11:09:50 2007 A unidirectional link is detected! Port Ethernet1/4 need to be shutted down!

%Oct 29 11:09:50 2007 Unidirectional port Ethernet1/4 shutted down!

2.3.4 ULDP 排错帮助

配置注意事项:

- ☞ 为了使得 ULDP 能够监测到光纤口上的一端未连或是交叉错连, 端口必须为全双工模式, 且速率相同。
- ☞ 当一端错连的光纤端口的自动协商机制在决定端口的工作模式和速度时, 即使 ULDP 已经使能, 也不起作用。在这种情况下, 认为该端口是 Down 的。
- ☞ 为了确保能正确的建立邻居和监测单向链路, 要求: 链路两端都使能 ULDP, 且认证模式和密码相同, 目前两端均不需验证密码。
- ☞ 发送 hello 报文的 hello interval 间隔可调 (默认 10 秒, 可调范围 5-100 秒), 以便根据不同的网络环境使 ULDP 对链路连接错误做出更快的响应。但是这个时间间隔必须小于 1/3 的 STP 收敛时间, 如果设置的时间过长, 在 ULDP 发现并关闭单向连接端口之前 STP loop 已经形成。反之, 如果设置的时间间隔太短, 端口的网络负担会增加, 带宽会减少。
- ☞ ULDP 不处理任何 LACP 事件, 把汇聚组 (如 Port-channel, trunk 端口等) 的每个链路看作是独立的, 分别进行处理。
- ☞ ULDP 不能和其它厂商相似的协议兼容, 也就是说, 不能在一端使能 ULDP, 而在另一端使能其他的相似的协议。
- ☞ ULDP 功能默认是关闭的。打开全局 ULDP 功能后, 可以同时打开调试开关来查看调试信息。设计了多条 DEBUG 命令来打印调试信息, 如事件, 状态机, 错误, 报文信息。对于报文信息, 还可以分不同参数来打印不同类型的报文信息。
- ☞ Recovery 定时器 timer 默认不启用。只有用户配置了 recovery time (30-86400 秒) 才启用。
- ☞ reset 命令和重启机制只能重启 ULDP 自动关闭的端口, 被用户手工关闭的或是被其他模块关闭的端口不会被 ULDP 重新启动。

2.4 LLDP

2.4.1 LLDP 功能简介

链路层发现协议 (Link Layer Discovery Protocol, LLDP) 是 802.1ab 中定义的新协议, 它可使邻近设备向其他设备发出其状态信息的通知, 并且所有设备的每个端口上都存储着定义自己的信息, 如果需要还可以向与它们直接连接的近邻设备发送更新的信息, 近邻的设备会将信息存储在标准的 SNMP MIBs。网络管理系统可从 MIB 处查询出当前第二层的连接情况。LLDP 不会配置也不会控制网络元素或流量, 它只是报告第二层的配置。802.1ab 中的另一个内容是使网络管理软件利用 LLDP 所提供的信息去发现某些第二层的矛盾之处。IEEE 目前使用的是 IETF 现有的物理拓扑、接口和 Entity MIBs。

简单说来, LLDP 是一种邻近发现协议。它为以太网网络设备, 如交换机、路由器和无线局域网接入点定义了一种标准的方法, 使其可以向网络中其他节点公告自身的存在, 并保存各个邻近设备的发现信息。例如设备配置和设备识别等详细信息都可以用该协议进行公告。

具体来说, LLDP 定义了一个通用公告信息集、一个传输公告的协议和一种用来存储所收到的公告信息的方法。要公告自身信息的设备可以将多条公告信息放在一个局域网数据包内传输, 传输的形式为类型长度值 (TLV) 域。所有具备 LLDP 能力的设备必须支持设备机身 ID 和端口 ID 公告, 但根据预计, 多数设备还要支持系统名称、系统描述和系统能力公告。系统名称和系统描述公告可以为收集网络流量数据提供非常有用的信息。系统描述公告可以包含诸如公告设备的全名和系统的硬件类型及软件操作系统的版本信息等数据。

802.1AB 链接层发现协议 (Link Layer Discovery Protocol), 将能够使企业网络的故障查找变得更加容易, 并加强网络管理工具在多厂商环境中发现和保持精确网络拓扑结构的能力。

许多网络管理软件都使用“自动发现”功能 (Automated Discovery) 来跟踪拓扑的变化和条件, 但绝大多数软件最多也只是到达第三层, 将设备分组到各个 IP 子网而已。但这些都是非常原始的数据, 只是有关设备增加和移除的基本事件, 而不是这些设备在哪里或怎样与网络服务进行操作等详情。

第二层发现 (Layer 2 Discovery) 则深度触及了诸如哪些设备附带有哪端口, 以及哪些交换机与其他设备相互连接等信息, 并显示出了客户端、交换机、路由器和应用服务器以及网络服务器之间的路径。如此详细的信息对计划和查询网络失败的根源将很有意义。

LLDP 将成为一种非常有用的管理工具, 提供精确的网络映射、流量数据和网络故障查找信息。

2.4.2 LLDP 功能配置任务序列

1. 全局启动 LLDP 功能

2. 配置基于端口的 LLDP 功能开关
3. 配置端口 LLDP 运行状态
4. 配置 LLDP 更新报文发送间隔
5. 配置 LLDP 报文老化时间乘法器
6. 配置更新报文发送延迟
7. 配置 Trap 报文发送间隔
8. 配置端口使能 Trap 功能
9. 配置端口发送信息可选属性
10. 配置端口 Remote Table 存储空间大小
11. 配置端口 Remote Table 满时的操作类型
12. 显示和调试 LLDP 功能相关信息

1. 全局启动 LLDP 功能

命令	解释
全局配置模式	
lldp enable lldp disable	全局启动或关闭 LLDP 功能。

2. 配置基于端口的 LLDP 功能开关

命令	解释
端口配置模式	
lldp enable lldp disable	设置基于端口的 LLDP 功能开关。

3. 配置端口 LLDP 运行状态

命令	解释
端口配置模式	
lldp mode{send receive both disable}	设置指定端口 LLDP 运行状态。

4. 配置 LLDP 更新报文发送间隔

命令	解释
全局配置模式	
lldp tx-interval <integer> no lldp tx-interval	设置 LLDP 更新报文发送间隔为配置值或默认值。

5. 配置 LLDP 报文老化时间乘法器

命令	解释
全局配置模式	
lldp msgTxHold <value> no lldp msgTxHold	设置 LLDP 报文老化时间乘法器为配置值或默认值。

6. 配置更新报文发送延迟

命令	解释
全局配置模式	
lldp transmit delay <seconds> no lldp transmit delay	设置 LLDP 更新报文发送延迟为配置值或默认值。

7. 配置 Trap 报文发送间隔

命令	解释
全局配置模式	
lldp notification interval <seconds> no lldp notification interval	设置 Trap 报文发送间隔为配置值或默认值。

8. 配置端口使能 Trap 功能

命令	解释
端口配置模式	
lldp trap <enable disable>	设置端口使能 Trap 功能打开或关闭。

9. 配置端口发送信息可选属性

命令	解释
端口配置模式	
lldp transmit optional tlv [portDesc] [sysName] [sysDesc] [sysCap] no lldp transmit optional tlv	设置端口发送信息可选属性为选择值或默认值。

10. 配置端口 Remote Table 存储空间大小

命令	解释
端口配置模式	
lldp neighbors max-num <value> no lldp neighbors max-num	设置 Remote Table 存储空间大小为配置值或默认值。

11. 配置端口 Remote Table 满时的操作类型

命令	解释
端口配置模式	
lldp tooManyNeighbors {discard delete}	设置当 Remote Table 满时的操作类型。

12. 显示和调试 LLDP 功能相关信息

命令	解释
特权，全局配置模式	
show lldp	显示当前 LLDP 配置信息。
show lldp interface ethernet <IFNAME>	显示当前端口 LLDP 配置信息。
show lldp traffic	显示各种计数器信息。
show lldp neighbors interface ethernet <IFNAME>	显示当端口 LLDP 的邻居信息。
show debugging lldp	显示所有开启 LLDP debug 开关的端口。
特权模式	
debug lldp no debug lldp	DEBUG 开关打开或关闭。
debug lldp packets interface ethernet <IFNAME> no debug lldp packets interface ethernet <IFNAME>	端口或全局 DEBUG 收发包信息功能打开或关闭。
端口配置模式	
clear lldp remote-table	清空端口 Remote-table。

2.4.3 LLDP 功能典型案例

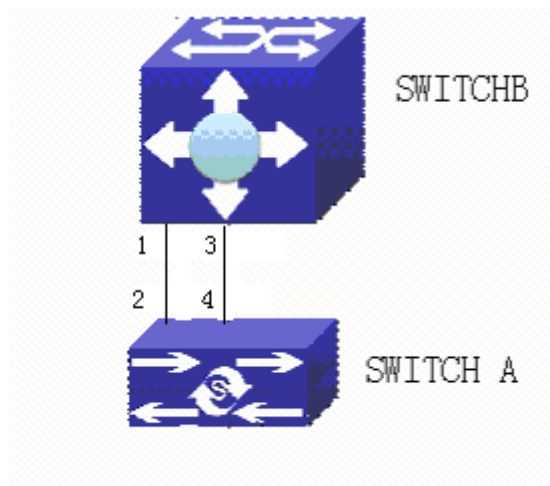


图 2-6 LLDP 功能典型配置案例

在上述网络拓扑图中, SWITCH B 的端口 1、3 与 SWITCH A 的端口 2、4 相连, SWITCH B 端口 1 配置了 LLDP 只接收报文模式, SWITCH A 的端口 4 配置了 Option TLV 为 portDes 和 SysCap。

SWITCH A 配置任务序列:

```
SwitchA(config)#lldp enable
```

```
SwitchA(config)#interface ethernet 1/4
```

```
SwitchA(Config-If-Ethernet1/4)#lldp transmit optional tlv portDesc sysCap
```

```
SwitchA(Config-If-Ethernet1/4)#exit
```

SWITCH B 配置任务序列:

```
SwitchB(config)#lldp enable
```

```
SwitchB(config)#interface ethernet1/1
```

```
SwitchB(Config-If-Ethernet1/1)#lldp mode receive
```

```
SwitchB(Config-If-Ethernet1/1)#exit
```

2.4.4 LLDP 功能排错帮助

- ☞ LLDP 功能默认是关闭的。打开 LLDP 全局开关后, 可以同时打开调试关 debug lldp 来查看调试信息。
- ☞ 使用 LLDP 功能的 show 功能可以看到全局或端口的配置信息。

2.5 Port Channel

2.5.1 Port Channel 介绍

在介绍 Port Channel 之前，先介绍一下 Port Group 的概念：Port Group 是配置层面上的一个物理端口组，配置到 Port Group 里面的物理端口才可以参加链路汇聚，并成为 Port Channel 里的某个成员端口。在逻辑上，Port Group 并不是一个端口，而是一个端口序列。加入 Port Group 中的物理端口满足某种条件时进行端口汇聚，形成一个 Port Channel，这个 Port Channel 具备了逻辑端口的属性，才真正成为一个独立的逻辑端口。端口汇聚是一种逻辑上的抽象过程，将一组具备相同属性的端口序列，抽象成一个逻辑端口。Port Channel 是一组物理端口的集合体，在逻辑上被当作一个物理端口。对用户来讲，完全可以将这个 Port Channel 当作一个端口使用，因此不仅能增加网络的带宽，还能提供链路的备份功能。端口汇聚功能通常在交换机连接路由器、主机或者其他交换机时使用。

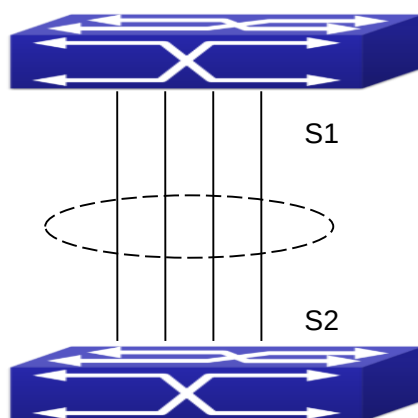


图 2-7 端口聚合示意图

如上图显示交换机 S1 的 1-4 号端口汇聚成一个 Port Channel，该 Port Channel 的带宽为 4 个端口带宽的总和。而 S1 如果有流量要经过 Port Channel 传输到 S2，S1 的 Port Channel 将根据流量的源 MAC 地址及目的 MAC 地址的最低位进行流量分配运算，根据运算结果决定由 Port Channel 中的某一成员端口承担该流量。当 Port Channel 中的一个端口连接失败，应该由该端口承担的流量将再次通过流量分配算法分配给其他连接正常的端口分担。流量分配算法由交换机的硬件决定。

交换机提供了两种配置端口汇聚的方法：手工生成 Port Channel、LACP（Link Aggregation Control Protocol）动态生成 Port Channel。只有双工模式为全双工模式的端口才能进行端口汇聚。

为使 Port Channel 正常工作，本交换机 Port Channel 的成员端口必须具备以下相同的属性：

- ☞ 端口均为全双工模式；
- ☞ 端口速率相同；
- ☞ 端口同为 Access 端口并且属于同一个 VLAN 或同为 Trunk 端口或同为 hybrid 端口；
- ☞ 如果端口同为 Trunk 端口或同为 hybrid 端口，则其 Allowed VLAN 和 Native VLAN 属性也应该相同。

当交换机通过手工方式配置 Port Channel 或 LACP 方式动态生成 Port Channel，系统将自动选举出 Port Channel 中端口号最小的端口作为 Port Channel 的主端口(Master Port)。若交换机打开 Spanning-tree 功能，Spanning-tree 视 Port Channel 为一个逻辑端口，并且由主端口发送 BPDU 帧。

另外，端口汇聚功能的实现与交换机所使用的硬件有密切关系，本交换机支持任意两个相同属性的物理端口的汇聚，最大组数为 128 个，组内最多的端口数为 8 个。

汇聚端口一旦汇聚成功就可以把它当成一个普通的端口使用，在交换机中还建立了汇聚接口配置模式，与 VLAN 和物理接口配置模式一样，用户能在汇聚接口配置模式下对汇聚接口进行相关的配置。

2.5.2 LACP 简介

基于 IEEE802.3ad 标准的 LACP (Link Aggregation Control Protocol，链路汇聚控制协议)是一种实现链路动态汇聚的协议。LACP 协议通过 LACPDU(Link Aggregation Control Protocol Data Unit，链路汇聚控制协议数据单元)与对端交互信息。

使能某端口的 LACP 协议后，该端口将通过发送 LACPDU 向对端通告自己的系统优先级、系统 MAC 地址、端口优先级、端口号和操作 Key。对端接收到这些信息后，将这些信息与其它端口所保存的信息比较以选择能够汇聚的端口，从而双方可以对端口加入或退出某个动态汇聚组达成一致。

操作 Key 是在端口汇聚时，LACP 协议根据端口的配置（即速率、双工、基本配置、管理 Key）生成的一个配置组合。

动态汇聚端口在使能 LACP 协议后，其管理 Key 缺省为零。静态汇聚端口在使能 LACP 后，端口的管理 Key 与汇聚组 ID 相同。

对于动态汇聚组而言，同组成员一定有相同的操作 Key，而手工和静态汇聚组中，Active 的端口有相同的操作 Key。

端口汇聚是将多个端口汇聚在一起形成一个汇聚组，以实现出/入负荷在汇聚组中各个成员端口中的分担，同时也提供了更高的连接可靠性。

2.5.2.1 静态 LACP 汇聚

静态 LACP 汇聚由用户手工配置，不使能 LACP 协议，在配置时是使用 on 模式强制端口进入汇聚组。

2.5.2.2 动态 LACP 汇聚

1. 动态 LACP 汇聚概述

动态 LACP 汇聚是一种系统自动创建/删除的汇聚，不允许用户增加或删除动态 LACP 汇聚中的成员端口。只有速率和双工属性相同、连接到同一个设备、有相同基本配置的端口

才能被动态汇聚在一起。即使只有一个端口也可以创建动态汇聚，此时为单端口汇聚。动态汇聚中，端口的 LACP 协议处于使能状态。

2. 动态汇聚组中的端口状态

在动态汇聚组中，端口可能处于两种状态：**selected** 或 **standby**。**selected** 端口和 **standby** 端口都能收发 LACP 协议，但 **standby** 端口不能转发用户报文。

由于设备所能支持的汇聚组中的最大端口数有限制，如果当前的成员端口数量超过了最大端口数的限制，则本端系统和对端系统会进行协商，根据设备 ID 优的一端的端口 ID 的大小，来决定端口的状态。具体协商步骤如下：

比较设备 ID（系统优先级+系统 MAC 地址）。先比较系统优先级，如果相同再比较系统 MAC 地址。设备 ID 小的一端被认为优。

比较端口 ID（端口优先级+端口号）。对于设备 ID 优的一端的各个端口，首先比较端口优先级，如果优先级相同再比较端口号。端口 ID 小的端口为 **selected** 端口，剩余端口为 **standby** 端口。

在一个汇聚组中，处于 **selected** 状态且端口号最小的端口为汇聚组的主端口，其他处于 **selected** 状态的端口为汇聚组的成员端口。

2.5.3 Port Channel 配置任务序列

1. 全局模式下建立一个 port group
2. 分别在端口模式下将这些端口加入指定的 group
3. 进入 port-channel 配置模式
4. 设置 port-group 的 load-balance 方式
5. 设置 LACP 协议的系统优先级
6. 设置 LACP 协议中当前端口的端口优先级
7. 设置 LACP 协议中当前端口的 Timeout 模式

1.创建 port group

命令	解释
全局配置模式	
port-group <port-group-number> no port-group <port-group-number>	创建或删除一个 port group。

2.把物理端口加入 port group

命令	解释
端口配置模式	
port-group <port-group-number> mode {active passive on}	把端口加入 port group，并设置模式。

no port-group	

3.进入 port-channel 配置模式

命令	解释
全局配置模式	
interface port-channel <port-channel-number>	进入 port-channel 配置模式。

4.设置 port-group 的 load-balance 方式

命令	解释
汇聚端口配置模式	
load-balance {src-mac dst-mac dst-src-mac src-ip dst-ip dst-src-ip}	设置 port-group 的 load-balance。

5.设置 LACP 协议的系统优先级

命令	解释
全局配置模式	
lacp system-priority <system-priority> no lacp system-priority	设置 LACP 协议的系统优先级, no 命令表示恢复为默认值。

6.设置 LACP 协议中当前端口的端口优先级

命令	解释
端口配置模式	
lacp port-priority <port-priority> no lacp port-priority	设置当前端口 LACP 协议中的端口优先级, no 命令表示恢复为默认值。

7. 设置 LACP 协议中当前端口的 Timeout 模式

命令	解释
端口配置模式	
lacp timeout {short long} no lacp timeout	设置当前端口 LACP 协议中的端口 Timeout 模式, no 命令表示恢复为默认值。

2.5.4 Port Channel 举例

案例 1：以 LACP 方式配置 Port Channel。

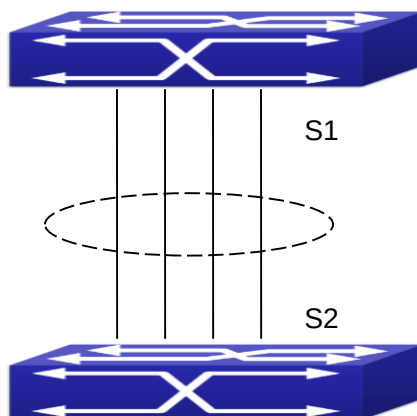


图 2-8 以 LACP 方式配置 Port Channel

如上图所示，交换机 S1 上的 1、2、3、4 端口都是 access 口，将这 4 个端口以 active 方式加入 group 1，交换机 S2 上 6、8、9、10 端口为 access 口，将这 4 个端口以 passive 方式加入 group 2，将以上对应端口分别用网线相连。

配置步骤如下：

```
Switch1#config
```

```
Switch1(config)#interface ethernet 1/1-4
```

```
Switch1(Config-If-Port-Range)#port-group 1 mode active
```

```
Switch1(Config-If-Port-Range)#exit
```

```
Switch1(config)#interface port-channel 1
```

```
Switch1(Config-If-Port-Channel1)#
```

```
Switch2#config
```

```
Switch2(config)#port-group 2
```

```
Switch2(config)#interface ethernet 1/6
```

```
Switch2(Config-If-Ethernet1/6)#port-group 2 mode passive
```

```
Switch2(Config-If-Ethernet1/6)#exit
```

```
Switch2(config)#interface ethernet 1/8-10
```

```
Switch2(Config-If-Port-Range)#port-group 2 mode passive
```

```
Switch2(Config-If-Port-Range)#exit
```

```
Switch2(config)#interface port-channel 2
```

```
Switch2(Config-If-Port-Channel2)#
```

配置结果：

过一段时间后，shell 提示端口汇聚成功，此时交换机 S1 的端口 1、2、3、4 汇聚成一个汇

聚端口，汇聚端口名为 Port-Channel1，交换机 S2 的端口 6、8、9、10 汇聚成一个汇聚端口，汇聚端口名为 Port-Channel2，并且都可以进入汇聚接口配置模式进行配置。

案例 2：以 ON 方式配置 Port Channel。

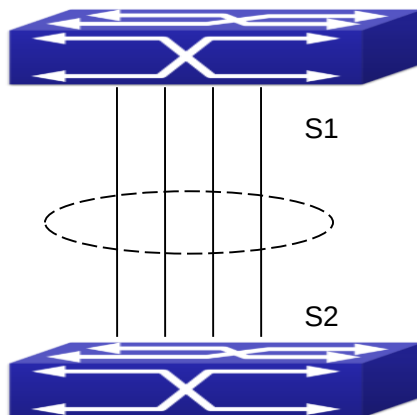


图 2-9 以 ON 方式配置 Port Channel

如图所示，交换机 S1 上的 1、2、3、4 端口都是 access 口，将这 4 个端口以 on 方式加入 group 1，在交换机 S2 上 6、8、9、10 端口为 access 口，将这 4 个端口以 on 方式加入 group 2。

配置步骤如下：

```
Switch1#config
Switch1(config)#interface ethernet 1/1
Switch1(Config-If-Ethernet1/1)#port-group 1 mode on
Switch1(Config-If-Ethernet1/1)#exit
Switch1(config)#interface ethernet 1/2
Switch1 (Config-If-Ethernet1/2)#port-group 1 mode on
Switch1 (Config-If-Ethernet1/2)#exit
Switch1 (config)#interface ethernet 1/3
Switch1 (Config-If-Ethernet1/3)#port-group 1 mode on
Switch1 (Config-If-Ethernet1/3)#exit
Switch1 (config)#interface ethernet 1/4
Switch1 (Config-If-Ethernet1/4)#port-group 1 mode on
Switch1 (Config-If-Ethernet1/4)#exit
```

```
Switch2#config
Switch2 (config)#port-group 2
```

```
Switch2 (config)#interface ethernet 1/6
Switch2 (Config-If-Ethernet1/6)#port-group 2 mode on
Switch2 (Config-If-Ethernet1/6)#exit
Switch2 (config)#interface ethernet 1/8-10
Switch2 (Config-If-Port-Range)#port-group 2 mode on
Switch2 (Config-If-Port-Range)#exit
```

配置结果：

将交换机 S1 上的 1、2、3、4 端口依次加入 port-group1 后我们可以看到，以 on 方式加入一个组完全是强制性的，两端的交换机并不会通过交换 LACP PDU 来完成汇聚，汇聚也是触发式的，当敲入将端口 1/2 加入 port-group1 的命令时，1 和 2 马上汇聚在一起形成 port-channel1，当将端口 1/3 加入 port-group1 时，1 和 2 汇聚成的 port-channel1 被拆散，马上 1、2、3 三个端口又重新汇聚成 port-channel1，当将端口 1/4 加入 port-group1 时，1、2、3 端口汇聚成的 port-channel1 被拆散，马上 1、2、3、4 端口又重新汇聚成 port-channel1（需要说明的是，当有一个新的端口要加入已经汇聚成功的组时，必须先拆散原先的组，然后再能汇聚成一个新的组）。结果是交换机 S1 和交换机 S2 上的三个端口都以 ON 模式汇聚起来，各自形成一个汇聚端口。

2.5.5 Port Channel 排错帮助

当配置端口聚合功能出现问题时，请检查是否是如下原因：

- ☞ 端口聚合组中的端口不具有相同的属性，即双工模式是否为全双工模式，速率是否强制成相同的速率，以及 VLAN 的属性等。如果检查不相同，则修改成相同。
- ☞ 一些命令不能在 port-channel 上的端口使用，包括：arp, bandwidth, ip, ip-forward 等。

2.6 MTU

2.6.1 MTU 介绍

JUMBO（超大帧）在业界现在还没有一个确定的标准（包括帧格式和帧的长度），一般将大小在 1519-9000 之间的帧视为超大帧。超大帧的网络实现可以将整个网络的速度提高 2%到 5%。JUMBO 在使用上只是将交换机在收发的帧的长度增加了。但是考虑到 JUMBO 帧的长度问题，所以 JUMBO 是不送到 CPU 的，我们将送到 CPU 的 JUMBO 帧在收包处理中将 JUMBO 帧丢掉了。

2.6.2 MTU 配置任务序列

1. 配置 MTU 功能使能

1. 配置 MTU 功能使能

命令	解释
全局配置模式	
mtu [<mtu-value>]	启用 mtu 帧的收发功能。
no mtu	该命令的 NO 操作禁用 mtu 帧的收发功能。

2.7 EFM OAM

2.7.1 EFM OAM 介绍

以太网最初为局域网而设计，但随着以太网的发展，应用扩展到了城域网和广域网，链路长度和网络规模上都迅速扩大，而由于有效管理维护机制的缺乏，严重阻碍以太网技术在城域网和广域网的应用。因此，在以太网上实现 OAM 功能成为必然的发展趋势。

关于以太网 OAM，有四个协议标准：802.3ah(EFM OAM)、802.3ag(CFM)、E-LMI、Y.1731；EFM OAM 和 CFM 为 IEEE 组织制定，EFM OAM 工作在数据链路层，可以有效地发现和管理底层存在的数据链路情况，利用 EFM OAM 技术可以有效提高对以太网的管理和维护能力，保障网络的稳定运行。CFM 为网络级以太网 OAM 技术，多应用于网络的接入汇聚层，用于监测整个网络的连通性、定位网络的连通性故障。Y.1731 是由 ITU，即国际电信联盟制定的标准，功能比 CFM 强大许多，也可以说 CFM 实现的功能是 Y.1731 的子集。E-LMI 是由 MEF 制定的标准，只应用于 UNI（用户边界和用户面对的供应商边界之间）。四个协议分别适用于不同的网络拓扑和管理要求，四者之间是互补的关系。

EFM OAM（Ethernet in the First Mile Operation, Administration and Maintenance，最后一英里以太网操作，管理和维护），工作在 OSI 模型的数据链路层，通过 OAM 子层来实现相关的功能，如下图所示：

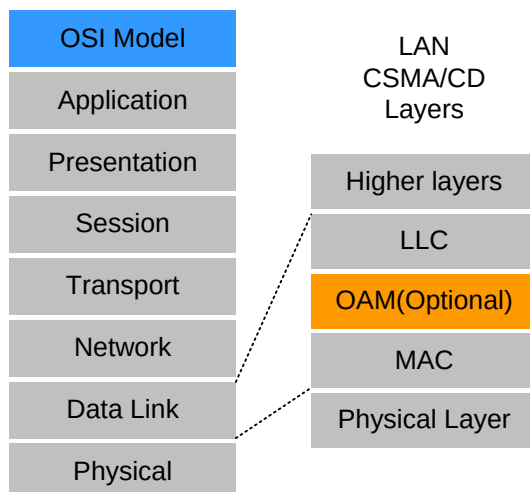


图 2-10 OAM 在 OSI 参考模型中的位置

OAM protocol data units(OAMPDU)使用慢协议的目的 mac 地址 01-80-c2-00-00-02, 最大发送速率为 10Pkt/s。

EFM OAM 功能建立在 OAM 连接建立的基础上, 提供了诸如链路监控, 远端故障指示和远端环回控制等管理链路运行的一种机制, 下面是 EFM OAM 工作流程的简单介绍。

1. 建立 OAM 连接

本端 OAM 实体通过交互 Information OAMPDU 发现远端 OAM 实体、并与之建立稳定会话。EFM OAM 的工作模式有两种: 主动模式、被动模式。EFM OAM 连接只能由主动模式的 OAM 实体发起, 而被动模式的 OAM 实体只能等待对端 OAM 实体的连接请求。EFM OAM 连接建立后, 两端的 OAM 实体通过持续发送 Information OAMPDU 保持连接。若在 5 秒钟内未收到对端 OAM 实体的 Information OAMPDU, 则认为连接超时, OAM 连接中断。

2. 监控链路性能

以太网的故障检测非常困难, 特别是在网络物理通信没有中断而网络性能缓慢下降的情况下。链路性能监控用于在各种环境下检测和发现链路层故障, EFM OAM 通过交互 Event Notification OAMPDU (事件通知报文) 来监控链路: 当一端 OAM 实体监控到一般链路事件时, 将向其对端发送 Event Notification OAMPDU 进行通报, 同时将监控信息记入日志并发送 SNMP Trap 上报网管系统; 而对端 OAM 实体收到事件通知报文后, 同样也将其记入日志并上报。这样, 管理员就可以通过观察日志信息和网管系统动态地掌握网络的状况。

EFM OAM 监控的一般链路事件是指链路上发生的一般性错误事件, 包括错误符号周期事件, 错误帧事件, 错误帧周期事件, 错误帧秒总数事件。

错误符号周期事件: 在指定 M 秒内, 错误符号数大于或等于设定的低阈值。(符号: 指物理介质上的最小数据传输单元。对于编码系统是唯一的, 不同的物理介质上的 symbol 也可能是不同的。符号率即每秒电子状态改变的次数)

错误帧周期事件: 指定帧数 N 为周期, 在收到 N 个帧的周期内错误帧数大于或等于设

定的低阈值。（错误帧：指收到的 CRC 检测出错的帧）

错误帧事件：在指定 M 秒内，收到的错误帧数大于或等于设定的低阈值。

错误帧秒总数事件：在指定 M 秒内，收到的错误帧秒数大于或等于设定的低阈值。（错误帧秒：某一秒内，收到至少 1 个错误帧，这一秒就称为错误帧秒）

3.远端故障指示

由于设备故障或不可用而导致流量中断的情况，OAMPDU 定义了一个标志（即 Flags 域）允许以太网 OAM 实体将故障信息通知给对端。由于在以太网 OAM 连接已建立的情况下，两端的 OAM 实体会不断地交互 Information OAMPDU，因此本端 OAM 实体可以将本端发生的紧急链路事件通过 Information OAMPDU 告诉远端 OAM 实体，同时记录日志并发送 SNMP Trap 告警。这样，管理员可以通过观察日志信息和网管系统动态地了解链路状态，对相应的错误及时进行处理。

Information OAMPDU 将紧急链路事件即链路故障分为三种，分别是 Critical Event, Dying Gasp 和 Link Fault；不同厂商对这三种故障的具体定义都不同，我们的定义为：

Critical Event，端口 EFM OAM 功能关闭；

Link Fault，本地处于单向操作或错误数大于等于设定的高阈值；单向操作（Unidirectional Operation）是指在全双工且不支持自动协商的链路上，链路的一个方向无法工作（一般发生在通过光纤连接的链路上，收或发端因为某些原因无法连通）；EFM OAM 可以检测到该故障，并通过发送 Information OAMPDU 来通知远端 OAM 实体。

Dying Gasp，暂无定义；设备虽然不生成 Dying Gasp OAMPDU，但依然能够接收并处理对端发送的该类型 OAMPDU。

4. 远端环回

远端环回只有在以太网 OAM 连接建立完成后才能实现，且只有工作在 OAM 主动模式的端口才能发起环回请求。在 OAM 连接已经建立的情况下，主动模式的 OAM 实体发起远端环回命令，对端实体对该命令进行响应。当远端处于环回模式下，除了 OAMPDU 报文以外的所有报文都将按照原路返回，网络管理员通过远端环回可以检测链路的延时，抖动和吞吐量。定期地进行环回检测可以及时发现网络故障，并通过分段环回检测来帮助定位故障发生的具体区域。注意，在远端环回模式下，将不能进行正常通信。

EFM OAM 典型应用拓扑如下图所示，它主要用于点到点链路和仿真点到点的 IEEE 802.3 链路，通过在两个点到点连接的设备上启用 EFM OAM 功能，监控以太网接入的“最后一英里”中常见的链路问题。用户端到电信运营商局端设备的这一段连接从使用者的角度看是“最前一英里”，而从运营者的角度则是“最后一英里”。

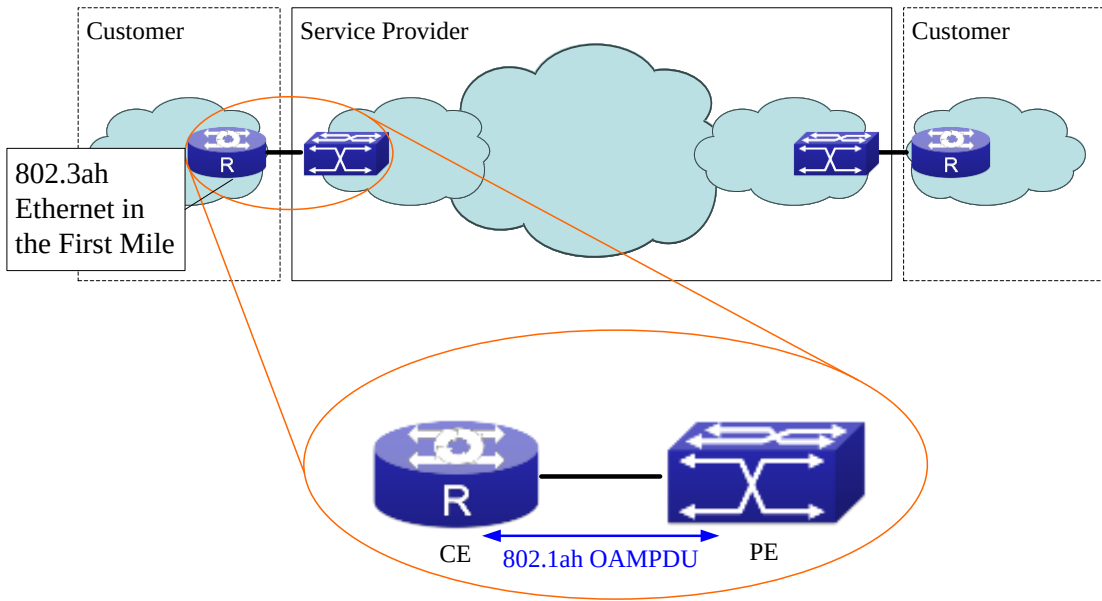


图 2-11 OAM 典型应用拓扑图

2.7.2 EFM OAM 基本配置

EFM OAM 配置任务序列如下：

- 1.使能端口 EFM OAM 功能
- 2.配置 EFM OAM 的链路监控（Link monitor）参数
- 3.配置 EFM OAM 的远端错误指示（Remote Failure）参数
- 4.使能端口 EFM OAM 环回功能

注意：配置 OAM 参数时，需要先启动 OAM 功能。

1. 使能端口 EFM OAM 功能

命令	解释
端口配置模式	
ethernet-oam mode {active passive}	配置 EFM OAM 的工作模式,默认为 active 主动模式。
ethernet-oam no ethernet-oam	使能端口 EFM OAM 功能; 本命令的 no 操作作为关闭端口 EFM OAM 功能。

ethernet-oam period <seconds> no ethernet-oam period	配置 OAMPDU 的发送周期（可选），本命令的 no 操作为恢复 OAMPDU 发送周期为默认值。
ethernet-oam timeout <seconds> no ethernet-oam timeout	配置 EFM OAM 连接超时时间（可选），本命令的 no 操作为恢复连接超时时间为默认值。

2. 配置 EFM OAM 的链路监控（Link monitor）参数

命令	解释
端口配置模式	
ethernet-oam link-monitor no ethernet-oam link-monitor	使能 EFM OAM 的链路监控功能（默认使能），本命令的 no 操作为关闭 EFM OAM 的链路监控功能。
ethernet-oam errored-symbol-period {threshold low <low-symbols> window <seconds>} no ethernet-oam errored-symbol-period {threshold low window }	配置错误符号周期事件的低阈值和窗口周期值；本命令的 no 操作为恢复各个配置为默认值。（可选）
ethernet-oam errored-frame-period {threshold low <low-frames> window <seconds>} no ethernet-oam errored-frame-period {threshold low window }	配置错误帧周期事件的低阈值和窗口周期值；本命令的 no 操作为恢复各个配置为默认值。（可选）
ethernet-oam errored-frame {threshold low <low-frames> window <seconds>} no ethernet-oam errored-frame {threshold low window }	配置错误帧事件的低阈值和窗口周期值；本命令的 no 操作为恢复各个配置为默认值。（可选）

ethernet-oam errored-frame-seconds {threshold low <low-frame-seconds> window <seconds>} no ethernet-oam errored-frame-seconds {threshold low window}	配置错误帧秒总数事件的低阈值和窗口周期值；本命令的 no 操作作为恢复各个配置为默认值。（可选）

3. 配置 EFM OAM 的远端错误指示（Remote Failure）参数

命令	解释
端口配置模式	
ethernet-oam remote-failure no ethernet-oam remote-failure	使能 EFM OAM 的远端故障指示功能(故障指本地发生 critical-event 或 link-fault 事件)，本命令的 no 操作为关闭 EFM OAM 的远端故障指示功能。（可选）
ethernet-oam errored-symbol-period threshold high {high-symbols none} no ethernet-oam errored-symbol-period threshold high	配置错误符号周期事件的高阈值；本命令的 no 操作为恢复配置为默认值。（可选）
ethernet-oam errored-frame-period threshold high {high-frames none} no ethernet-oam errored-frame-period threshold high	配置错误帧周期事件的高阈值；本命令的 no 操作为恢复配置为默认值。（可选）
ethernet-oam errored-frame threshold high {high-frames none} no ethernet-oam errored-frame threshold high	配置错误帧事件的高阈值；本命令的 no 操作为恢复配置为默认值。（可选）
ethernet-oam errored-frame-seconds threshold high {high-frame-seconds none} no ethernet-oam errored-frame-seconds threshold high	配置错误帧秒总数事件的高阈值；本命令的 no 操作为恢复配置为默认值。（可选）

4. 使能端口 EFM OAM 环回功能

命令	解释

端口配置模式	
ethernet-oam remote-loopback no ethernet-oam remote-loopback	使能远端 EFM OAM 实体进入 OAM 环回模式（对端需要配置 OAM 环回支持），本命令的 no 操作为取消远端的 OAM 环回。
ethernet-oam remote-loopback supported no ethernet-oam remote-loopback supported	使能端口的远端环回支持，本命令的 no 操作为取消端口的远端环回支持。

2.7.3 EFM OAM 举例

案例：

在点到点链路的 CE 和 PE 设备上开启 EFM OAM 功能，监控该“最后一英里”链路的性能，当出现错误事件时，会记录日志并上报网管系统；在必要时，可以使用远端环回功能检测链路。

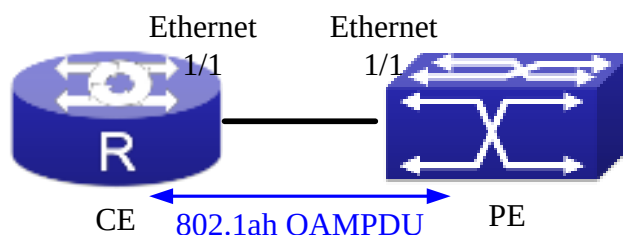


图 2-12 OAM 典型应用拓扑图

配置步骤：（下面省略了 SNMP 及 Log 的配置过程，具体请参考手册的相关内容）

在 CE 设备上的配置

```
CE(config)#interface ethernet 1/1
```

```
CE (config-if-ethernet1/1)#ethernet-oam mode passive
```

```
CE (config-if-ethernet1/1)#ethernet-oam
```

```
CE (config-if-ethernet1/1)#ethernet-oam remote-loopback supported
```

其他参数使用默认配置

在 PE 设备上的配置

```
PE(config)#interface ethernet 1/1
```

```
PE (config-if-ethernet1/1)#ethernet-oam
```

其他参数使用默认配置

当需要使用远端环回时，执行以下命令

```
PE (config-if-ethernet1/1)#ethernet-oam remote-loopback
```

检测完毕后，执行以下命令，使对端退出 OAM 环回

```
PE (config-if-ethernet1/1)#no ethernet-oam remote-loopback
```

当不需要支持远端环回时，执行以下命令

```
CE (config-if-ethernet1/1)#no ethernet-oam remote-loopback supported
```

2.7.4 EFM OAM 排错帮助

当在使用 EFM OAM 过程中遇到问题，请检查是否是如下原因：

- ☞ 检查链路两端的 OAM 实体是否都处于被动模式，都处于被动模式下的两个 OAM 实体之间无法建立 EFM OAM 连接。
- ☞ 请确保 SNMP 配置的正确性，否则错误事件无法上报网管系统。
- ☞ 处于 OAM 环回模式的链路，不能进行正常通信；在检测完链路的性能后，应该及时取消远端环回。
- ☞ 只有部分板卡支持远端环回功能，请确保所使用的板卡支持该功能。
- ☞ OAM 远端环回功能与 STP，MRPP，ULPP，Flow Control，loopback detection 功能互斥，因此开启了 OAM 环回功能的端口不能配置上述功能。

2.8 DDM

2.8.1 DDM 介绍

2.8.1.1 DDM 简介

DDM 即数字诊断监测（Digital Diagnostic Monitor），在 SFF-8472 MSA 中，规范了数字诊断功能的详细内容。该规范规定，在模块内部的电路板上侦测和数字化参数信号；然后，提供经过标定的结果或提供数字化的测量结果及标定参量，这些信息被存贮在标准的内存结构中，以便通过双缆串行接口读取。

在通常我们所说的智能光模块上均支持数字诊断功能。通过光模块内部的 MCU，网络管理单元可以实时的监测光模块的温度（temperature）、工作电压（voltage）、激光偏置电流（bias current）以及发射光功率（tx power）和接收光功率（rx power），获取光模块各参数的阈值，并能得到当前光模块的实时状态。通过这些手段，可以帮助网络管理单元找出光纤链路中发生故障的位置，简化维护工作，提高系统的可靠性。

光接口交换机的数字诊断功能（DDM）是运营商客户的基本需求，可做以下应用：

1、模块寿命预测

通过监测激光的偏置电流来预测激光器的寿命,可以粗略的估计激光器的使用寿命是否接近终了。通过对收发模块内部的工作电压和温度进行实时监测,可以让系统管理员发现一些潜在的问题:

- (1) Vcc 电压过高,会带来 CMOS 器件的击穿, Vcc 电压过低,激光器不能正常工作;
- (2) 接收功率太高,会损坏接收模块;接收功率过低,接收模块断路不能正常工作;
- (3) 工作温度太高,会加速器件的老化;
- (4) 通过对接收到的光功率的监测,可以对线路和远端发射机的性能进行监控。

2、故障定位

在光链路中,定位故障的发生位置对业务的快速加载至关重要。故障隔离特性则可以使系统管理员快速定位链路故障的位置。此特性可以定位故障是在模块内还是在线路上,是在本地模块还是在远端模块。通过快速定位故障,减少了系统的故障修复时间。

通过数字诊断功能,对温度 (temperature)、工作电压 (voltage)、激光偏置电流 (bias current) 以及发射光功率 (tx power) 和接收光功率 (rx power) 的警告 (warning) 和警报 (alarm) 状态进行综合分析,可以快速的定位故障。状态变量 Tx Fault (发射失败) 和 Rx LOS (接收信号丢失) 等状态都对故障的分析起着重要的作用。

3、兼容性验证

兼容性验证就是分析模块的工作环境是否符合数据手册或和相关的标准兼容。模块的性能只有在这种兼容的工作环境下才能得到保证。在有些情况下,由于环境参数超出数据手册或相关的标准,将造成模块性能下降,从而出现传输误码。工作环境与模块不兼容的情况有:

- (1) 电压超出规定范围;
- (2) 接收光功率过载或低于接收机灵敏度;
- (3) 温度超出工作温度范围。

2.8.1.2 DDM 功能

DDM 功能描述具体如下:

1. 显示收发器的监测信息

收发器的实时参数有发射功率 (TX power)、接收功率 (RX power)、温度 (Temperature)、电压 (Voltage)、偏置电流 (Bias current),通过对这些参数的检测以及对警告 (warning)、报警 (alarm)、实时状态、阈值等相关监测信息的查询,网络管理员可以了解到当前收发器的工作状态,发现一些潜在的问题。通过查看光模块的故障信息可以帮助网络管理员快速定位线路故障,加快修复故障的时间。

2. 用户自定义阈值

收发器对于实时参数 TX power、RX power、Temperature、Voltage、Bias current 有固定的阈值,但由于用户的使用环境不一样,需要自定义阈值来监测收发器的工作状态,因此,通过自定义阈值,可以让用户灵活的监测收发器的工作状态,及时的发现故障。阈值类型包括报警的高阈值 (high alarm)、报警的低阈值 (low alarm)、警告的高阈值 (high warn)、警告的低阈值 (low warn)。

用户配置的阈值和厂商配置的阈值可以同时显示出来，供用户查看。当用户定义的阈值不合理时，会提示用户，并自动按厂商的阈值进行报警或警告。用户可以恢复所有阈值为厂商阈值，可以恢复单个阈值为厂商阈值。

阈值的合理性：报警的高低阈值应该包含警报的高低阈值，并且高阈值应该大于低阈值，即 $\text{high alarm} \geq \text{high warn} \geq \text{low warn} \geq \text{low alarm}$ 。

对于 sfp 光模块，接收功率的校验方式分内部校验和外部校验，这个是厂商决定的。参数的实时监测值和厂商预设的阈值采用相同的校验方式。

3. 软件监控

用户除了实时的查看收发器模块的工作状态，还需要监控之前收发器发生异常时的时间、异常类型等详细情况。软件监控这个功能可以让用户通过查看日志发现之前收发器产生的异常情况，也可以让用户通过命令查询上一次的发生异常的详细情况。当用户发现光模块有异常信息，对其进行处理后，需要重新监控光模块的信息时，可以清楚软件监控的异常信息，重新开始监控。

2.8.2 DDM 配置任务序列

DDM 配置任务序列如下：

1. 显示收发器（transceiver）实时检测信息
2. 配置收发器（transceiver）各监测参数的报警或警告阈值
3. 配置收发器（transceiver）的软件监控状态
 - （1）配置收发器（transceiver）的软件监控的时间间隔
 - （2）配置收发器（transceiver）的软件监控的使能状态
 - （3）显示收发器（transceiver）的软件监控信息
 - （4）清除收发器（transceiver）的软件监控信息

1. 显示收发器（transceiver）实时检测信息

命令	解释
用户模式、特权模式、全局配置模式	
show transceiver [interface ethernet <interface-list>] [detail]	显示收发器的监测信息，

2. 配置收发器（transceiver）各监测参数的报警或警告阈值

命令	解释
端口配置模式	

transceiver threshold {default {temperature voltage bias rx-power tx-power} {high-alarm low-alarm high-warn low-warn} {<value> default}}	用户配置自定义的阈值。

3. 配置收发器（transceiver）的软件监控状态

(1) 配置收发器（transceiver）的软件监控的时间间隔

命令	解释
全局配置模式	
transceiver-monitoring interval <minutes> no transceiver-monitoring interval	设置软件监控的时间间隔。本命令的 no 命令为设置时间间隔为默认时间 间隔 15 分钟。

(2) 配置收发器（transceiver）的软件监控的使能状态

命令	解释
端口配置模式	
transceiver-monitoring {enable disable}	设定端口是否开启软件监控功能。只 有开启软件监控任务的端口，系统才 会记录其异常状态。当端口关闭了软 件监控任务后，此端口的异常信息将 被清空。

(3) 显示收发器（transceiver）的软件监控信息

命令	解释
特权模式、全局配置模式	
show transceiver threshold-violation [interface ethernet <interface-list>]	显示软件监控信息，包括上一次超过 阈值的信息、当前软件监控的时间间 隔和该端口是否使能了软件监控。

(4) 清除收发器（transceiver）的软件监控信息

命令	解释
特权模式	
clear transceiver threshold-violation [interface ethernet <interface-list>]	清空软件监控的阈值异常信息。

2.8.3 DDM 举例

举例 1:

在交换机上的 21 口和 23 口插上支持 DDM 的 SFP 光模块, 在 24 口上插上不支持 DDM 的 SFP 光模块, 22 口不插任何 SFP 光模块, 显示光模块的 DDM 信息。

a、显示所有正常读取出实时参数的端口信息 (不支持或者没有插光模块, 则不显示), 例如:

```
Switch#show transceiver
```

Interface	Temp (°C)	Voltage (V)	Bias (mA)	RX Power (dBm)	TX Power (dBm)
1/21	33	3.31	6.11	-30.54(A-)	-6.01
1/23	33	5.00 (W+)	6.11	-20.54(W-)	-6.02

b、显示指定端口的信息(不支持或没插光模块的显示 N/A), 例如:

```
Switch#show transceiver interface ethernet 1/21-22;23
```

Interface	Temp (°C)	Voltage (V)	Bias (mA)	RX Power (dBm)	TX Power (dBm)
1/21	33	3.31	6.11	-30.54(A-)	-6.01
1/22	N/A	N/A	N/A	N/A	N/A
1/23	33	5.00 (W+)	6.11	-20.54(W-)	-6.02

c、显示详细的信息, 包括端口的基本信息、实时监测参数值、警告(warning)、报警(alarm)、异常状态、阈值信息、序列号, 例如:

```
Switch#show transceiver interface ethernet 1/21-22;24 detail
```

```
Ethernet 1/21 transceiver detail information:
```

```
Base information:
```

```
SFP found in this port, manufactured by company, on Sep 29 2010.
```

```
Type is 1000BASE-SX. Serial Number is 1108000001.
```

```
Link length is 550 m for 50um Multi-Mode Fiber.
```

```
Link length is 270 m for 62.5um Multi-Mode Fiber.
```

```
Nominal bit rate is 1300 Mb/s, Laser wavelength is 850 nm.
```

```
Brief alarm information:
```

```
RX loss of signal
```

```
Voltage high
```

```
RX power low
```

```
Detail diagnostic and threshold information:
```

	Diagnostic		Threshold		
	Realtime Value	High Alarm	Low Alarm	High Warn	Low Warn
Temperature (°C)	33	70	0	70	0

Voltage (V)	7.31(A+)	5.00	0.00	5.00	0.00
Bias current (mA)	6.11(W+)	10.30	0.00	5.00	0.00
RX Power (dBm)	-30.54(A-)	9.00	-25.00	9.00	-25.00
TX Power (dBm)	-6.01	9.00	-25.00	9.00	-25.00

Ethernet 1/22 transceiver detail information: N/A

Ethernet 1/24 transceiver detail information:

Base information:

SFP found in this port, manufactured by company, on Sep 29 2010.

Type is 1000BASE-SX. Serial Number is 1108000001.

Link length is 550 m for 50um Multi-Mode Fiber.

Link length is 270 m for 62.5um Multi-Mode Fiber.

Nominal bit rate is 1300 Mb/s, Laser wavelength is 850 nm.

Brief alarm information: N/A

Detail diagnostic and threshold information: N/A

说明：如果序列号全为 0，表示序列号没有指定，则显示为：

SFP found in this port, manufactured by company, on Sep 29 2010.

Type is 1000BASE-SX. Serial Number is not specified.

Link length is 550 m for 50um Multi-Mode Fiber.

Link length is 270 m for 62.5um Multi-Mode Fiber.

Nominal bit rate is 1300 Mb/s, Laser wavelength is 850 nm.

举例 2:

在交换机上的 21 口插上支持 DDM 的 SFP 光模块，显示光模块的 DDM 信息后配置光模块的阈值。

步骤 1：显示光模块的 DDM 详细信息

Switch#show transceiver interface ethernet 1/21 detail

Ethernet 1/21 transceiver detail information:

Base information:

.....

Brief alarm information:

RX loss of signal

Voltage high

RX power low

Detail diagnostic and threshold information:

Diagnostic

Threshold

	Realtime Value	High Alarm	Low Alarm	High Warn	Low Warn
	-----	-----	-----	-----	-----
Temperature (°C)	33	70	0	70	0
Voltage (V)	7.31(A+)	5.00	0.00	5.00	0.00
Bias current (mA)	6.11(W+)	10.30	0.00	5.00	0.00
RX Power (dBm)	-30.54(A-)	9.00	-25.00	9.00	-25.00
TX Power (dBm)	-13.01	9.00	-25.00	9.00	-25.00

步骤 2: 配置光模块的发送功率阈值, 警告下限阈值为-12, 报警下限阈值为-10.00

Switch#config

Switch(config)#interface ethernet 1/21

Switch(config-if-ethernet1/21)#transceiver threshold tx-power low-warning -12

Switch(config-if-ethernet1/21)#transceiver threshold tx-power low-alarm -10.00

步骤 3: 显示光模块的 DDM 详细信息。报警采用用户配置的阈值, 厂商设置的阈值用括号标注起来, 由于-13.01<-12.00, 所以发送功率有 (A-) 的报警。

Switch#show transceiver interface ethernet 1/21 detail

Ethernet 1/21 transceiver detail information:

Base information:

.....

Brief alarm information:

RX loss of signal

Voltage high

RX power low

TX power low

Detail diagnostic and threshold information:

	Diagnostic			Threshold	
	Realtime Value	High Alarm	Low Alarm	High Warn	Low Warn
	-----	-----	-----	-----	-----
Temperature (°C)	33	70	0	70	0
Voltage (V)	7.31(A+)	5.00	0.00	5.00	0.00
Bias current (mA)	6.11(W+)	10.30	0.00	5.00	0.00
RX Power (dBm)	-30.54(A-)	9.00	-25.00	9.00	-25.00
TX Power (dBm)	-13.01(A-)	9.00	-12.00(-25.00)	9.00	-10.00(-25.00)

举例 3:

在交换机上的 21 口插上支持 DDM 的 SFP 光模块, 显示光模块的软件监控信息后使能端口的软件监控功能。

步骤 1: 显示光模块的软件监控信息。21 口和 22 口都没有使能软件监控功能，软件监控的时间间隔是 30 分钟。

```
Switch(config)#show transceiver threshold-violation interface ethernet 1/21-22
```

Ethernet 1/21 transceiver threshold-violation information:

Transceiver monitor is disabled. Monitor interval is set to 30 minutes.

The last threshold-violation doesn't exist.

Ethernet 1/22 transceiver threshold-violation information:

Transceiver monitor is disabled. Monitor interval is set to 30 minutes.

The last threshold-violation doesn't exist.

步骤 2: 使能 21 口的软件监控功能。

```
Switch(config)#interface ethernet 1/21
```

```
Switch(config-if-ethernet1/21)#transceiver-monitoring enable
```

步骤 3: 显示光模块的软件监控信息。从显示可以看到，21 口使能了软件监控功能，上一次超过阈值出现报警的时间是 2011 年 1 月 2 日的 11 点零 50 秒，超过阈值时详细的 DDM 信息也显示了出来。

```
Switch(config-if-ethernet1/21)#quit
```

```
Switch(config)#show transceiver threshold-violation interface ethernet 1/21-22
```

Ethernet 1/21 transceiver threshold-violation information:

Transceiver monitor is enabled. Monitor interval is set to 30 minutes.

The current time is Jan 02 12:30:50 2011.

The last threshold-violation time is Jan 02 11:00:50 2011.

Brief alarm information:

RX loss of signal

RX power low

Detail diagnostic and threshold information:

	Diagnostic			Threshold	
	Realtime Value	High Alarm	Low Alarm	High Warn	Low Warn
-----	-----	-----	-----		
Temperature (°C)	33	70	0	70	0
Voltage (V)	7.31	10.00	0.00	5.00	0.00
Bias current (mA)	3.11	10.30	0.00	5.00	0.00
RX Power (dBm)	-30.54(A-)	9.00	-25.00(-34)	9.00	-25.00
TX Power (dBm)	-1.01	9.00	-12.05	9.00	-10.00

Ethernet 1/22 transceiver threshold-violation information:

Transceiver monitor is disabled. Monitor interval is set to 30 minutes.

The last threshold-violation doesn't exist.

2.8.4 DDM 排错帮助

当在使用 DDM 功能的过程中遇到问题，请检查是否是如下原因：

- ☞ 请确保端口上的光模块收发器已经插紧，否则无法显示 DDM 信息（显示为 N/A 或没有 DDM 信息）；
- ☞ 请确保 SNMP 配置的正确性，否则报警事件无法上报网管系统；
- ☞ 只有部分板卡和盒式交换机支持 SFP 的 DDM 功能或 XFP 的 DDM 功能，请确保所使用的板卡和盒式交换机支持相应的功能；
- ☞ 使用 `show transceiver` 命令或 `show transceiver detail` 命令时，交换机要检查所有端口，所以显示的时间会比较慢，建议查询指定端口上收发器的监测信息。
- ☞ 请确保用户自定义的阈值的正确性，当任意一个阈值设置错误时，收发器将自动按照厂商的设置的方式进行报警。
- ☞ QSFP+端口 DDM 信息暂不支持发射功率（TX power），这是由 SFF-8436 决定的。
- ☞ QSFP+端口如果配置为 40G 模式，会显示全部 4 个通道的 DDM 信息；如果配置为 4x10G 模式，则只显示当前 10G 端口对应的通道信息。
- ☞ DAC 线缆自带光模块，如果是 passive 模式的线缆，没有 DDM 信息；如果是 active 模式的线缆，可以查看 DDM 信息。

2.9 LLDP-MED

2.9.1 LLDP-MED 介绍

LLDP-MED（Link Layer Discovery Protocol-Media Endpoint Discovery）以 IEEE 的 802.1AB LLDP（Link Layer Discovery Protocol，链路层发现协议）为基础。LLDP 提供了一种标准的链路层发现方式，可以将本端设备的主要能力、管理地址、设备标识、接口标识等信息组织成不同的 TLV（Type/Length/Value，类型/长度/值），并封装在 LLDPDU（Link Layer Discovery Protocol Data Unit，链路层发现协议数据单元）中发布给与自己直连的邻居，邻居收到这些信息后将其以标准 MIB（Management Information Base，管理信息库）的形式保存起来，以供网络管理系统查询及判断链路的通信状况。

802.1AB LLDP 中没有涉及到对语音设备信息的传输和管理。为了方便对网络中语音设备的部署和管理，LLDP-MED 对 LLDP 进行扩展，定义一些新的 TLV。新的 TLV 消息将提供 PoE（Power over Ethernet）、联网策略（Network Policy）以及紧急电话服务的介质端点位置等信息。

2.9.2 LLDP-MED 配置任务序列

1. LLDP-MED 基本功能配置

命令	解释
端口配置模式	
lldp transmit med tlv all no lldp transmit med tlv all	配置指定端口发送所有 LLDP-MED TLV 的功能。no 命令为关闭端口发送 LLDP-MED TLV 的功能。
lldp transmit med tlv capability no lldp transmit med tlv capability	配置指定端口发送 LLDP-MED Capability TLV。no 命令为关闭端口发送 LLDP-MED Capability TLV 的能力。
lldp transmit med tlv networkPolicy no lldp transmit med tlv networkPolicy	配置指定端口发送 LLDP-MED Network Policy TLV。no 命令为关闭端口发送 LLDP-MED Network Policy TLV 的能力。
lldp transmit med tlv extendPoe no lldp transmit med tlv extendPoe	配置指定端口发送 LLDP-MED Extended Power-Via-MDI TLV。no 命令为关闭端口发送 LLDP-MED Extended Power-Via-MDI TLV 的能力。
lldp transmit med tlv location no lldp transmit med tlv location	配置指定端口发送 LLDP-MED Location Identification TLV。no 命令为关闭端口发送 LLDP-MED Location Identification TLV 的能力。
lldp transmit med tlv inventory no lldp transmit med tlv inventory	配置端口发送 LLDP 报文时携带 LLDP-MED Inventory Management TLVs 集合。no 命令为关闭端口发送 LLDP-MED Inventory Management TLVs 的能力。

network policy {voice voice-signaling guest-voice guest-voice-signaling softphone-voice video-conferencing streaming-video video-signaling} [status {enable disable}] [tag {tagged untagged}] [vid {<vlan-id> dot1p}] [cos <cos-value>] [dscp <dscp-value>] no network policy {voice voice-signaling guest-voice guest-voice-signaling softphone-voice video-conferencing streaming- video video-signaling}	配置端口的网络策略，包括端口的 VLAN ID、支持的应用(如语音和视频)、应用优先级以及使用策略等。
civic location {dhcp server switch endpointDev} <country-code> no civic location	配置 Civic Address LCI 格式地址的设备类型和国家编号，同时进入端口的 Civic Address LCI 格式地址配置模式，no 命令操作为取消端口 Civic Address LCI 格式地址的所有配置。
ecs location <tel-number> no ecs location	在端口上配置 ECS ELIN 格式的地址，no 命令操作取消在端口上已经配置的 ECS ELIN 格式的地址。
lldp med trap {enable disable}	配置指定端口使能或者禁止 LLDP-MED 网络拓扑发生变化发送 TRAP 消息功能。
Civic Address LCI 格式地址配置模式	
{description-language province-state city county street locationNum location floor room postal otherInfo} <address> no {description-language province-state city county street locationNum location floor room postal otherInfo}	进入端口的 Civic Address LCI 格式地址配置模式后，配置详细的地址信息。
全局配置模式	
lldp med fast count <value> no lldp med fast count	LLDP-MED 快速启动机制激活的情况下，需要快速发送含有 LLDP-MED TLV 的 LLDP 报文，该命令对快速发送报文

	的个数进行设置。no 命令恢复快速发送报文的个数为默认值。
特权配置模式	
show lldp	显示全局 LLDP 和 LLDP-MED 的配置信息。
show lldp [interface ethernet <IFNAME>]	显示当前端口的 LLDP 和 LLDP-MED 配置信息。
show lldp neighbors [interface ethernet <IFNAME>]	显示端口邻居的 LLDP 和 LLDP-MED 信息。
show lldp traffic	显示端口 LLDP 和 LLDP-MED 收发包统计。

2.9.3 LLDP-MED 举例

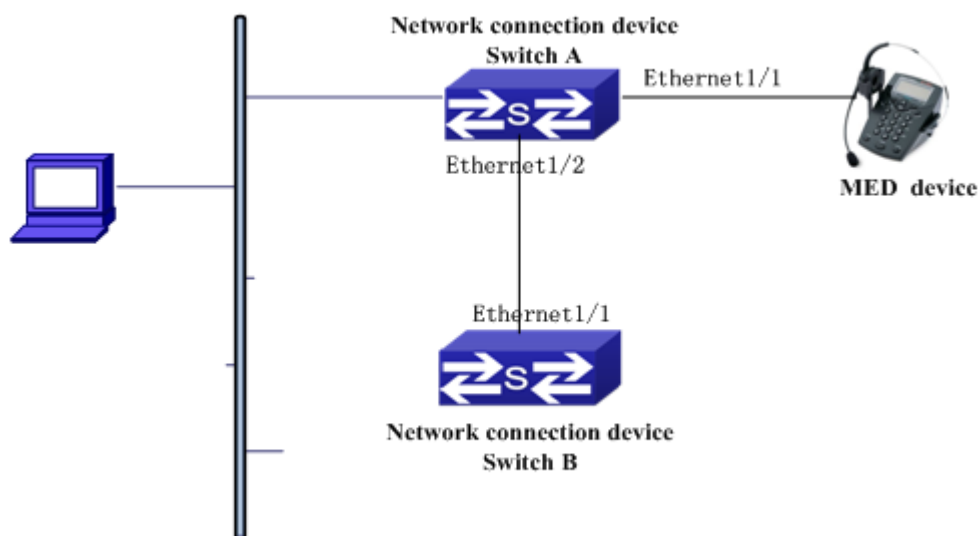


图 2-13 LLDP-MED 基本配置功能组网图

1) 配置 Switch A

```
SwitchA(config)#interface ethernet1/1
```

```
SwitchA (Config-If-Ethernet1/1)# lldp enable
```

```
SwitchA (Config-If-Ethernet1/1)# lldp mode both （本配置可省略，默认工作模式为 RxTx）
```

```
SwitchA (Config-If-Ethernet1/1)# lldp transmit med tlv capability
```

```
SwitchA (Config-If-Ethernet1/1)# lldp transmit med tlv network policy
```

```
SwitchA (Config-If-Ethernet1/1)# lldp transmit med tlv inventory
```

```
SwitchB (Config-If-Ethernet1/1)# network policy voice tag tagged vid 10 cos 5 dscp 15
```

```
SwitchA (Config-If-Ethernet1/1)# exit
```

```
SwitchA (config)#interface ethernet1/2
```

```
SwitchA (Config-If-Ethernet1/2)# lldp enable
```

```
SwitchA (Config-If-Ethernet1/2)# lldp mode both
```

2) 配置 Switch B

```
SwitchB (config)#interface ethernet1/1
```

```
SwitchB(Config-If-Ethernet1/1)# lldp enable
```

```
SwitchB (Config-If-Ethernet1/1)# lldp mode both
```

```
SwitchB (Config-If-Ethernet1/1)# lldp transmit med tlv capability
```

```
SwitchB (Config-If-Ethernet1/1)# lldp transmit med tlv network policy
```

```
SwitchB (Config-If-Ethernet1/1)# lldp transmit med tlv inventory
```

```
SwitchB (Config-If-Ethernet1/1)# network policy voice tag tagged vid 10 cos 4
```

3) 验证配置结果

在 Switch A 上显示全局和接口的状态信息。

```
SwitchA# show lldp neighbors interface ethernet 1/1
```

```
Port name : Ethernet1/1
```

```
Port Remote Counter : 1
```

```
TimeMark :20
```

```
ChassisIdSubtype :4
```

```
ChassisId :00-03-0f-00-00-02
```

```
PortIdSubtype :Local
```

```
PortId :1
```

```
PortDesc :****
```

```
SysName :****
```

```
SysDesc :*****
```

```
SysCapSupported :4
```

```
SysCapEnabled :4
```

```
LLDP MED Information :
```

```
MED Codes:
```

```
(CAP)Capabilities, (NP) Network Policy
```

```
(LI) Location Identification, (PSE)Power Source Entity
```

```
(PD) Power Device, (IN) Inventory
```

```
MED Capabilities:CAP,NP,PD,IN
```

```
MED Device Type: Endpoint Class III
```

```
Media Policy Type :Voice
```

```
Media Policy :Tagged
```

```
Media Policy Vlan id :10
```

Media Policy Priority :3
Media Policy Dscp :5
Power Type : PD
Power Source :Primary power source
Power Priority :low
Power Value :15.4 (Watts)
Hardware Revision:
Firmware Revision:4.0.1
Software Revision:6.2.30.0
Serial Number:
Manufacturer Name:****
Model Name:Unknown
Assert ID:Unknown
IEEE 802.3 Information :
 auto-negotiation support: Supported
 auto-negotiation support: Not Enabled
 PMD auto-negotiation advertised capability: 1
 operational MAU type: 1
SwitchA# show lldp neighbors interface ethernet 1/2
Port name : interface ethernet 1/2
Port Remote Counter: 1
Neighbor Index: 1
Port name : Ethernet1/2
Port Remote Counter : 1
TimeMark :20
ChassisIdSubtype :4
ChassisId :00-03-0f-00-00-02
PortIdSubtype :Local
PortId :1
PortDesc :Ethernet1/1
SysName :****
SysDesc :*****

SysCapSupported :4
SysCapEnabled :4

说明:

1) switch A 的端口 2 和 switch B 的端口 1, 由于都是网络连接设备的端口, 不会主动发送

含有 MED TLV 的 LLDP 报文，所以尽管在 switch B 的端口 1 上配置发送 MED TLV，该端口也不会发送 MED 相关信息，导致该端口在 switch A 端口 2 上对应的 Remote 表项中没有 MED 相关信息。

- 2) LLDP-MED 端设备可以主动发送含有 MED TLV 的 LLDP 报文，所以可以看到 MED 端设备在 switch A 端口 1 上对应的 Remote 表项含有 LLDP MED 信息。

2.9.4 LLDP-MED 排错帮助

当配置 LLDP-MED 功能出现问题时，请检查是否是如下原因：

- ☞ LLDP全局功能是否开启。
- ☞ 网络连接设备不会主动发送LLDP-MED TLV，只有在收到邻居MED设备发送的含有LLDP-MED TLV的LLDP报文后，才会发送LLDP-MED TLV。如果在网络连接设备上配置了发送LLDP-MED TLV的命令，端口在发送的报文中仍然不携带LLDP-MED TLV，就说明端口还未收到MED信息，端口发送LLDP-MED信息的功能还未使能。
- ☞ 如果邻居设备已经向网络连接设备发送了LLDP-MED信息，但是通过show lldp neighbors命令查看，没有显示LLDP-MED信息，说明邻居发送的LLDP-MED信息有误。

2.10 bpdu-tunnel

2.10.1 bpdu-tunnel 介绍

BPDU Tunnel 是一种二层隧道技术，它使不同地域私网用户的二层协议报文，通过修改源 MAC 地址的方式在运营商网络内的指定通道进行透明传输。

2.10.1.1 bpdu-tunnel 的作用

在城域网应用中，一个企业的多个分支机构可能会通过运营商网络进行互联。运营商通过提供 VPN 功能来使这些位于不同地理位置的网络组成一个“本地”局域网。这需要运营商网络提供“隧道”功能，即用户网络内产生的数据信息可以完整的通过运营商网络到达同一企业的其它网络。为了维持一个“本地”概念，不仅需要对于用户网络内的数据进行隧道传输，而且对于维护用户网络内的某些二层协议报文也要进行隧道传输。

2.10.1.2 bpdu-tunnel 的产生背景

在实际组网中，用户经常利用运营商提供的专线来构建自己的二层网络，这使同一用户的私网通常都分布在运营商公网的两侧。如下图所示，用户 A 拥有属于相同 VLAN 的两台设备 CE 1 和 CE 2，该用户的网络分为网络 1 和网络 2，二者通过运营商网络相连接。当二层协议报文在运营商网络中无法透传时，该用户的网络将无法独立进行二层协议的计算

（如 STP 协议的生成树计算），从而与运营商网络的二层协议计算相互影响。

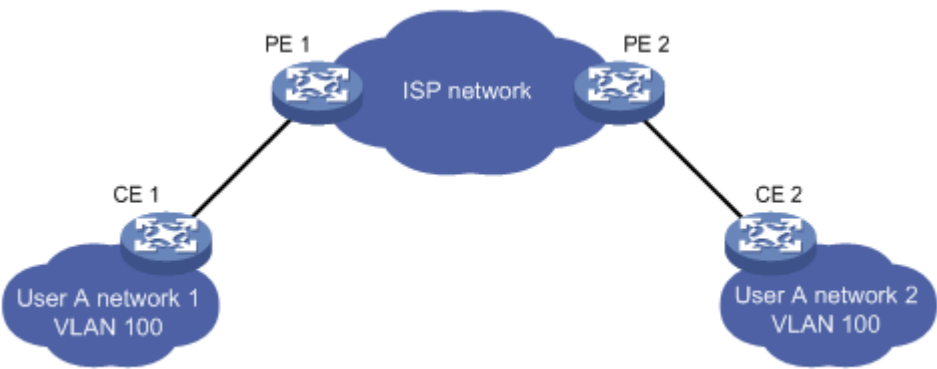


图 2-14 BPDUs Tunnel 应用

2.10.2 bpdu-tunnel 配置任务序列

bpdu-tunnel 配置任务序列如下：

- 1. 配置全局隧道MAC地址
- 2. 配置端口支持隧道功能

1. 配置全局隧道 MAC 地址

命令	解释
全局配置模式	
<pre>bpdu-tunnel-protocol user-defined-protocol <strings> protocol-mac <mac-address> { default-group-mac encap-type group-mac } no bpdu-tunnel-protocol user-defined-protocol <strings> bpdu-tunnel <dot1x gvrp stp > <default-group-mac group-mac> <mac-address> no bpdu-tunnel <dot1x gvrp stp ></pre>	配置或取消全局隧道 MAC 地址。

2. 配置端口支持隧道功能

命令	解释
端口配置模式	

bpdu-tunnel {stp gvrp dot1x user-defined-protocol} no bpdu-tunnel {stp gvrp dot1x user-defined-protocol}	开启或关闭端口上支持隧道功能

2.10.3 bpdu-tunnel 举例

典型应用场景如下图所示，在实际组网中，用户经常利用运营商提供的专线来构建自己的二层网络，这使同一用户的私网通常都分布在运营商公网的两侧。如图所示，用户 A 拥有属于相同 VLAN 的两台设备 CE 1 和 CE 2，该用户的网络分为网络 1 和网络 2，二者通过运营商网络相连接。当二层协议报文在运营商网络中无法透传时，该用户的网络将无法独立进行二层协议的计算（如 STP 协议的生成树计算），从而与运营商网络的二层协议计算相互影响。

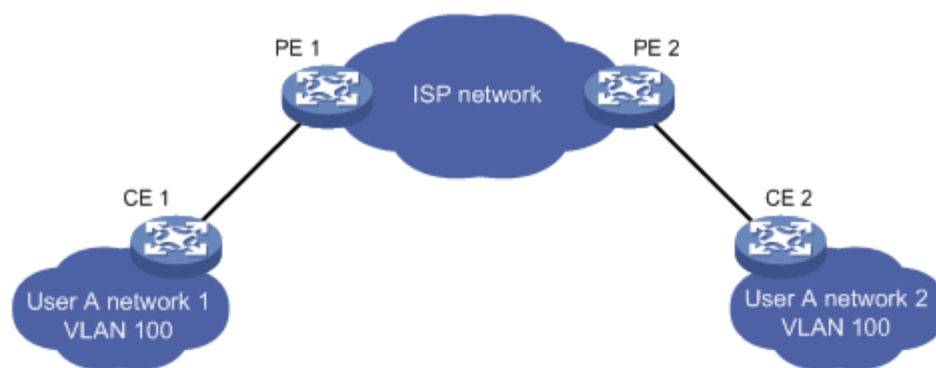


图 2-15 BPDU Tunnel 应用环境

利用 BPDU Tunnel 功能，可以在运营商网络中透传用户网络的二层协议报文：

1. 运营商网络一端的 PE 1 对从用户 A 的网络 1 收到的二层协议报文进行封装，将其目的 MAC 地址替换成一个特定的组播 MAC 地址，然后在运营商网络中进行转发；
2. 封装好的二层协议报文（称为 BPDU Tunnel 报文）被转发至运营商网络另一端的 PE 2，解封封装后被还原为原始的目的 MAC 地址，并发送给用户 A 的网络 2。

边缘交换机 PE1 和 PE2 上关于 bpdu-tunnel 的配置如下：

PE1 的配置：

```
PE1(config)# bpdu-tunnel-protocol dot1x group-mac 01-02-03-04-05-06
PE1(config-if-ethernet1/1)# bpdu-tunnel stp
PE1(config-if-ethernet1/1)# bpdu-tunnel gvrp
PE1(config-if-ethernet1/1)# bpdu-tunnel dot1x
```

PE2 的配置：

```
PE2(config)# bpdu-tunnel-protocol dot1x group-mac 01-02-03-04-05-06
PE2(config-if-ethernet1/1)# bpdu-tunnel stp
```

```
PE2(config-if-ethernet1/1)# bpdu-tunnel gvrp
PE2(config-if-ethernet1/1)# bpdu-tunnel dot1x
```

2.10.4 bpdu-tunnel 排错帮助

端口开启 stp, gvrp 和 dot1x 功能不能直接在该端口上配置 bpdu-tunnel 功能，需要在该端口上先关闭上述功能后才能配置 bpdu-tunnel 功能。

2.11 PORT SECURITY

2.11.1 PORT SECURITY 介绍

Port security 是一种基于 MAC 地址对网络接入进行控制的安全机制，是对已有的 802.1X 认证和 MAC 地址认证的扩充。这种机制通过检测端口收到的数据帧中的源 MAC 地址来控制非授权设备对网络的访问，通过检测从端口发出的数据帧中的目的 MAC 地址来控制对非授权设备的访问。Port security 的主要功能是通过定义各种端口安全模式，让设备学习到合法的源 MAC 地址，以达到相应的网络管理效果。启动了 Port security 功能之后，当发现非法报文时，交换机将触发相应特性，并按照预先指定的方式进行处理，既方便用户的管理又提高了系统的安全性。

2.11.2 PORT SECURITY 配置任务序列

1.PORT SECURITY 基本功能配置

命令	解释
端口配置模式	
switchport port-security no switchport port-security	配置端口的 port-security 功能。
switchport port-security mac-address <mac-address> [vlan <vlan-id>] no switchport port-security mac-address <mac-address> [vlan <vlan-id>]	配置端口的静态安全 MAC。
switchport port-security maximum <value> [vlan <vlan-list>] no switchport port-security maximum <value> [vlan <vlan-list>]	配置端口允许的最大安全 MAC 地址数。
switchport port-security violation {protect recovery restrict shutdown}	当超过设定 MAC 地址数量的最大值，或访问该端口的设备

no switchport port-security violation	MAC 地址不是这个 MAC 地址表中该端口的 MAC 地址，或同一个 VLAN 中一个 MAC 地址被配置在几个端口上时，就会引发违反 MAC 地址安全。
switchport port-security aging {static time <value> type {absolute inactivity}} no switchport port-security violation aging {static time type}	本命令用来使能端口的 port-security 老化表项的功能，指定老化时间以及类型等。
switchport port-security violation shutdown recovery{ <time> } no switchport port-security violation	当端口触发 violation 而 shutdown 后自动恢复为 up 的等待时间。 no 命令删除此操作。 注意：该交换机不支持此命令。
特权配置模式	
clear port-security {all configured dynamic sticky} [[address <mac-addr> interface <interface-id>] [vlan <vlan-id>]]	本命令用来清除端口的安全 MAC 表项。
show port-security [interface <interface-id>] [address vlan]	显示 port-security 配置信息。

2.11.3 PORT SECURITY 举例

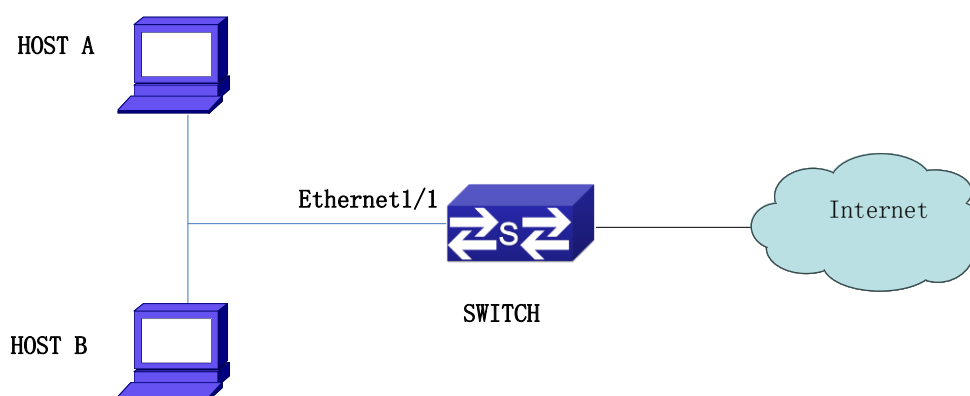


图 2-16 Port security 功能典型组网图

在交换机端口启动 Port security 功能时，配置端口允许的最大安全 MAC 数量为 10，该端口最多只允许 10 个用户接入 Internet，超过最大数量后，新来的用户无法访问 Internet，既可以限制用户的数量又起到安全访问的功能。如果配置端口允许的最大安全 MAC 数量为

1, 则只允许一个用户访问, HOST A 和 B 只有一个能访问网络。

配置步骤:

#配置交换机 Switch

```
Switch(config)#interface Ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#switchport port-security
```

```
Switch(config-if- ethernet1/1)#switchport port-security maximum 10
```

```
Switch(config-if- ethernet1/1)#exit
```

```
Switch(config)#
```

2.11.4 PORT SECURITY 排错帮助

当配置 PORT SECURITY 功能出现问题时, 请检查是否是如下原因:

- ☞ 功能是否正常开启
- ☞ 是否配置了允许的最大MAC数量

2.12 CFM-OAM

2.12.1 概述

以太网技术自诞生起, 就以其简单易用和价格低廉的特点逐步成为局域网的主导技术。近年来, 随着千兆、万兆以太网技术的相继应用, 也促使网络运营商、设备制造商和标准化组织致力于将以太网技术向城域网和广域网领域推进。

然而, 以太网天生就是一项 LAN 技术, 由于在一个 LAN 的环境中, 对远程监控、SLA (服务等级协议) 测试等手段的需求并不迫切, 所以传统的以太网并没有运营网络所需要的 OAM 功能。此外, 传统以太网数十秒或最短 1 秒以上的故障链路倒换时间, 对网络运营商来说, 也是不能接受的。所以, 小于 50 毫秒的故障倒换时间也是城域以太网要面对的一个挑战。

随着几个以太网 OAM 标准 (如 IEEE 802.3ah、IEEE802.1ag 和 ITU Y.1731) 的出台, OAM 已经不再是以太网“标志性”的弱点。以 IEEE802.1ag 为例, 这个也被称为连接故障管理 (CFM) 的标准, 提供了端到端的网络监控和操作工具, 可以在一个大型 L2 以太网中执行诸如 MAC Ping、L2 Trace Route 等任务, 简化了运营级以太网的故障排除和 SLA 监测。同时, 在以太网 OAM 标准中定义的 CCM (连续性检查消息) 可以帮助实现低于 50 毫秒的保护倒换。当前标准中规定的最小 CCM 发送时间是 3.3 毫秒, 3 个连续丢失的 CCM 报文会使主链路宣告“无效”而由备份链路取而代之。CFM 的有效应用建立在合理的网络部署和配置之上。

2.12.1.1 以太网 OAM 的协议规范

关于以太网 OAM, 有四个协议标准: 802.3ah (第一英里以太网, 简称 EFM)、802.1ag

(连接故障管理, 简称 CFM)、E-LMI (以太网本地管理接口)、Y.1731 (故障和性能监测), 由不同的组织制定, 其对应关系如下:

IEEE 802.3ah: Ethernet Link OAM (EFM OAM)

IEEE 802.1ag: Connectivity Fault Management (CFM)

ITU-Y.1731: OAM functions and mechanisms for Ethernet based network

MEF E-LMI: Ethernet Local Management Interface

其中 EFM OAM 和 CFM 为 IEEE 组织制定, EFM OAM 工作在数据链路层, 如图 12-1 所示, 可以有效地发现和管理底层存在的数据链路情况, 我们可以利用 EFM OAM 技术有效提高对以太网的管理和维护能力, 保障网络的稳定运行。CFM 为网络级以太网 OAM 技术, 多应用于网络的接入汇聚层, 用于监测整个网络的连通性、定位网络的连通性故障。Y.1731 是由 ITU, 即国际电信联盟制定的标准, 功能比 CFM 强大许多, 也可以说 CFM 实现的功能是 Y.1731 的子集。E-LMI 是由 MEF 制定的标准, 只应用于 UNI (用户边界和用户面对的供应商边界之间)。

2.12.2 CFM OAM 基本概念

802.1ag 把整个网络 (customer, provider, operator) 划分成很多的维护域 (MD, Maintenance Domain)。在每个维护域内, 针对该 MD 所提供的每一个服务 (service) 进行维护管理, 这些服务会跨越网络中的很多个节点设备。802.1ag 的核心思想是在这个服务所涉及的部分或全部节点上进行维护监控, 从而发现故障并定位出故障段。参与维护监控的节点称为维护点 (MP, Maintenance Point, 又分为 MEP (维护边缘点) 和 MIP (维护中间点)), 维护点配置在桥的端口上。

2.12.2.1 维护域

在逻辑上将网络从内到外划分为不同的层次, 称为维护域 (MD, Maintenance Domain)。维护域可以嵌套, 但不可以交叉。每个维护域的维护者只能看到自己的维护域, 底层的维护域为它直接相邻的上层提供服务。这样, 可将网络的维护者 (如运营商) 同使用者 (如用户) 清晰地隔离开, 方便后续网络问题的定位。

网络中可以有多个维护域, 每个维护域都处于特定的级别, 共有 8 个级别 (level), 级别越高, 维护的网络范围越大, 高级别的域可以嵌套低级别的域。在实践中, CFM 通常会应用到如下图场景中, 此时会划分用户域 (customer domain), 服务提供商域 (service provider domain), 运营商域 (operator domain) 等。

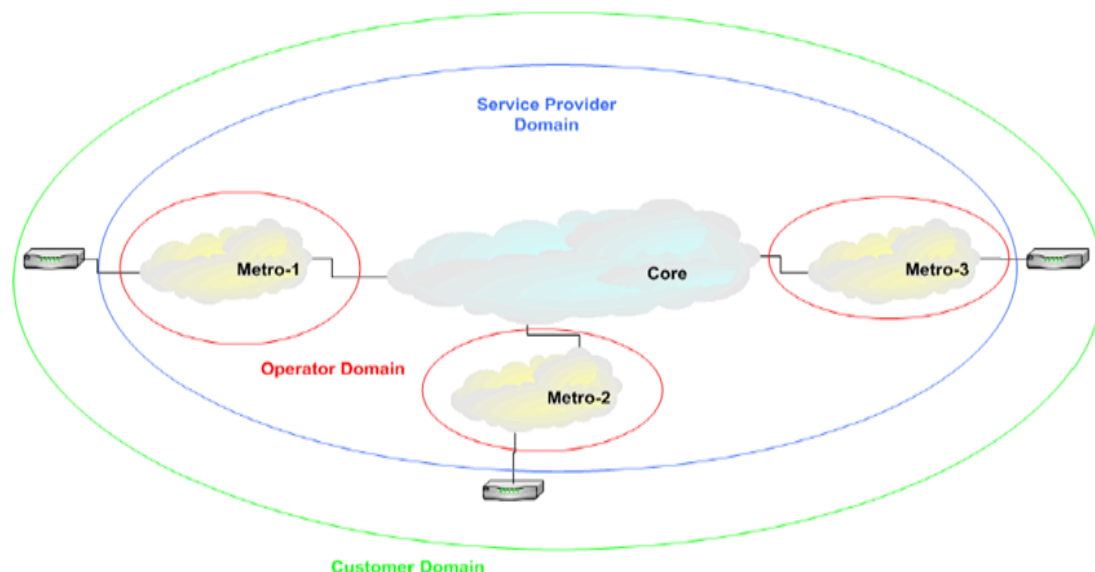


图 2-17 维护域

除了 level，每个 MD 都有一个全局唯一的 MD Name，用于标识该 MD。

2.12.2.2 维护集

维护域内的每一个服务实例（service instance）就是一个维护集（MA，Maintenance Association）。一个 MD 通常对外提供多个服务实例，802.1ag 就是对一个个 MA 进行管理维护，检测 MA 中的故障。每个 MA 都具有一个 MD 内唯一的名字。在网络中，服务实例通常由 vlanId 标识，而 MA 又跟服务实例对应，所以 MA 跟 vlanId 之间有一定关联。一个 MA 对应着一个 vlanId，同时，协议允许多个 vlanId 受一个 MA 来管理维护，在这些 vlanId 中，有一个 vlan 是 primary vlan。在一个 MD 内，对于一个 vlanId 对应多个 MA 的情况，协议中存在不确定的情况，所以暂不考虑。综上可知，md 和 primary vlan Id 可以唯一地标识一个 MA。

2.12.2.3 维护端点

它属于某个维护集，是维护集的边界点，配置在端口上。MEP 负责发起所有的 CFM 报文（CCM，LTM，LBM），CFM 主要的协议行为和状态机都发生在 MEP 上。MEP 分为 UP MEP 和 DOWN MEP。在一个桥里面，如果一个 MEP 从 Lan 侧发送和接收对应 MA 的 CFM 报文，那么这个 MEP 就是 Down MEP；如果一个 MEP 从 Bridge Relay 侧发送和接收它对应的 MA 的 CFM 报文，那么这个 MEP 就是 Up MEP。如图 12-2 所示。

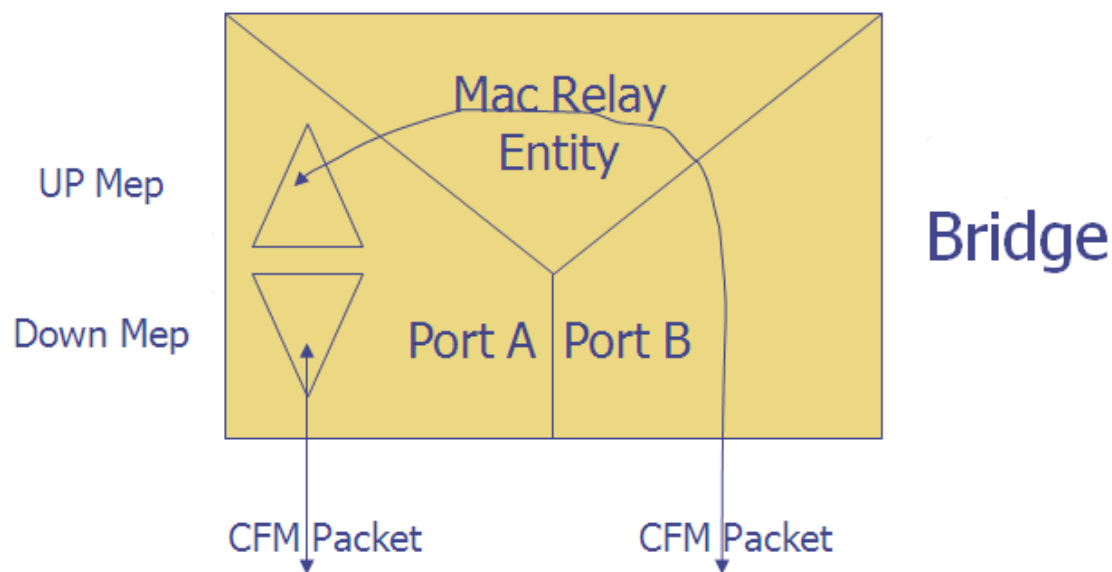


图 2-18 Down MEP 和 Up MEP 区别

通俗地讲就是，MEP 从本地端口接收和发送 CFM 报文时，它是 Down MEP，从非本地端口接收和发送 CFM 报文时，它是 Up MEP。MEP 继承了其所在的 MD 和 MA 的属性，即 MD level + vlan Id。维护端点在 MA 内用一个整数唯一地标识，称为 MEPID。

2.12.2.4 维护中间点

它属于某个维护集，是维护集的中间点，配置在端口上。MIP 不能主动发送 CFM 报文，只能响应收到的 CFM 报文，发送 Reply 以及转发。它不负责检测和上报故障，但可辅助 MEP 进行故障检测。MIP 继承了其所在的 MD 和 MA 的属性，即 MD level + vlan Id。

MIP 并不是直接配置的，它是由系统根据一定的规则生成的。在一个端口上，每个维护集最多只有一个 MIP。生成 MIP 的三种规则如下：

none：不创建 MIP 节点。

default：若指定端口上没有更高级别的 MEP，同时也不存在更低级别的 MIP，则在指定端口上创建本级别的 MIP。

explicit：若指定端口上没有更高级别的 MEP，但存在低级别的 MEP，同时也不存在低级别的 MIP，则在指定端口上创建本级别的 MIP。

defer：对于是否创建 MIP 节点，创建规则由包含本 MA 的 MD 配置的创建规则决定。

2.12.3 CFM OAM 功能介绍

802.1ag 提供了 5 项功能：故障检测（Fault detection），故障确认和定位（Fault verification and isolation），路径发现（Path discovery），故障通知（Fault notification）和故障恢复（Fault recovery）。其中，故障恢复要联合其它协议一起执行。

2.12.3.1 故障检测

维护端点（MEP）周期性地向同一维护集（MA）内的其它远端点（Remote MEP）发送 CCM 报文，同时，也接收其它远端点发来的 CCM 报文。如果在三个周期内没有收到对方的 CCM 报文，则认为链路出现故障，报告故障。如图 12-3 所示。

CCM 报文可发送给同一维护集（MA）内的所有其它 MEP，它是组播报文。组播地址的最后三个 bit 代表不同的维护域 level，这样可方便硬件处理。对于较低级别的 MEP，在收到较高级别的报文后，硬件可根据组播 MAC 地址和 VLAN 直接进行转发。

CCM 报文的周期发送可检测如下故障：

- MEP 不能按时收到 CCM 报文，表示维护集内出现链路故障；
- MEP 检查收到的 CCM 报文，可检测出发送间隔配置不一致故障；
- MEP 接收到故障的 MEPID 或 MAID，表示 MA 内配置故障或者交叉连接故障；
- MEP 接收到低级别的 CCM 报文，表示 MA 内配置故障或者交叉连接故障；
- MEP 接收携带 MAC 状态信息的 CCM 报文，可监视紧贴维护集外侧的故障；

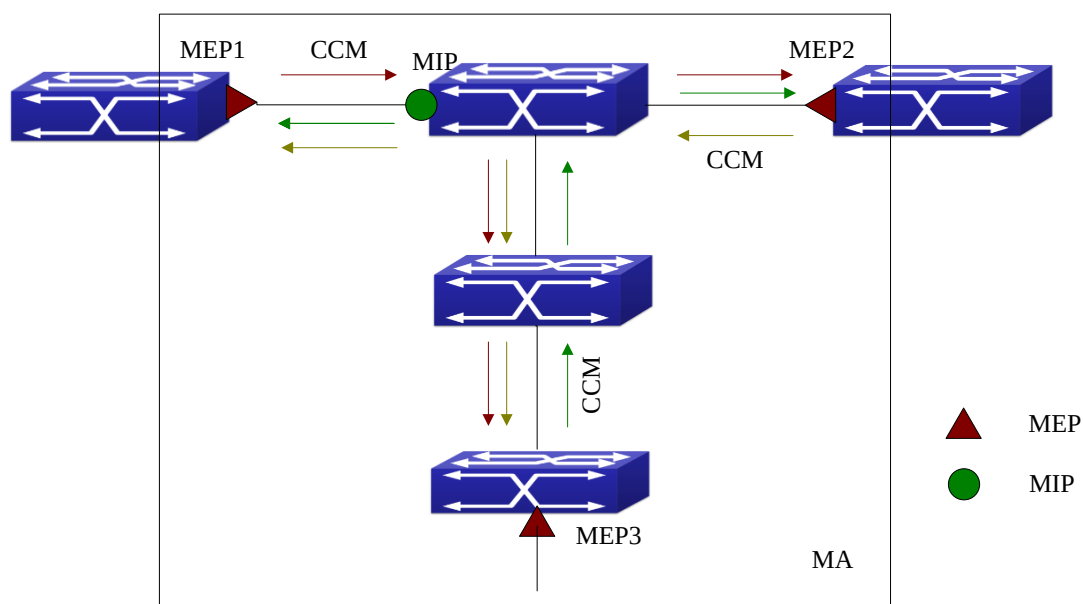


图 2-19 连通性检测示意图

2.12.3.2 路径发现

由 MEP 和 MIP 通过 Linktrace 手段来完成。802.1ag Linktrace 功能与 IP Traceroute 功能类似，通过发送测试报文和接收应答报文来检测从本设备到目的设备的路径或定位故障点。具体过程如下：MEP 发送链路跟踪报文（LTM）给目标 MP（MEP 或 MIP），链路中间的每一个 MIP 在收到 LTM 后都发送一个链路跟踪应答报文（LTR）给源 MEP，然后继续转发 LTM 报文，直到 LTM 到达目的 MP 或无法继续转发为止。源 MEP 根据返回的 LTR 确认链路情况，获得到达目标 MAC 所经过的路径。如图 12-4 所示。

链路跟踪报文（LTM）的目的 MAC 地址是组播地址。组播地址的最后三个 bit 代表不同的维护域 level。链路跟踪应答报文（LTR）是单播报文，目的地址为 LTM 的源 MAC 地址。

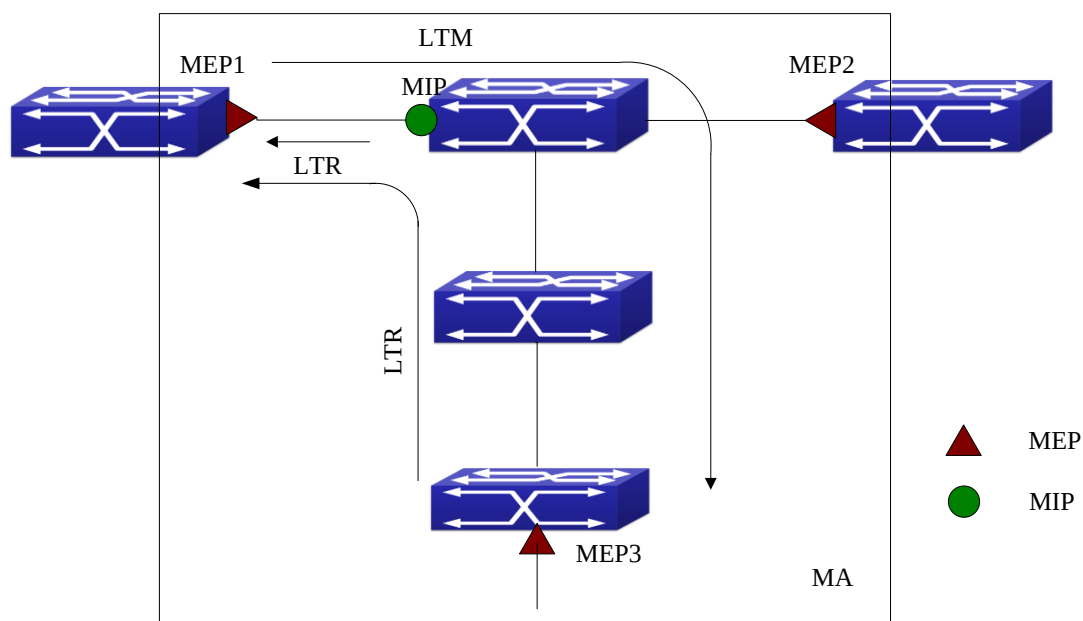


图 2-20 路径发现示意图

2.12.3.3 故障确认和定位

它可由上面提到的 802.1ag Linktrace 功能实现,也可由 802.1ag Loopback 功能实现。环回功能与 IP Ping 类似,通过发送测试报文和接收应答报文来检测从本设备到目的设备是否可达。具体过程如下: MEP 发送单播环回报文 (LBM), 报文的目的地址为远端 MP, 中间设备收到 LBM 后转发,远端目的 MP 在收到 LBM 后,向源 MEP 发送环回应答报文(LBR),源 MEP 据此可判断远端 MP 是否可达。如图 12-5 所示。

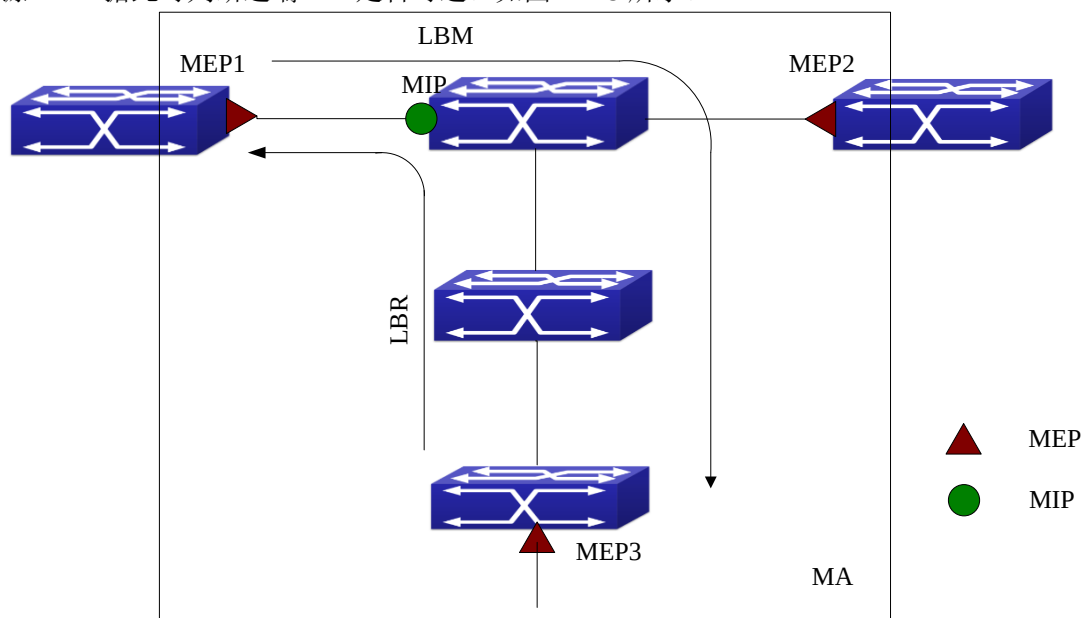


图 2-21 环回功能示意图

2.12.3.4 故障通知

CFM 检测到链路故障后，可有多种处理方式：

检测到链路故障后，由 MEP 发送 SNMP TRAP 消息到管理节点，提醒故障；

检测到链路故障后，由 MEP 将故障信息记录到设备日志，管理员查看日志发现问题；

检测到链路故障后，与其它自动保护协议（APS 等）配合，进行故障恢复。CFM 将故障事件通知这些协议，由协议自动完成链路切换。

2.12.4 CFM OAM 基础功能配置

2.12.4.1 CFM 管理拓扑设计

执行 CFM OAM 功能前需先对网络进行如下规划工作：

1. 对整个网络的维护域进行分级，确定各级别维护域的边界。
2. 确定各维护域的名称，同一维护域在不同设备上的名称相同。
3. 根据需要监控的 VLAN，确定各维护域内的维护集。
4. 确定各维护集的名称，同一维护域内同一维护集在不同设备上的名称相同。
5. 确定同一维护域内同一维护集的维护端点列表，在不同设备上应保持相同。
6. 在维护域和维护集的边界端口上应规划维护端点，非边界设备或端口上可规划维护中间点。

因此需要准备以下数据。

序号	数据
1	MD 的名称和级别
2	MA 的名称、MA 关联的 VLAN 的 ID
3	MEP 的 ID、MEP 所在接口的名称、MEP 的类型
4	RMEP 的 ID
5	MIP 的生成规则
6	MA 内 MEP 发送或探测 CCM 消息的时间间隔

2.12.4.2 CFM OAM 配置任务序列

1. 选择使能 CFM OAM 功能模式
2. 全局使能 CFM OAM 功能
3. 全局使能 y1731 功能（可选）
4. 创建 MD
5. 创建 MA
6. 创建 MEP
7. 配置 RMEP
8. 创建 MIP（可选）
9. 使能故障指示功能（可选）
10. 配置 CC 发送及检测

11. 检查 CFM 配置结果

1. 选择使能 CFM OAM 功能模式

命令	解释
全局配置模式	
ethernet cfm mode {hw sw auto} no ethernet cfm mode	选择交换机使能 CFM OAM 的模式，只能在未使能 CFM OAM 功能之前使用。 no 操作将设备支持方式恢复为缺省方式自动。

2. 全局使能 CFM OAM 功能

命令	解释
全局配置模式	
ethernet cfm global no ethernet cfm global	全局使能 CFM OAM 功能。 no 操作关闭全局使能 CFM OAM 功能。

3. 全局使能 y1731 功能（可选）

命令	解释
全局配置模式	
ethernet cfm y1731 global no ethernet cfm y1731 global	全局使能 y1731 功能。 使能此功能后，交换机进入 y1731 模式，发送的报文都是 Y1731 的报文格式，处理接收的报文也按照 Y1731 报文格式进行解析。需要注意的是：在使用此命令之前必须先使用命令 ethernet cfm global ，否则此命令不能生效。 no 操作取消使能 y1731 功能。

4. 创建 MD

命令	解释
全局配置模式	
ethernet cfm domain < domain-name > level < level-id > no ethernet cfm domain < domain-name >	创建 MD 并进入 MD 模式。 MD 一旦创建成功，则不允许更改所属的 level。 no 操作删除已创建的 MD
MD 配置模式	

id {mac-address XX-XX-XX-XX-XX-XX domain-number < domain-number > dns < dns-name > null } no id	配置 MDID。维护域的名字以命令行 ethernet cfm domain 配置的 domain-name 不再填充在报文中，填充 此 MDID 与 ma name 形成 MAID, MAID 的总长度为 44，不能超过此长度，否则 将会报错。 no 操作删除已配置的 id。

5. 创建 MA

命令	解释
MD 配置模式	
service { < ma-name > number < ma-num > pvlan < vlan-id > } { port pvlan < vlan-id > } [vlan < WORD >] [direction down] no service { < ma-name > number < ma-num > pvlan < vlan-id > }	创建 MA，配置 MA 的 UP/DOWN 属性，并进入 MA 模式。 一个 MA 只能关联一个 vlan；MA 一旦创建成功，则不允许更改它的关联 vlan 及 UP/DOWN 属性。若需要更改，请删除 MA 之后重新创建。 一台设备上最多创建 512 个 MA。 no 操作删除已创建的 MA

6. 创建 MEP

命令	解释
MA 配置模式	
mep mepid < WORD > no mep mepid [< WORD >]	创建该 MA 内所有的 MEP。no 操作删除创建的 MEP。
端口配置模式	
ethernet cfm mep < mepid > domain < domain-name > service { < ma-name > number < ma-num > pvlan < vlan-id > } no ethernet cfm mep < mepid > domain < domain-name > service { < ma-name > number < ma-num > pvlan < vlan-id > }	端口上创建 MEP。 MEP 的属性在创建 MA 的时候已经形成，若 MA 为 UP/DOWN 性质，则此 MA 内所有的 MEP 点均为 UP/DOWN 性质。 no 操作在端口上删除已创建的 MEP。

7. 配置 RMEP

命令	解释
MA 配置模式	

continuity-check receive rmep <mep-id> [active time <time>] no continuity-check receive rmep <mep-id>	配置 RMEP。 一个 MA 内若 mepid 已经被配置为 MEP，则不允许继续配置为 RMEP。 no 操作删除已配置的 RMEP。

8. 创建 MIP（可选）

命令	解释
MD 配置模式；MA 配置模式	
mip auto-create [lower-mep-only none] no mip auto-create	在该 MD/MA 内自动创建 MIP。 缺省情况下，没有配置维护中间点的创建规则，也不存在维护中间点；默认不携带 sender-id。 no 操作删除该 MD/MA 内自动生成的 MIP。
全局模式	
ethernet cfm mip auto-create level <level-id> vlan <WORD> [lower-mep-only] [sender-id chassis] no mip auto-create	配置在不被 MA 所关联的 level 上自动创建 MIP。 缺省情况下，没有配置维护中间点的创建规则，也不存在维护中间点；默认不携带 sender-id。 no 操作删除该 level 内自动生成的 MIP。

9. 使能故障指示功能（可选）

命令	解释
全局配置模式	
ethernet cfm alarm {delay <mseconds> notification { all error-xcon mac-remote-error-xcon none remote-error-xcon xcon} reset <mseconds>} no ethernet cfm alarm { delay notification { all error-xcon mac-remote-error-xcon none remote-error-xcon xcon} reset }	使能故障指示功能。no 操作恢复故障指示为缺省配置。
ethernet cfm logging no ethernet cfm logging	开启 CFM 记入 log 功能。开启 log 记录功能后，若发生 alarm，则进行记入日志或提示故障。 默认发生 alarm 记入日志。 no 操作关闭 CFM 记录 log 功能。

ethernet cfm snmp-server enable traps no ethernet cfm snmp-server enable traps	使能发生 alarm 时进行 snmp trap 通知。使能成功之后，若发生 alarm，则进行 snmp trap 通告或提示故障。默认开启故障 trap 通知功能。 no 操作关闭故障 trap 通知功能。

10. 配置 CC 发送及检测

命令	解释
MA 配置模式	
continuity-check enable no continuity-check enable	开启本地 CCM 报文的发送及检测。 此命令使能本设备上该 MA 的 MEP 周期性发送及接收 CCM 报文。 no 操作取消本地 CCM 报文发送及检测。
continuity-check interval < interval-value > no continuity-check interval	配置 MEP 发送的 CCM 报文中时间间隔域的值。软件模式下最小发送周期为 100ms(interval value=3)，硬件模式下最小发送周期为 3.3ms (interval value=1)。 no 操作恢复 CCM 发送周期为 1s (interval value=4)。

11. 检查 CFM 配置结果

命令	解释
特权模式	
show ethernet cfm domain { < domain_name > brief }	查看 MD 的配置消息。
show ethernet cfm service [domain < domain-name > [service { ma-name number < ma-num > pvlan < vlan-id > }]]	查看 MA 的配置消息。
show ethernet cfm maintenance-points local [detail] [mep mip] [domain < domain-name > interface { ethernet } <IFNAME>]	查看 MEP 及 MIP 的配置消息。

show ethernet cfm maintenance-points remote detail (mac XX-XX-XX-XX-XX domain WORD (service ((WORD))(number <0-65535>)((pvlan <1-4094>))) mepid <1-4094>)	查看 RMEP 的详细配置消息。
show ethernet cfm maintenance-points remote (domain WORD (service (WORD number <0-65535> pvlan <1-4094>) (mepid <1-4094>))))	查看 RMEP 的运行消息。
show ethernet cfm statistic [domain < domain-name > [service { ma-name number < ma-num > pvlan < vlan-id > }]]	查看 CFM 的报文收发信息。

2.12.5 CFM OAM 故障确认

2.12.5.1 管理拓扑确认

执行故障确认之前，请保证完成 CFM OAM 功能配置。

手动检测故障需要准备以下数据：

序号	数据
1	MD 的名字
2	MA 的名字
3	目的 MEP 的 ID 或者 MAC
4	loopback 功能需要发送的报文个数、大小、超时时间。
5	linktrace 功能需要发送的 TTL 值。

2.12.5.2 执行环回功能

特权模式下，执行命令：

命令	解释
特权模式	

ping ethernet [target-mep < mepid > target-mac < mac-address >] { domain < domain-name > service { < ma-name > number < ma-num > pvlan < vlan-id > } } [number < number >] [packet-size < size >] [timeout < timeout >]	开启环回功能，从指定的维护端点向本维护集中其它的维护点发送 LBM 报文并接收 LBR 报文。 缺省情况下，环回功能未启用。若输入的是 target-mep 时，则查找其所对应的 mac 地址，若查找不到，则提示故障。若输入的是 mac 地址，则直接根据此 mac 地址作为目的地址进行环回。若输入为 domain-name，则需先开启 y1731 功能，才能发出组播 LBM 报文。

2.12.6 CFM OAM 故障定位

2.12.6.1 管理拓扑确认

执行故障确认之前，请保证完成 CFM OAM 功能配置。

手动检测故障需要准备以下数据：

序号	数据
1	MD 的名字
2	MA 的名字
3	目的 MEP 的 ID 或者 MAC
4（可选）	最大跟踪跳数
5（可选）	只查询 FDB；或需要同时查询 FDB 和 MIP 数据库来进行故障确认
6（可选）	发起 LTM 的 MEP 的 ID

2.12.6.2 CFM OAM 故障定位任务序列

1. 执行 linktrace 功能

命令	解释
特权模式	
traceroute ethernet { target-mep < target-mep-id > target-mac < mac-address > } { domain < domain-name > service { < ma-name > number < ma-num > pvlan < vlan-id > } } [fdb-only source < mepid >] [ttl < ttl-value >]	查找指定维护端点到目的维护点的路径。 缺省情况下发出的 LTM 报文 ttl=64，而且查询 FDB 及 MIP 数据库。

2. 配置 auto-linktrace 功能（可选）

命令	解释
MA 配置模式	
traceroute ethernet auto no traceroute ethernet auto	开启自动发送链路跟踪报文功能。缺省情况下不开启自动发送链路跟踪报文功能。 使能本功能后，当维护端点在 3.5 个 CCM 报文发送周期内未收到远端维护端点发来的 CCM 报文，从而判定同远端维护端点的连接出错时，将发送 LTM 报文（该 LTM 报文的目标为端维护端点，LTM 报文中 TTL 字段为最大值 255），通过检测回应的 LTR 报文来定位故障。 no 操作关闭自动发送链路跟踪报文功能。
全局配置模式	
ethernet cfm auto-traceroute cache { size < size-value > hold-time <minutes>} no ethernet cfm auto-traceroute cache { size hold-time }	配置保存自动 LT 检测结果的大小。 缺省情况下，缓冲区只记录最近 5 次自动检测的结果；超时时间为 100 分钟。 no 操作恢复 LT 检测结果为缺省值。

3. 查看 auto-linktrace 结果

命令	解释
特权模式	
show ethernet cfm traceroute-reply auto [domain < domain_name > [service { ma-name number < ma-num > pvlan < vlan-id > }]]	显示自动 LT 的结果。 如果没有指定 domain，则显示该设备上的所有自动 LT 的结果。 如果没有指定 ma，则显示该设备上的特定 domain 内的所有自动 LT 的结果。

2.12.7 ULPP 联动（可选）

2.12.7.1 ULPP 联动任务序列

1. 配置 ulpp 联动

- ☞ 请按照拓扑配置完毕并确保 CC 功能运行正常。
- ☞ 必须先配置好 ULPP 功能。
- ☞ ULPP 联动只与 down MEP 联动

☞ 联动 MEP 所关联的 vlan 必须在对应 ULPP 组的保护 vlan 里。

命令	解释
端口配置模式	
switchport ulpp group <group-id> track cfm cc level <level-value>	配置 ulpp 组成员端口与 cfm cc 检测联动的 level 信息。当 ulp 组成员端口接收到与配置匹配的 cfm 信息（超时或超时恢复）时，进行联动操作。

2. 检查 ULPP 联动配置结果

命令	解释
特权模式	
show ulpp group <word>	检查 ULPP 联动配置的结果，若配置成功则 Track-cfm-level 部分会有 MD 的 level 显示。

2.12.7.2 ULPP 联动应用举例

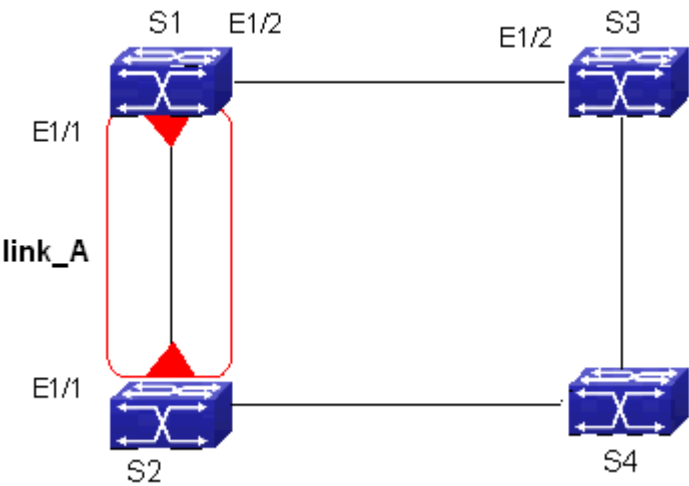


图 2-22 ULPP 联动配置图

1.配置思路

- 采用如下思路配置 ULPP 联动功能
- 创建 VLAN，并将相关接口加入对应的 VLAN。
- 在 S1、S2 上创建 MD link_A，级别为 4
- 在 customer_A 下创建并配置 MA1。（MA1 和 VLAN1 关联。）
- 在 S1、S2 上配置相应的 MEP 及 RMEP
- 在 S1 上创建 ulpp group1，并配置主副端口
- 在 S2、S3 上配置相应 flush 报文接收端口
- 在 S1 上配置 ulpp 联动端口。

2.配置步骤

- (1)创建 VLAN，并将相关接口加入才对应的 VLAN（略）

(2)在 S1、S2 上创建 MD link_A 及相应的 MEP、RMEP

#在 S1 上创建 link_A 并配置相应的 MEP 及 RMEP

```
Switch(config)#ethernet cfm domain link_A level 4
```

```
Switch(config-ecfm)#service MA1 pvlan 1 direction down
```

```
Switch(config-ecfm-srv)#mep mepid 1-2
```

```
Switch(config-ecfm-srv)#continuity-check enable
```

```
Switch(config-ecfm-srv)#continuity-check receive rmep 2
```

```
Switch(config-ecfm-srv)#exit
```

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

```
Switch(config)#interface ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 1 domain link_A service MA1
```

在 S2 上用相同办法创建 link_A 及对应 MEP、RMEP

(3)在 S1 上创建 ulpp group 1 ， 保护 vlan1

#在 S1 上创建 ulpp group 1 并开启抢占模式， 保护 vlan1

```
Switch(config)#spanning-tree mst configuration
```

```
Switch(config-mstp-region)#instance 1 vlan 1
```

```
Switch(config-mstp-region)#exit
```

```
Switch(config)#ulpp group 1
```

```
Switch(ulpp-group-1)#protect vlan-reference-instance 1
```

```
Switch(ulpp-group-1)#flush enable mac-vlan
```

```
Switch(ulpp-group-1)#preemption mode
```

#将 S1 的 1/1 配置成 master 端口， 1/2 配置成 slave 端口

```
Switch(config)#interface ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#ulpp group 1 master
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(config-if-ethernet1/2)#ulpp group 1 master
```

#在 S2 的接收端口 1/1 上配置接收 flush 报文

```
Switch(config-if-ethernet1/1)#ulpp flush enable mac-vlan
```

在 S3 上用相同办法在接收端口 1/2 上配置接收 flush 报文

(4)在 S1 配置 ulpp 联动

#在 S1 的 1/1 端口上配置 ulpp 联动

```
Switch(config-if-ethernet1/1)#switchport ulpp group 1 track cfm level 4
```

3.检查配置结果

#在 S1 上检查 ulpp 联动配置结果

```
Switch(config)#show ulpp group 1
```

ULPP group 1 information:

Description:

Preemption mode: ON

Preemption delay: 30s
Control VLAN: 1
Flush packet: MAC MAC-VLAN ARP
Protected VLAN: Reference Instance 1

Member	Role	State	Track-cfm-level
Ethernet1/1	MASTER	FORWARD	4
Ethernet1/2	SLAVE	STANDBY	-

#若 CFM 检测到 CC 超时，则通知 ULPP 模块进行主副端口切换：
%Jan 01 00:12:19 2012 ULPP: ULPP group 1:Master port Ethernet1/3 receives cfm event
type:CFM_ALARM_RMEP_CCM vlan:1 level:4.
%Jan 01 00:12:19 2006 ULPP: ULPP group 1:Master port Ethernet1/3 change state to
down,Slave port Ethernet1/1 change state to forwarding.
%Jan 01 00:12:21 2006 CFM:A CFM_ALARM_RMEP_CCM of Interface Ethernet1/3 is
detected.

2.12.8 配置应用举例

如下图所示是 CFM 的配置应用的说明图，为了实现检测链路状态，可以按如下的步骤进行配置。

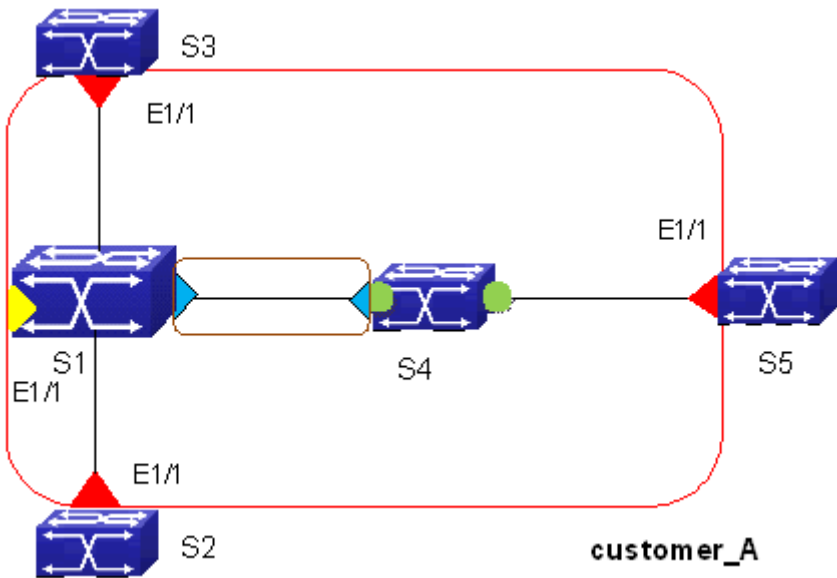


图 2-23 MD, MA, MEP 和 MIP 配置图

一、配置思路

采用如下的思路配置以太网 CFM 基本功能：
创建 VLAN，并将相关接口加入对应的 VLAN。

在 S1、S2、S3、S4、S5 设备上创建 customer_A，customer_A 的级别为 6。

在 customer_A 下创建并配置 MA1。（MA1 和 VLAN1 关联。）

在 S1、S5 设备上创建 operator_A，operator_A 的级别为 3，

在 operator_A 上创建并配置 MA2。（MA2 和 VLAN2 关联）

在 S1、S2、S3、S4 上的 customer_A 的 MA1 下创建本地和远端 MEP。

在 S4 上的 customer_A 的 MA1 下创建 MIP

在 S1、S5 上的 operator_A 的 MA2 下配置本地和远端 MEP。

使能 CCM 消息的发送和接收功能。

二、配置步骤

1.创建 VLAN，并将相关接口加入才对应的 VLAN（略）

2.全局开启 CFM 功能并创建 customer_A 并配置 MA1

#在 S1 上创建 customer_A，并配置 UP 方向的 MA1

```
Switch(config)#ethernet cfm global
```

```
Switch(config)#ethernet cfm domain customer_A level 6
```

```
Switch(config-ecfm)#service MA1 pvlan 1
```

3.在 S2、S3、S4、S5 上创建 customer_A 并配置 MA1

#在 S2 上创建 customer_A，并配置 Down 方向的 MA2

```
Switch(config)#ethernet cfm global
```

```
Switch(config)#ethernet cfm domain customer_A level 6
```

```
Switch(config-ecfm)#service MA1 pvlan 1 direction down
```

在其余设备上用同样的方法创建 MD 和 MA

4.在 S1、S2、S3、S4 上的 customer_A 的 MA1 下创建本地和远端 MEP，在 S5 上创建 MIP

#在 S1 的 MA1 内创建 MEP list 为 1-4，配置 RMEP2-4，在 Ethernet1/1 上创建 MEP1。

```
Switch(config-ecfm-srv)#mep mepid 1-4
```

```
Switch(config-ecfm-srv)#continuity-check receive rmap 2-4
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet1/1
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 1 domain customer_A service MA1
```

#在 S2 的 MA1 内创建 MEP list 为 1-4，配置 RMEP1；3-4，在 Ethernet1/1 上创建

MEP2

```
Switch(config-ecfm-srv)#mep mepid 1-4
```

```
Switch(config-ecfm-srv)#continuity-check receive rmap 1; 3-4
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 2 domain customer_A service MA1
```

#在 S3 的 MA1 内创建 MEP list 为 1-4, 配置 RMEP1; 2; 4, 在 Ethernet1/1 上创建 MEP3

```
Switch(config-ecfm-srv)#mep mepid 1-4
```

```
Switch(config-ecfm-srv)#continuity-check receive rmep 1; 2; 4
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 3 domain customer_A service MA1
```

#在 S4 的 MA1 内创建 MEP list 为 1-4, 配置 RMEP1-3, 在 Ethernet1/1 上创建 MEP3

```
Switch(config-ecfm-srv)#mep mepid 1-4
```

```
Switch(config-ecfm-srv)#continuity-check receive rmep 1-3
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 4 domain customer_A service MA1
```

在 S5 的 MA1 内上用 default 规则创建 MIP

```
Switch(config-ecfm-srv)#mip auto-create
```

5. 在 S1、S5 上创建 operator_A 并配置 MA2

#在 S1 上创建 operator_A, 并配置 MA2, 类型为 port

```
Switch(config)#ethernet cfm domain operator_A level 3
```

```
Switch(config-ecfm)#service MA2 port direction down
```

在 S5 上用相同的办法创建 operator_A 及 MA2

6. 在 S1、S5 上的 operator_A 的 MA2 上创建本地和远端 MEP

#在 S1 上创建 MEP list1-2, 配置 RMEP2, 在 Ethernet1/2 上创建 MEP 1

```
Switch(config-ecfm-srv)#mep mepid 1-2
```

```
Switch(config-ecfm-srv)#continuity-check receive rmep 2
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 1 domain customer_A service MA2
```

#在 S2 上创建 MEPlist1-2, 配置 RMEP1, 在 Ethernet1/2 上创建 MEP2

```
Switch(config-ecfm-srv)#mep mepid 1-2
```

```
Switch(config-ecfm-srv)#continuity-check receive rmep 1
```

```
Switch(config-ecfm-srv)exit
```

```
Switch(config-ecfm)exit
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(config-if-ethernet1/1)#ethernet cfm mep 2 domain customer_A service MA2
```

7. 在 S1、S2、S3、S4 上的 MA1 内使能 CCM 消息的发送和接收功能

#在 S1 的 MA1 内使能 CCM 消息的发送和接收功能

Switch(config-ecfm-srv)#continuity-check receive enable

在其余设备上用相同办法开启 CC 功能。

#在 S1、S5 的 MA2 内使能 CCM 消息的发送和接收功能

Switch(config-ecfm-srv)#continuity-check receive enable

8. 在 S1 上查看 customer_A 中 MA1 内的维护端点的具体配置。

Switch#show ethernet cfm maintenance-points local detail mep domain customer_A

Mepid:1

Port:Ethernet1/1

Active:1

Domain Name: customer_A

Service Name:MA1

Level:6

Vlan:1

Direction:Up

CCM:

CC Send:Enable

CC Received:Enable

Interval:1(s)

2.12.9 CFM 排错帮助

在配置、使用 CFM-OAM 时，可能会由于物理连接、配置错误等原因导致功能未能正常运行。

☞ 保证整条链路连接正常，且 MA 需要关联的 vlan 已经存在

1. 配置失败

☞ 保证系统全局启动了 OAM 功能，否则配置任何有关 OAM 的命令均会失败

☞ 配置的 MA/MD 名非法：

MD name: 1~43 个字符的字符串。由字母、数字、下划线组成，首尾字符不能为下划线。

MA name: 1~43 个字符的字符串。由字母、数字、下划线组成，首尾字符不能为下划线。维护集与维护域的名字总和不能超过 44 个字符。

☞ 同级别内，一个 primary vlan 只能被一个 MA 关联。

☞ MD 的 level、MA 关联的 pvlan 及 UP/DOWN 属性在创建之后就不可更改。

☞ 配置 MEP 及 RMEP 之前需要先创建 MEP 的 ID。

☞ 一个 MA 内的 MEP ID，被配置成 MEP 之后不可再配置成 RMEP

☞ 一台设备上一个 DOWN 属性的 MA 内只允许存在一个 MEP；一个 UP 属性的 MA 允许存在多个 MEP，但不允许一个端口上配置多个同一 MA 的 MEP。

2. 无法建立 CC 连接

☞ 通过 show ethernet cfm 命令查看，确保两端的 level、MD name、MA name、MA 所关联的 pvlan 中一致。

☞ 保证本端配置的接收 RMEP 与对端配置的 MEP 的 ID 一致。

☞ 保证本端及对端 MA 开启 CC 发送及检测功能。

- ☞ 保证两端配置的 CCM 发送周期一致。
- ☞ 端口上配置 down mep, 则该 mep 从本端口接收报文; 若配置 up mep, 则该 mep 从其余端口接收报文, 请保证 up mep 配置在非接收端口上。
- ☞ 被 STP 协议阻塞的端口不能接收、发送和回应 CFM 协议报文; 但如果端口被配置为 MEP, 那么即使该端口已被 STP 协议阻塞, 其仍会接收和发送 CCM 报文。只有二层以太网端口支持配置 CFM 功能。
- ☞ MEP 和 MIP 可以配置在 port channel 上, 但是此时 port channel 的成员端口上配置的 MEP 及 MIP 会失效, 若增加 CFM MEP 和 MIP 端口到 port channel 上, 则该端口的 MEP 及 MIP 均会失效。若想恢复该端口上的 MEP 及 MIP, 则需将此端口从 port-channel 中删除。

3. MIP 点未创建

- ☞ 保证 MIP 点生成规则正确
 - default 规则: 在指定端口没有更高级别的 MEP, 在低级别维护集上没有 MIP 时, 本级别创建 MIP。
 - Explicit 规则: 如果在指定端口没有更高级别的 MEP, 在低级别维护集上没有维护中间点时, 本级别是否创建维护中间点依赖于低级别维护集上是否有维护端点。
- ☞ 只有 UP 状态的端口上才会生成 MIP; 一个端口基于一个 vlan 只能生成一个 MIP; 低级别的 MIP 点生成优先。
- ☞ 一个 DOWN 属性的 MA 内只要在端口上配置了 MEP 点, 则不会再生成 MIP, 即使配置了 MEP 的端口加入了 port-channel, 导致此 MEP 失效, 该 MA 内不能生成 MIP 点。

如果使用检查仍无法解决 CFM OAM 的问题, 那么请使用 debug ethernet cfm 等调试命令, 然后将 3 分钟内的 DEBUG 信息拷贝下来, 发送给本公司技术服务中心。

2.13 QSFP+端口拆分与合并

2.13.1 QSFP+端口拆分与合并操作功能介绍

QSFP+端口可以作为一个单独的 40GE 端口使用, 也可以拆分成四个 10GE 的 SFP+ 端口, 从而能够提高端口密度, 减少用户使用成本, 增加组网灵活性。比如, QSFP+ 端口 Ethernet5/21 缺省为 40GE, 可以根据需要拆分成四个 10GE 的 SFP+ 端口 Ethernet5/21:1、Ethernet5/21:2、Ethernet5/21:3、Ethernet5/21:4。拆分出来的 SFP+ 端口, 支持的配置和特性均和普通 SFP+ 物理端口相同。QSFP+ 端口拆分后需要使用一分四的专用线缆连接。

如果用户需要更大的带宽，可以将已拆分的四个 10GE 端口合并为一个 40GE 端口使用。合并后，需要将一分四的专用线缆连接更换成一对一的专用线缆，关于线缆的具体描述请参见产品的相关手册。

QSFP+端口拆分或合并操作必须进行配置保留，并且重启交换机后生效。

2.13.2 QSFP+端口配置

QSFP+端口配置任务序列如下：

1. 进入端口配置模式
2. QSFP+端口拆分/合并操作
3. 配置保留
4. 重启交换机

1. QSFP+端口拆分/合并操作

命令	解释
全局配置模式	
hardware profile mode 4x10G no hardware profile mode 4x10G	将 QSFP+端口拆分为四个 10GE 端口，本命令的 no 操作将已拆分的四个 10GE 端口合并为一个 40GE 端口

2.13.3 QSFP+端口配置典型案例

配置 5/1 的四万兆端口模式为 4*10G 的模式端口。

```
Switch(config)#interface ethernet 5/21
```

```
Switch(config-if-ethernet5/21)# hardware profile mode 4x10G
```

The new configuration will take effect after slot 5 restart!

（重启端口所在的板卡……）

```
Switch(config)#show interface ethernet status
```

Codes: A-Down - administratively down, a - auto, f - force, G - Gigabit

Interface	Link/Protocol	Speed	Duplex	Vlan	Type	Alias Name
5/21:1	DOWN/DOWN	auto	auto	1	XFP	
5/21:2	DOWN/DOWN	auto	auto	1	XFP	
5/21:3	DOWN/DOWN	auto	auto	1	XFP	
5/21:4	DOWN/DOWN	auto	auto	1	XFP	
5/22	DOWN/DOWN	auto	auto	1	QSFP+	

2.13.4 QSFP+端口配置排错帮助

- 可以依次配置多个端口的拆分或合并，只有执行配置操作并重启后方生效。
- 如 40G 模式时端口名为 Ethernet5/21，执行拆分操作后，衍生为四个 10GE 的 SFP+ 端口，Ethernet5/21:1、Ethernet5/21:2、Ethernet5/21:3、Ethernet5/21:4。
- 每个端口配置和特性与普通物理 SFP+端口相同。

合并后，在 40G 模式下使用的是一对一的专用线缆，两端都是 QSFP+模块；拆分后，在 4x10G 模式下使用的是一分四的专用线缆，一端为 QSFP+接口模块，一端为四个 SFP+模块。

2.14 VLAN

2.14.1 VLAN 介绍

VLAN（Virtual Local Area Network）即虚拟局域网，这项技术可以根据功能、应用或者管理的需要将局域网内部的设备逻辑地划分为一个个网段，从而形成一个个虚拟的工作组，并且不需要考虑设备的实际物理位置。IEEE 颁布了 IEEE802.1Q 协议以规定标准化 VLAN 的实现方案，交换机的 VLAN 功能即按照 802.1Q 的标准实现。

VLAN 技术的特点在于可以根据需要动态的将一个大的局域网划分成许多不同的广播域：

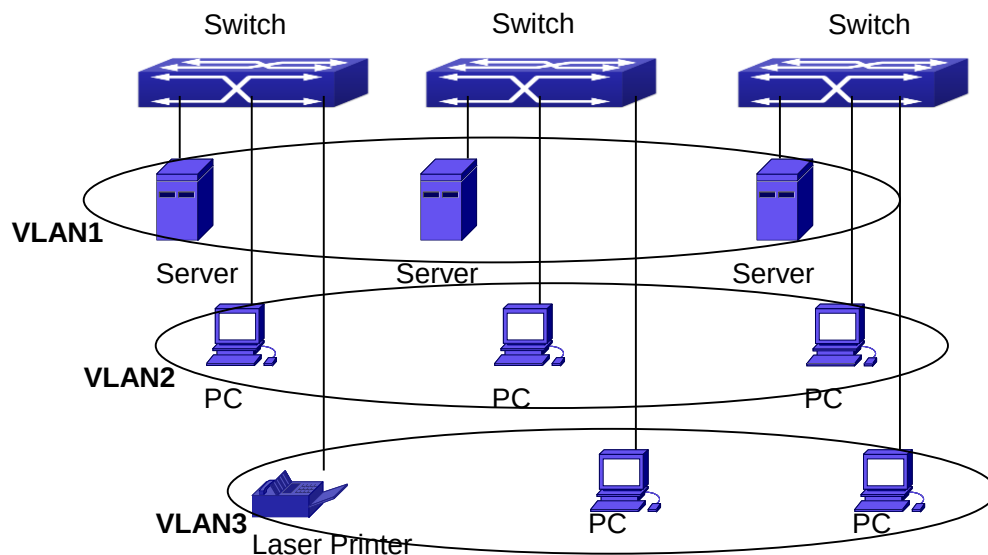


图 2-24 按照逻辑定义的 VLAN 网络

每个广播域即一个 VLAN，VLAN 和物理上的局域网有相同的属性，不同之处只在于 VLAN 是逻辑的而不是物理的划分，所以 VLAN 的划分不必根据实际的物理位置，而每个 VLAN 内部的广播、组播和单播流量都是与其它 VLAN 隔绝的。

基于 VLAN 的以上特性，VLAN 技术给我们带来以下的便利：

- ☞ 改善网络性能
- ☞ 节约网络资源
- ☞ 简化网络管理
- ☞ 降低网络成本
- ☞ 提高网络安全

交换机根据端口在转发报文时对是否带 tag 标签的不同处理方式，把以太网端口分为三种链路类型：Access、Hybrid 和 Trunk。

Access 类型的端口只能属于一个 VLAN，一般用于连接计算机的端口。

Trunk 类型的端口可以允许多个 VLAN 通过，可以接收和发送多个 VLAN 的报文，一般用于交换机之间连接的端口。

Hybrid 类型的端口可以允许多个 VLAN 通过，可以接收和发送多个 VLAN 的报文，可以用于交换机之间连接，也可以用于连接用户的计算机。

Hybrid 端口和 Trunk 端口在接收数据时，处理方法是一样的，唯一不同之处在于发送数据时：Hybrid 端口可以允许多个 VLAN 的报文发送时不打标签，而 Trunk 端口只允许 native VLAN 的报文发送时不打标签。

在交换机中，实现了 802.1Q 所定义的 VLAN 和 GVRP（GARP VLAN Registration Protocol），本章将对交换机 VLAN 和 GVRP 的使用和配置进行详细的说明。

2.14.2 VLAN 的配置任务序列

1. 创建或删除 VLAN
2. 指定或删除 VLAN 名称
3. 为 VLAN 分配交换机端口
4. 设置交换机端口类型
5. 设置 Trunk 端口
6. 设置 Access 端口
7. 设置 Hybrid 端口配置
8. 打开或关闭全局的 VLAN 入口规则
9. 配置 Private VLAN
10. 设置 Private VLAN 的绑定操作
11. 指定内部 VLAN ID

1. 创建或删除 VLAN

命令	解释

全局配置模式	
vlan WORD no vlan WORD	创建/删除 VLAN 或进入 VLAN 模式。

2.指定或删除 VLAN 名称

命令	解释
VLAN 配置模式	
name <vlan-name> no name	设置/删除 VLAN 名称。

3.为 VLAN 分配交换机端口

命令	解释
VLAN 配置模式	
switchport interface <interface-list> no switchport interface <interface-list>	为 VLAN 分配交换机端口。

4.设置交换机端口类型

命令	解释
端口配置模式	
switchport mode {trunk access hybrid}	设置当前端口为 Trunk, Access 端口或 Hybrid 端口。

5.设置 Trunk 端口

命令	解释
端口配置模式	
switchport trunk allowed vlan {WORD all add WORD except WORD remove WORD} no switchport trunk allowed vlan	设置/删除 Trunk 端口允许通过的 VLAN。
switchport trunk native vlan <vlan-id> no switchport trunk native vlan	设置/删除 Trunk 端口的 PVID。

6.设置 Access 端口

命令	解释
端口配置模式	

switchport access vlan <vlan-id> no switchport access vlan	将当前端口加入/退出到指定 VLAN。

7. 设置 Hybrid 端口

命令	解释
端口配置模式	
switchport hybrid allowed vlan {WORD all add WORD except WORD remove WORD} {tag untag} no switchport hybrid allowed vlan	设置/删除 Hybrid 端口允许以 tag 或 untag 方式通过的 VLAN。
switchport hybrid native vlan <vlan-id> no switchport hybrid native vlan	设置/删除端口的 PVID。

8. 打开或关闭 VLAN 的入口规则

命令	解释
端口配置模式	
vlan ingress enable no vlan ingress enable	打开及关闭 VLAN 的入口规则。

9. 配置 Private VLAN

命令	解释
VLAN 配置模式	
private-vlan {primary isolated community} no private-vlan	将当前 VLAN 设置为 Private VLAN。

10. 设置 Private VLAN 的绑定操作

命令	解释
VLAN 配置模式	
private-vlan association <secondary-vlan-list> no private-vlan association	设置/删除 Private VLAN 的绑定关系。

11. 指定内部 VLAN ID

命令	解释
全局配置模式	
vlan <2-4094> internal	指定内部 VLAN 的 ID 号。

2.14.3 VLAN 典型应用

案例：

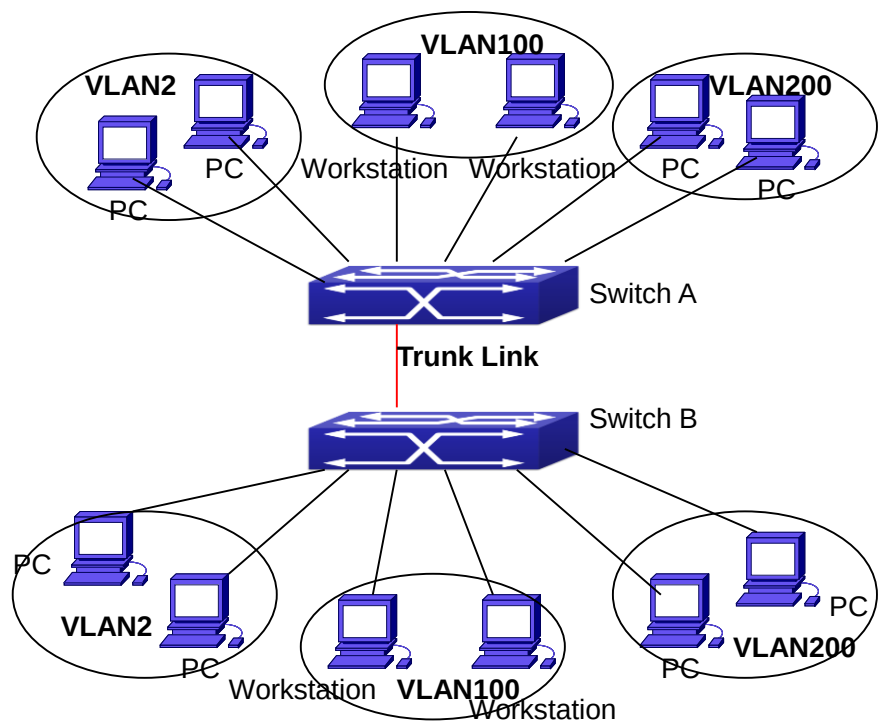


图 2-25 VLAN 典型应用拓扑图

由于局域网安全和应用的需要，需要将现有的整个局域网划分为 3 个 VLAN：VLAN2、VLAN100 和 VLAN200，并且要求这三个 VLAN 跨越两个地域 A 和 B，现在两处分别放置一台交换机，因此 VLAN 流量只要可以在交换机间传输，就可以满足跨地域的要求。

配置项目	配置说明
VLAN2	A 地、B 地交换机 2～4 端口
VLAN100	A 地、B 地交换机 5～7 端口
VLAN200	A 地、B 地交换机 8～10 端口
Trunk port	A 地、B 地交换机的 11 端口

将两台交换机的 Trunk port 相连，成为 Trunk link，用于承载跨交换机的 vlan 流量；将各种网络设备连接到交换机的各个 VLAN 的端口上，就归属到相应的 VLAN 之中。

在本例中，1 号和 12 号端口空闲，可以做为管理端口或其它用途。

配置步骤如下：

A 交换机：

```
Switch (config)#vlan 2
Switch (Config-Vlan2)#switchport interface ethernet 1/2-4
Switch (Config-Vlan2)#exit
Switch (config)#vlan 100
Switch (Config-Vlan100)#switchport interface ethernet 1/5-7
Switch (Config-Vlan100)#exit
Switch (config)#vlan 200
Switch (Config-Vlan200)#switchport interface ethernet 1/8-10
Switch (Config-Vlan200)#exit
Switch (config)#interface ethernet 1/11
Switch (Config-If-Ethernet1/11)#switchport mode trunk
Switch (Config-If-Ethernet1/11)#exit
Switch (config)#
```

B 交换机：

```
Switch (config)#vlan 2
Switch (Config-Vlan2)#switchport interface ethernet 1/2-4
Switch (Config-Vlan2)#exit
Switch (config)#vlan 100
Switch (Config-Vlan100)#switchport interface ethernet 1/5-7
Switch (Config-Vlan100)#exit
Switch (config)#vlan 200
Switch (Config-Vlan200)#switchport interface ethernet 1/8-10
Switch (Config-Vlan200)#exit
Switch (config)#interface ethernet 1/11
Switch (Config-If-Ethernet1/11)#switchport mode trunk
Switch (Config-If-Ethernet1/11)#exit
```

2.14.4 Hybrid 端口的典型应用

案例：

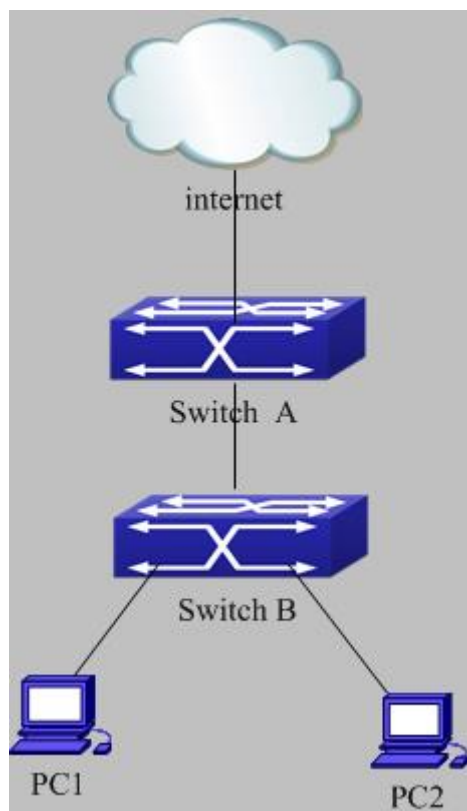


图 2-26 Hybrid 口的典型应用场景

PC1 与 Switch B 的 Ethernet 1/7 口相连，PC2 与 Switch B 的 Ethernet 1/9 口相连，Switch A 的 Ethernet 1/10 与 Switch B 的 Ethernet 1/10 相连。

出于安全的考虑，要求 PC1 和 PC2 之间不能互访，但是 PC1 和 PC2 可以通过网关 Switch A 访问其它的网络资源。这种情况下我们可以通过 Hybrid 端口来实现。

配置项如下：

端口	类型	PVID	允许通过的 VLAN
Switch A 的 1/10	Access	10	允许 VLAN 10 的报文以 untag 方式通过
Switch B 的 1/10	Hybrid	10	允许 VLAN 7,9,10 的报文以 untag 方式通过
Switch B 的 1/7	Hybrid	7	允许 VLAN 7,10 的报文以 untag 方式通过
Switch B 的 1/9	Hybrid	9	允许 VLAN 9,10 的报文以 untag 方式通过

配置步骤如下：

A 交换机：

```
Switch(config)#vlan 10
```

```
Switch(Config-Vlan10)#switchport interface ethernet 1/10
```

B 交换机:

```
Switch(config)#vlan 7;9;10
```

```
Switch(config)#interface ethernet 1/7
```

```
Switch(Config-If-Ethernet1/7)#switchport mode hybrid
```

```
Switch(Config-If-Ethernet1/7)#switchport hybrid native vlan 7
```

```
Switch(Config-If-Ethernet1/7)#switchport hybrid allowed vlan 7;10 untag
```

```
Switch(Config-If-Ethernet1/7)#exit
```

```
Switch(Config)#interface Ethernet 1/9
```

```
Switch(Config-If-Ethernet1/9)#switchport mode hybrid
```

```
Switch(Config-If-Ethernet1/9)#switchport hybrid native vlan 9
```

```
Switch(Config-If-Ethernet1/9)#switchport hybrid allowed vlan 9;10 untag
```

```
Switch(Config-If-Ethernet1/9)#exit
```

```
Switch(Config)#interface Ethernet 1/10
```

```
Switch(Config-If-Ethernet1/10)#switchport mode hybrid
```

```
Switch(Config-If-Ethernet1/10)#switchport hybrid native vlan 10
```

```
Switch(Config-If-Ethernet1/10)#switchport hybrid allowed vlan 7;9;10 untag
```

```
Switch(Config-If-Ethernet1/10)#exit
```

2.15 GVRP

2.15.1 GVRP 介绍

GVRP 即 GARP VLAN 属性注册协议，它是 GARP（通用属性注册）协议的一个应用。GARP 协议主要用于建立一种属性传递扩散的机制，以保证协议实体能够注册和注销该属性。GARP 作为一个属性注册的载体，可以用来传播属性。而根据传播的属性的不同，可以分为很多应用协议，例如 GMRP 和 GVRP。所以说 GVRP 是通过 GARP 这一协议机制，把 VLAN 属性信息传播到整个二层网络的协议。

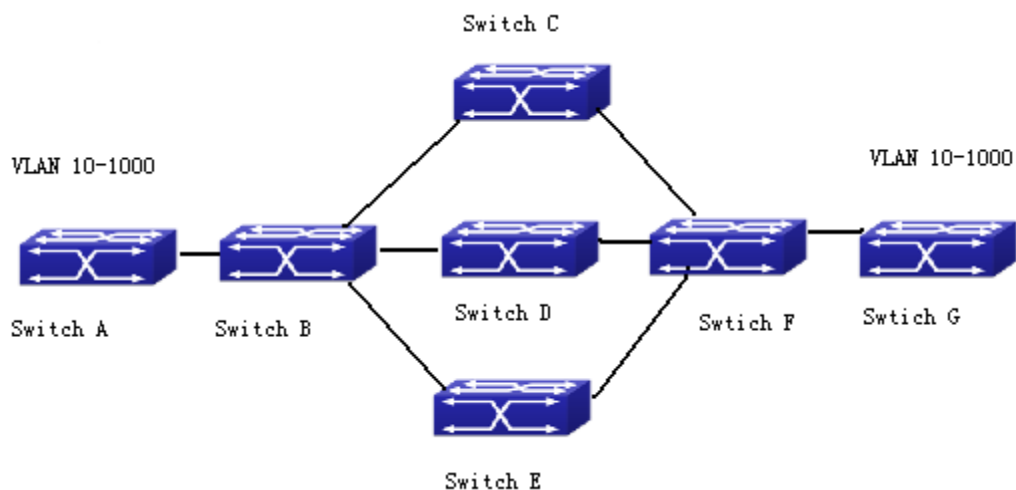


图 2-27 典型应用场景。

其中 A 和 G 是二层网络中没有直连的两个交换机，BCDEF 是二层网络连接 A 和 G 的中间的交换机，交换机 A 和 G 手工配置了 VLAN100-1000，BCDEF 没有此配置，在没有开启 GVRP 的情况下，A 和 G 的 VLAN100-1000 是不能相互通信的，因为中间交换机没有相关 VLAN，但是在所有交换机开启 GVRP 功能之后，通过 GVRP 的 VLAN 属性传播机制使得中间交换机 BCDEF 动态的注册了这些 VLAN，使得 A 和 G 的 VLAN100-1000 中每一个 VLAN 都可以相互通信，即可以看作是一个 VLAN。当 A 和 G 手动注销 VLAN100-1000 后，中间交换机 BCDEF 动态注册的这些 VLAN 也会随之注销。简单说来，通过 GVRP 协议，两个不相邻的交换机的相同 VLAN 可以互相通信，而不必去手动配置中间的每台路由器（这样的手动配置不但麻烦，并且很容易出错），从而达到在复杂的组网环境中简化 VLAN 配置的目的。

2.15.2 GVRP 配置任务序列

GVRP 配置任务序列如下：

- 1. 配置 GARP 定时器参数
- 2. 配置端口类型
- 3. 开启 GVRP 功能

1. 配置 GARP 定时器参数

命令	解释
全局配置模式	
garp timer join <200-500> garp timer leave <500-1200> garp timer leaveall <5000-60000> no garp timer (join leave leaveAll)	配置 GARP 的 leaveall,join 和 leave 定时器。

2. 配置端口类型

命令	解释
端口配置模式	
gvrp no gvrp	开启/关闭端口上 GVRP 功能。

3. 开启 GVRP 功能

命令	解释
全局配置模式	
gvrp no gvrp	全局开启/关闭 GVRP 功能。

2.15.3 GVRP 举例

GVRP 典型应用：

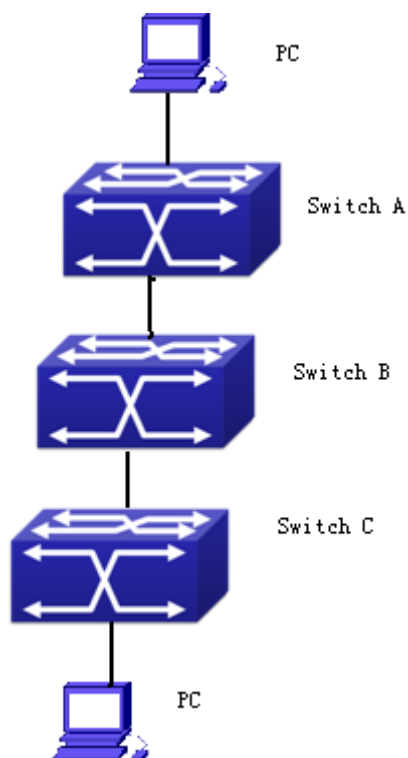


图 2-28 GVRP 典型应用拓扑图

为了实现交换机之间 VLAN 信息的动态注册和更新，需要在交换机上配置 GVRP 协议。在交换机 A、B 和 C 上配置 GVRP，使得交换机 B 学习到动态 VLAN100，这样，分别连接到交换机 A 和 C 的 VLAN100 上的两台工作站就可以通过没有配置静态 VLAN100 的交换机

B 互相通信。

配置项目	配置说明
VLAN100	交换机 A、C 的 2~6 端口
Trunk port	交换机 A、C 的 11 号端口，交换机 B 的 10、11 号端口
全局 GVRP	交换机 A、B、C
端口 GVRP	交换机 A、C 的 11 号端口，交换机 B 的 10、11 号端口

将两台工作站分别与交换机 A 和 B 的 VLAN100 的端口相连，将交换机 A 的 11 号端口与交换机 B 的 10 端口相连，交换机 B 的 11 号端口与交换机 C 的 11 号端口相连。

配置步骤如下：

Switch A:

```
Switch (config)#gvrp
Switch (config)#vlan 100
Switch (Config-Vlan100)#switchport interface ethernet 1/2-6
Switch (Config-Vlan100)#exit
Switch (config)#interface ethernet 1/11
Switch (Config-If-Ethernet1/11)#switchport mode trunk
Switch (Config-If-Ethernet1/11)#gvrp
Switch (Config-If-Ethernet1/11)#exit
```

Switch B:

```
Switch(config)#gvrp
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#switchport mode trunk
Switch(Config-If-Ethernet1/10)#gvrp
Switch(Config-If-Ethernet1/10)#exit
Switch (config)#interface ethernet 1/11
Switch (Config-If-Ethernet1/11)#switchport mode trunk
Switch (Config-If-Ethernet1/11)#gvrp
Switch (Config-If-Ethernet1/11)#exit
```

Switch C:

```
Switch (config)#gvrp
Switch (config)#vlan 100
Switch (Config-Vlan100)#switchport interface ethernet 1/2-6
Switch (Config-Vlan100)#exit
```

```
Switch (config)#interface ethernet 1/11
Switch (Config-If-Ethernet1/11)#switchport mode trunk
Switch (Config-If-Ethernet1/11)#gvrp
Switch (Config-If-Ethernet1/11)#exit
```

2.15.4 GVRP 排错帮助

Trunk 链路两端 Trunk 端口的 GARP 计数器设置必须相同, 否则 GVRP 不能正常工作。交换机的 GVRP 功能不能和 RSTP 同时打开, 如果要打开 GVRP 功能, 请首先关闭端口的 RSTP 功能。

2.16 Dot1q-tunnel

2.16.1 Dot1q-tunnel 介绍

Dot1q-tunnel 也称 QinQ (802.1Q-in-802.1Q), 是对 802.1Q 的扩展, 其核心思想是将用户私网 VLAN 标签 (CVLAN tag) 封装到公网 VLAN 标签 (SPVLAN tag) 上, 报文带着两层 VLAN 标签穿越服务商的骨干网络, 从而为用户提供一种较为简单的二层隧道。它简单而易于管理, 仅仅通过静态配置即可实现, 特别适用于小型的以三层交换机为骨干的企业网或小规模城域网。

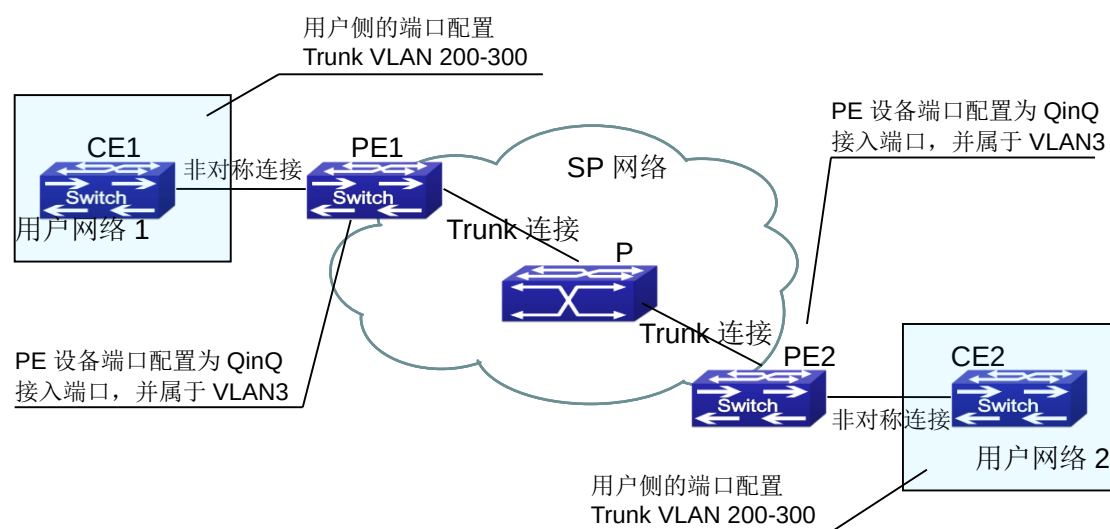


图 2-29 基于 Dot1q-tunnel 的互连模式

如上图所示, 在用户端口上使能 dot1q-tunnel 功能, 并为每个用户分配一个 SPVLAN 号 (SPVID), 此处为 3, 不同的 PE 上应该为同一网络用户分配相同的 SPVID。当报文从

CE1 到达 PE1 时，带用户内部网络的 VLAN tag 200~300，由于使能了 dot1q-tunnel 功能，PE1 上的用户端口将再次为报文加上另外一层 VLAN tag，其 ID 就是分配给该用户的 SPVID。此后该报文在服务提供商网络中传播时仅在 VLAN3 中进行且全程带有两层 VLAN tag（内层为进入 PE1 时的 tag，外层为 SPVID），但用户网络的 VLAN 信息对运营商网络来说是透明的。当报文到达 PE2，从 PE2 上的客户端口转发给 CE2 之前，外层 VLAN tag 被剥去，CE2 收到的报文内容与 CE1 发送的报文完全相同。PE1 到 PE2 之间的运营商网络对于用户来说，其作用就是提供了一条可靠的二层链路。

Dot1q-tunnel 技术的产生使得服务提供商可以靠自己的一个 VLAN 来支持客户的多个 VLAN。服务提供商和客户可以独立设置自己的 VLAN。

可见，使用 dot1q-tunnel 功能具有如下特点：

- 通过简单的静态配置即可实现，免去了繁杂的配置，维护工作。
- 运营商只需为每个用户分配一个 SPVID，提升了可以同时支持的用户数目；而用户也具有选择和管理 VLAN ID 资源的最大自由度（从 1~4096 中任意选择）。
- 用户网络具有较高的独立性，在服务提供商升级网络时，用户网络不必更改原有的配置。

本节将对 switch 的 dot1q-tunnel 使用和配置进行详细说明。

2.16.2 Dot1q-tunnel 配置

Dot1q-tunnel 的配置任务序列：

1. 设置端口的 dot1q-tunnel 功能
2. 设置端口的协议类型（TPID）

1. 设置端口的 dot1q-tunnel 功能

命令	解释
端口配置模式	
dot1q-tunnel enable no dot1q-tunnel enable	进入/退出端口的 dot1q-tunnel 模式。

2. 设置端口的协议类型（TPID）

命令	解释
端口配置模式	
dot1q-tunnel tpid {0x8100 0x9100 0x9200 <1-65535>}	在 trunk 端口上设置端口的协议类型。

2.16.3 Dot1q-tunnel 典型应用

案例：

服务提供商边界交换机 PE1 与 PE2 用 VLAN3 来承载客户网络 CE1 与 CE2 的 VLAN200~300 的数据业务。PE1 的端口 1 连接 CE1，端口 10 连接公网，所连设备的 TPID 为 9100；PE2 的端口 1 连接 CE2，端口 10 连接公网。

配置项目	配置说明
VLAN3	PE1、PE2 的端口 1
dot1q-tunnel	PE1、PE2 的端口 1
tpid	9100

配置步骤如下：

PE1:

```
Switch(Config)#vlan 3
Switch(Config-Vlan3)#switchport interface ethernet 1/1
Switch(Config-Vlan3)#exit
Switch(Config)#interface ethernet 1/1
Switch(Config-Ethernet1/1)# dot1q-tunnel enable
Switch(Config-Ethernet1/1)# exit
Switch(Config)#interface ethernet 1/10
Switch(Config-Ethernet1/10)#switchport mode trunk
Switch(Config-Ethernet1/10)#dot1q-tunnel tpid 0x9100
Switch(Config-Ethernet1/10)#exit
Switch(Config)#
```

PE2:

```
Switch(Config)#vlan 3
Switch(Config-Vlan3)#switchport interface ethernet 1/1
Switch(Config-Vlan3)#exit
Switch(Config)#interface ethernet 1/1
Switch(Config-Ethernet1/1)# dot1q-tunnel enable
Switch(Config-Ethernet1/1)# exit
Switch(Config)#interface ethernet 1/10
Switch(Config-Ethernet1/10)#switchport mode trunk
Switch(Config-Ethernet1/10)#dot1q-tunnel tpid 0x9100
Switch(Config-Ethernet1/10)#exit
Switch(Config)#
```


2.16.4 Dot1q-tunnel 排错帮助

- ☞ 在 Trunk 端口启用 dot1q-tunnel 功能会使数据包的标签层数据不可预知，也不是应用中所需要的，因而不推荐在 Trunk 端口启用 dot1q-tunnel 功能。
- ☞ 不能与 STP/MSTP 同时使用。
- ☞ 不能与 PVLAN 同时使用。

2.17 VLAN-translation

2.17.1 VLAN-translation 介绍

VLAN translation 即 VLAN 翻译，它根据用户的需求将原有的 VLAN ID 转换为新的 VLAN ID，这样可以在不同的 VLAN 之间交换数据。VLAN translation 分为入口翻译与出口翻译，分别在入口或出口对 VLAN ID 进行转换。

本节将对 VLAN translation 的使用和配置进行详细的说明。

2.17.2 VLAN-translation 配置

VLAN-translation 的配置任务序列：

1. 设置端口的 VLAN-translation 功能
2. 设置端口 VLAN-translation 的翻译关系
3. 设置端口 VLAN-translation 翻译关系查找失败是否丢包
4. 显示 VLAN-translation 相关配置

1. 设置端口的 VLAN-translation 功能

命令	解释
端口配置模式	
vlan-translation enable no vlan-translation enable	进入/退出端口的 VLAN-translation 模式。

2. 设置端口 VLAN-translation 的翻译关系

命令	解释
端口配置模式	
vlan-translation <old-vlan-id> to <new-vlan-id> {in out} no vlan-translation <old-vlan-id> {in out}	添加/删除 VLAN-translation 翻译关系。

3. 设置端口 VLAN-translation 翻译关系查找失败是否丢包

命令	解释
端口配置模式	
vlan-translation miss drop {in out both} no vlan-translation miss drop {in out both}	设置/取消 VLAN-translation 查找翻译关系失败时丢包。

4. 显示 VLAN-translation 相关配置

命令	解释
特权用户配置模式	
show vlan-translation	显示 vlan-translation 相关配置。

2.17.3 VLAN-translation 典型应用

案例：

服务提供商边界交换机 PE1 与 PE2 用 VLAN3 来承载客户网络 CE1 与 CE2 的 VLAN20 的数据业务。PE1 的端口 1/1 连接 CE1，端口 1/10 连接公网；PE2 的端口 1/1 连接 CE2，端口 1/10 连接公网（如下图）。

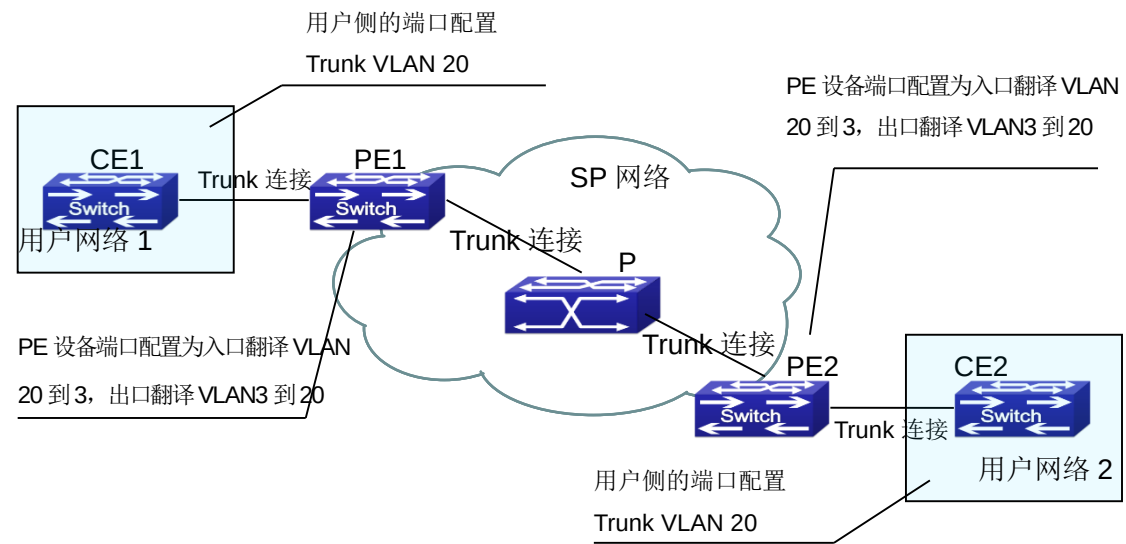


图 2-30 VLAN-translation 典型应用拓扑

配置项目	配置说明
VLAN-translation	PE1、PE2 的端口 1/1
Trunk port	PE1、PE2 的端口 1/1 与端口 1/10

配置步骤如下：

PE1、PE2:

```
switch(Config)#interface ethernet 1/1
switch(Config-Ethernet1/1)#switchport mode trunk
switch(Config-Ethernet1/1)# vlan-translation enable
switch(Config-Ethernet1/1)# vlan-translation 20 to 3 in
switch(Config-Ethernet1/1)# vlan-translation 3 to 20 out
switch(Config-Ethernet1/1)# exit
switch(Config)#interface ethernet 1/10
switch(Config-Ethernet1/10)#switchport mode trunk
switch(Config-Ethernet1/10)#exit
switch(Config)#
```

2.17.4 VLAN-translation 排错帮助

不能与 Dot1q-tunnel 同时使用。

目前部分产品不支持对非 0x8100 的报文进行 vlan translation，所以不建议在 vlan-translation 端口配置 dot1q-tunnel tpid 为非 0x8100 的值。

灵活 QINQ、vlan translation 以及 QINQ 对报文处理的优先级关系如下：灵活 QINQ > vlan translation > QINQ。

如果入口翻译不匹配，部分产品可能会出现对入口 tag 报文在外层再打一层 tag 的行为，所以当确定入口报文不匹配入口翻译表项时请取消 vlan-translation enable。

2.18 动态 VLAN

2.18.1 动态 VLAN 介绍

动态 VLAN 是相对于静态 VLAN（即基于端口的 VLAN）而言的。动态 VLAN 包括基于 MAC 地址的 VLAN（MAC-based VLAN）、基于 IP 子网的 VLAN（IP-subnet-based VLAN）和基于协议的 VLAN（Protocol-based VLAN），分述如下。

基于 MAC 地址划分 VLAN 的方法是根据每个主机的 MAC 地址来划分，即对每个 MAC 地址的主机都配置他属于哪个 VLAN。这种方式的 VLAN 允许网络用户从一个物理位置移动到另一个物理位置时，自动保留其所属 VLAN 的成员身份。由这种划分的机制可以看出，这种 VLAN 的划分方法的最大优点就是当用户物理位置移动时，即从一个交换机换到其他的交换机时，VLAN 不用重新配置，因为它是基于用户，而不是基于交换机的端口。

基于 IP 子网的 VLAN 是根据每个主机的源 IP 地址及其子网掩码来划分的 VLAN，它根据子网网段来为进入的数据包分配相应的 VLAN 号，使数据包进入指定的 VLAN。它的优点

与基于 MAC 地址划分的 VLAN 相同，可以在用户移动时不必重新进行配置。

VLAN 按网络层协议来划分，可使不同的协议属于不同的 VLAN。这对于希望针对具体应用和服务来组织用户的网络管理员来说是非常具有吸引力的。而且，用户可以在网络内部自由移动，但其 VLAN 成员身份仍然保留不变。这种方法的优点是用户的物理位置改变了，不需要重新配置所属的 VLAN，而且可以根据协议类型来划分 VLAN，这对网络管理者来说很重要。还有，这种方法不需要附加的帧标签来识别 VLAN，这样可以减少网络的通信量。

注意：动态 VLAN 需要跟端口的 Hybrid 属性配合才能很好的工作，需要用户将需要添加到动态 VLAN 的端口配置为 Hybrid 属性。

2.18.2 动态 VLAN 配置

动态 VLAN 的配置任务序列：

1. 设置端口的 MAC-based VLAN 功能
2. 指定一个 VLAN 为 MAC VLAN
3. 设置 MAC 地址与 VLAN 的对应关系
4. 设置端口的 IP-subnet-based VLAN 功能
5. 设置 IP 子网与 VLAN 的对应关系
6. 设置协议与 VLAN 的对应关系
7. 调整动态 VLAN 的优先顺序

1. 设置端口的 MAC-based VLAN 功能

命令	解释
端口配置模式	
switchport mac-vlan enable no switchport mac-vlan enable	打开/关闭端口的 MAC-based VLAN 功能。

2. 指定一个 VLAN 为 MAC VLAN

命令	解释
全局配置模式	
mac-vlan vlan <vlan-id> no mac-vlan vlan <vlan-id>	设置指定 VLAN 为 MAC VLAN；本命令的 no 命令为取消该 VLAN 为 MAC VLAN。

3. 设置 MAC 地址与 VLAN 的对应关系

命令	解释
全局配置模式	

mac-vlan mac <mac-addrss> <mac-mask> vlan <vlan-id> priority <priority-id> no mac-vlan {mac <mac-addrss> <mac-mask> all}	添加/删除 MAC 地址与 VLAN 的对应关系。即指定 MAC 地址加入/离开指定 VLAN。

4. 设置端口的 IP-subnet-based VLAN 功能

命令	解释
端口配置模式	
switchport subnet-vlan enable no switchport subnet-vlan enable	打开/关闭端口的 IP-subnet-based VLAN 功能。

5. 设置 IP 子网与 VLAN 的对应关系

命令	解释
全局配置模式	
subnet-vlan ip-address <ipv4-addrss> mask <subnet-mask> vlan <vlan-id> priority <priority-id> no subnet-vlan {ip-address <ipv4-addrss> mask <subnet-mask> all}	添加/删除 IP 子网与 VLAN 的对应关系。即指定 IP 子网加入/离开指定 VLAN。

6. 设置协议与 VLAN 的对应关系

命令	解释
全局配置模式	
protocol-vlan mode {ethernetii etype <etype-id> llc {dsap <dsap-id> ssap <ssap-id>}} snap etype <etype-id>} vlan <vlan-id> priority <priority-id> no protocol-vlan {mode {ethernetii etype <etype-id> llc {dsap <dsap-id> ssap <ssap-id>}} snap etype <etype-id>}} all}	添加/删除协议与 VLAN 的对应关系。即指定协议加入/离开指定 VLAN。

7. 调整动态 VLAN 的优先顺序

命令	解释
全局配置模式	

dynamic-vlan mac-vlan prefer dynamic-vlan subnet-vlan prefer	设置动态 VLAN 的优先顺序。

2.18.3 动态 VLAN 典型应用

案例：

办公网中某部门 A 属于 VLAN100，该部门有若干成员经常在整个办公网中移动办公，而其仍要保证是部门 A 的成员而访问 VLAN100 的资源。假定该其中一个成员为 M，其 PC 的 MAC 地址为 00-03-0f-11-22-33，当 M 移动到 VLAN200 或 VLAN300 中时，应当将 M 所在的端口配置为 hybrid 模式，并且以 untag 方式属于 VLAN100。这样当 VLAN100 中的数据会转发到 M 所在的端口，实现在 VLAN100 中的通信需求。

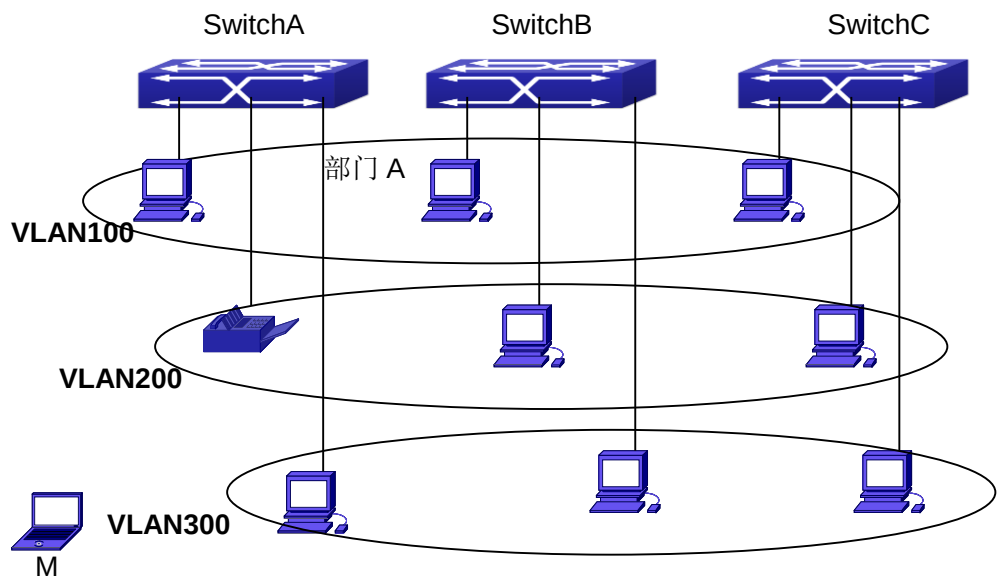


图 2-31 动态 VLAN 典型应用拓扑

配置项目	配置说明
MAC-based VLAN	Switch A, Switch B, Switch C 进行全局配置

例如 M 位于 Switch A 的 E1/1，则配置步骤如下：

对于 Switch A, Switch B, Switch C:

```
SwitchA(Config)#mac-vlan mac 00-03-0f-11-22-33 vlan 100 priority 0
SwitchA(Config)#interface ethernet 1/1
SwitchA(Config-Ethernet1/1)#swportport mode hybrid
SwitchA(Config-Ethernet1/1)#swportport hybrid allowed vlan 100 untagged
```

```
SwitchB(Config)#mac-vlan mac 00-03-0f-11-22-33 vlan 100 priority 0
SwitchB(Config)#exit
SwitchB#
```

```
SwitchC(Config)#mac-vlan mac 00-03-0f-11-22-33 vlan 100 priority 0
SwitchC(Config)#exit
SwitchC#
```

2.18.4 动态 VLAN 排错帮助

在配置有动态 VLAN 的交换机上,如果连接的两台设备(如 PC)均属于同一动态 VLAN,那么这两台设备之间初次通讯会有不通的现象。解决方法是两台设备主动向交换机发送数据包(如进行 ping 操作),使交换机学习到它们的源 MAC,之后,两台设备在该动态 VLAN 之中可正常通讯。如下图所示。

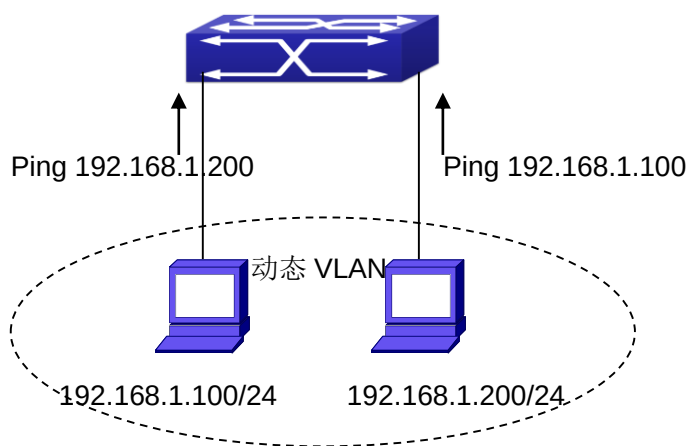


图 2-32 动态 vlan 排错示例

2.19 Voice VLAN

2.19.1 Voice VLAN 介绍

Voice VLAN 指为用户的语音数据流而专门划分的 VLAN。通过划分 Voice VLAN 并将连接语音设备的端口加入 Voice VLAN,可以为语音数据配置 QoS 服务(Quality of Service, 服务质量),提高语音流量的传输优先级,保证通话质量。

交换机可以根据进入端口的数据报文中的源 MAC 地址字段,判断该数据流是否为指定语音设备的语音数据流,源 MAC 地址符合系统设置的语音设备 OUI (Organizationally Unique Identifier, 全球统一标识符)地址的报文被认为是语音数据流,被划分到 Voice VLAN 中传输。

这种划分是基于 MAC 地址来划分的，它实现的机制就是每一个通过网络传输信息的语音设备都对应唯一的 MAC 地址，VLAN 交换机跟踪属于指定的 MAC 的地址。这种方式的 VLAN 允许语音设备从一个物理位置移动到另一个物理位置时，仍然属于 Voice VLAN。由这种划分的机制可以看出，这种 VLAN 的划分方法的最大优点就是根据语音流而自动将其划入 Voice VLAN，在特定优先级下传输语音数据。同时，当语音设备物理位置移动时，同样属于 Voice VLAN，而不用重新配置，因为它是基于语音设备的，而不是基于交换机的端口。

注意：Voice VLAN需要跟端口的Hybrid属性配合才能很好的工作，需要用户将需要添加到Voice VLAN的端口配置为Hybrid属性。

2.19.2 Voice VLAN 配置

Voice VLAN 的配置任务序列如下：

1. 设置 VLAN 为 Voice VLAN
2. 将语音设备加入 Voice VLAN
3. 使能端口的 Voice VLAN

1. 设置 VLAN 为 Voice VLAN

命令	解释
全局配置模式	
voice-vlan vlan <vlan-id> no voice-vlan	设置/取消指定 VLAN 为 Voice VLAN。

2. 将语音设备加入 Voice VLAN

命令	解释
全局配置模式	
voice-vlan mac <mac-address> <mac-address-mask> priority <priority-id> [name <voice-name>] no voice-vlan {mac <mac-address> <mac-address-mask> name <voice-name> all}	指定某语音设备加入/离开 Voice VLAN。

3. 使能端口的 Voice VLAN

命令	解释
端口配置模式	

switchport voice-vlan enable no switchport voice-vlan enable	打开/关闭端口的 Voice VLAN 功能。

2.19.3 Voice VLAN 典型应用

案例：

公司内部通过 Voice VLAN 的设置来实现通讯。IP-phone1 与 IP-phone2 可以接在交换机的任意端口上，即可实现正常通讯，并且可以通过上联端口与其它交换机进行互通。
IP-phone1 的 MAC 地址为 00-03-0f-11-22-33，"连接交换机的 1/1 口， IP-phone2 的 MAC 地址为 00-03-0f-11-22-55，连接交换机的 1/2 口，交换机的端口 1/10 为上联端口。如下图所示。

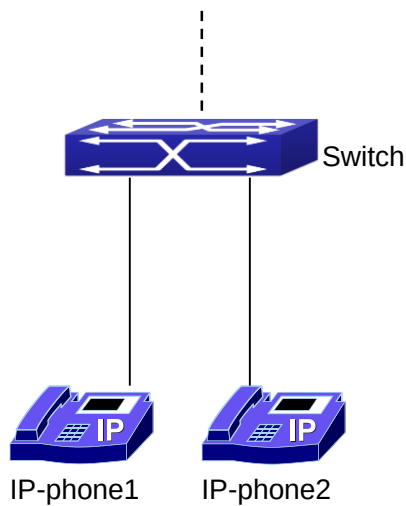


图 2-33 Voice VLAN 典型应用拓扑

配置项目	配置说明
Voice VLAN	Switch 进行全局配置

配置步骤如下：

Switch 1:

```
switch(Config)#vlan 100
switch(Config-Vlan100)#exit
switch(Config)#voice-vlan vlan 100
switch(Config)#voice-vlan mac 00-03-0f-11-22-33 mask 255 priority 5 name company
switch(Config)#voice-vlan mac 00-03-0f-11-22-55 mask 255 priority 5 name company
switch(Config)#interface ethernet1/10
switch(Config-If-Ethernet1/10)#switchport mode trunk
switch(Config-If-Ethernet1/10)#exit
```

```
switch(Config)#interface ethernet 1/1
switch(Config-If-Ethernet1/1)#switchport mode hybrid
switch(Config-If-Ethernet1/1)#switchport hybrid allowed vlan 100 untag
switch(Config-If-Ethernet1/1)#exit
switch(Config)#interface ethernet 1/2
switch(Config-If-Ethernet1/2)#switchport mode hybrid
switch(Config-If-Ethernet1/2)#switchport hybrid allowed vlan 100 untag
switch(Config-If-Ethernet1/2)#exit
```

2.19.4 Voice VLAN 排错帮助

Voice VLAN 与 MAC-based VLAN 不能同时使用。Voice VLAN 对于语音设备的支持最多为 1024 台，超出上限将不再支持。

- ☞ 端口上的 Voice VLAN 缺省为打开，若使用中某端口上已经设置好的数据不再进入 Voice VLAN，请首先考虑是否在端口上已将 Voice VLAN 功能关闭。

2.20 Super VLAN

2.20.1 Super VLAN 简介

Super VLAN 技术（也称 VLAN 聚合）引入 super VLAN 和 sub VLAN 的概念。一个 super VLAN 可以包含一个或多个保持着不同广播域的 sub VLAN。Sub VLAN 不再占用一个独立的子网网段。在同一个 super VLAN 中，无论主机属于哪一个 sub VLAN，它的 IP 地址都在 super VLAN 对应的子网网段内。

在 LanSwitch 网络中，VLAN 技术以其对广播域的灵活控制（可跨物理设备）、部署方便而得到了广泛的应用。但是在一般的三层交换机中，通常是采用一个 VLAN 对应一个三层接口的方式来实现广播域之间的互通的，这在某些情况下导致了对 IP 地址的较大浪费。我们来看下面这个例子，设备内 VLAN 划分如下图所示。

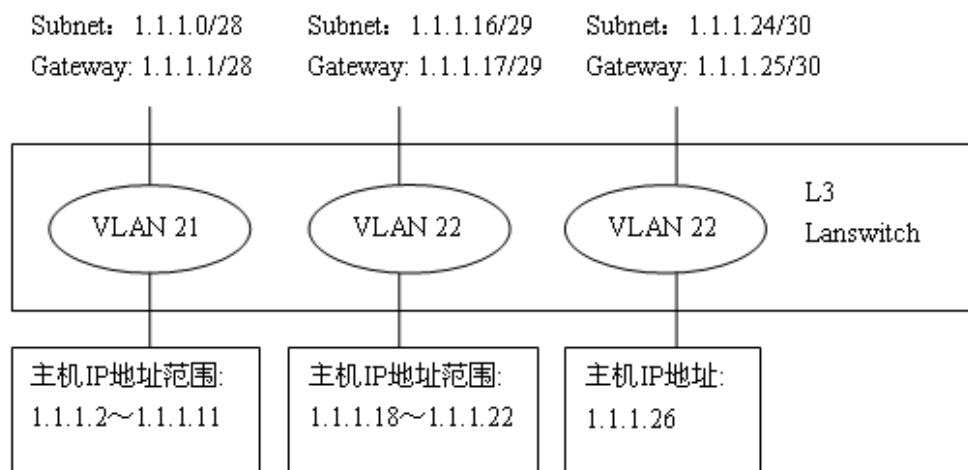


图 2-34 普通 VLAN 网络示意图

VLAN	IP Subnet	Gateway Address	Usable Hosts	Customer Hosts	Needed Hosts
21	1.1.1.0/28	1.1.1.1	14	13	10
22	1.1.1.16/29	1.1.1.17	6	5	5
23	1.1.1.24/30	1.1.1.25	2	1	1

图 2-35 普通 VLAN 主机地址划分示例

图 2-35 列出了地址的划分情况。VLAN 21 预计未来要有 10 个主机，给其分配一个掩码长 28 的子网——1.1.1.0/28。网段内子网号 1.1.1.0 和子网定向广播地址 1.1.1.15 不能用作主机地址，1.1.1.1 作为子网缺省网关地址也不可作为主机地址，剩下地址范围在 1.1.1.2~1.1.1.14 的主机地址可以被主机使用，这部分地址共有 13 个（ $2^{32-28}-3=13$ ）。这样，尽管 VLAN 21 只需要 10 个地址就可以满足需求了，但是按照子网划分却要分给它 13 个地址。

依此类推，VLAN 22 预计未来有 5 个主机地址需求，至少需要分配一个 29 位掩码的子网（1.1.1.16/29）才能满足要求。VLAN 23 只有 1 个主机，则占用子网 1.1.1.24/30。

这三个 VLAN 总共需要 $10+5+1=16$ 个地址，按照普通 VLAN 的编址方式，即使最优化的方案也需要占用 $2^{32-28}+2^{32-29}+2^{32-30}=28$ 个地址，地址浪费了将近一半。而且，如果 VLAN 21 后来并没有增长到 10 台主机，而只增长到了 3 台主机，本来分一个掩码长 29 的子网就够用了，但之前却分了一个子网掩码长 28 的子网给它，多出来的地址都因不能再被其他 VLAN 使用而被浪费掉了。

同时，这种划分也给后续的网络升级带来了很大不便。假设 VLAN23 所在的客户一段时间后需要增加 2 台主机，又不愿意改变已经分配的 IP 地址。在 1.1.1.24 后面的地址已经分配给了其他人的情况下，那没有办法，只能额外再给他分配一个的 29 位掩码的子网和一个新的 VLAN 给他。这样 VLAN23 的客户只有 3 台主机，却被迫分配了两个子网，不在同

一个 VLAN，造成管理上的极大不便。

由此我们可以看出，被诸如子网号、子网定向广播地址、子网缺省网关地址消耗掉的 IP 地址数量是相当可观。同时，这种地址分配的固有约束也严重降低了编址的灵活性，使许多闲置地址被浪费。为了解决这一问题，Super VLAN 应运而生。

可见，Super VLAN 的好处是：减少子网号、子网缺省网关地址和子网定向广播地址的消耗，实现了不同广播域使用同一子网网段地址，增加编址的灵活性，减少了闲置地址浪费。

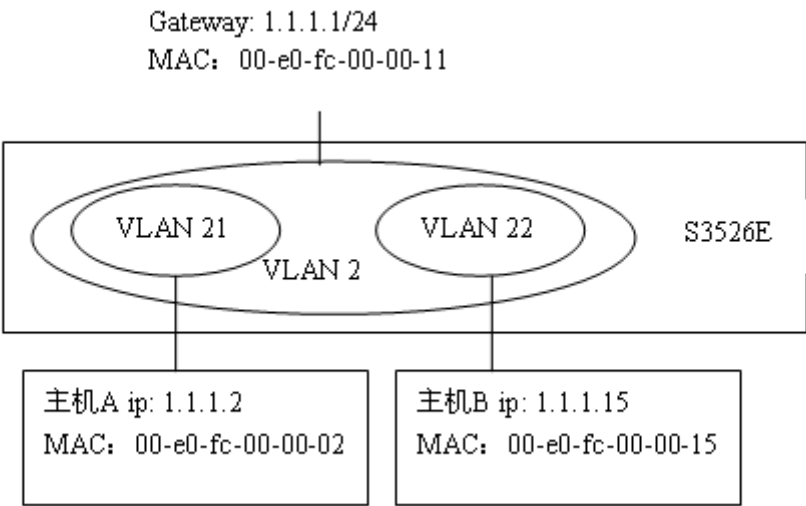


图 2-36 super vlan 功能示意图

- Super VLAN 不同于通常意义的 VLAN，Super VLAN 只是建立一个三层接口，不能包含任何端口，是一个三层的概念。
- Super VLAN 三层接口 UP：它所含 sub-VLAN 中存在 UP 状态的物理端口。

2.20.2 Super VLAN 配置任务序列

1. 创建或删除 supervlan
2. 指定或删除 subvlan
3. 开启或关闭 subvlan 的 arp-proxy 功能
4. 指定或删除接口的 ip-addr-range
5. 指定或删除 subvlan 的地址范围

1. 创建或删除 supervlan

命令	解释
VLAN 配置模式	
supervlan	创建/删除 supervlan。
no supervlan	

2. 指定或删除 subvlan

命令	解释

VLAN 配置模式	
subvlan WORD no subvlan {WORD all}	指定/删除 subvlan。

3. 开启或关闭 subvlan 的 arp-proxy 功能

命令	解释
接口配置模式	
arp-proxy subvlan {WORD all} no arp-proxy subvlan {WORD all}	开启/关闭 subvlan 的 arp-proxy 功能。

4. 指定或删除接口的 ip-addr-range

命令	解释
接口配置模式	
ip-addr-range <ipv4-addrss> to <ipv4-addrss> no ip-addr-range	指定/删除接口的地址范围。

5. 指定或删除 subvlan 的地址范围

命令	解释
接口配置模式	
arp-check-range subvlan <vlan-id> <ipv4-addrss> to <ipv4-addrss> no ip-addr-range subvlan <vlan-id>	指定/删除 subvlan 的地址范围。

2.20.3 Super VLAN 典型应用

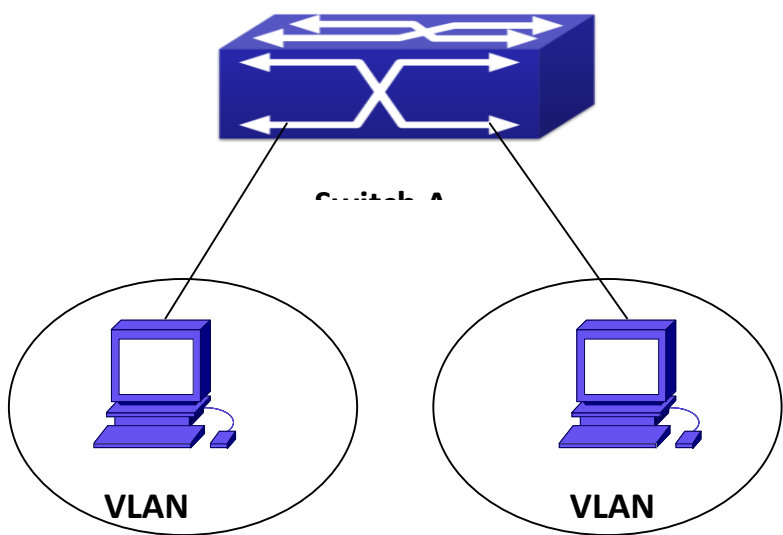


图 2-37 supervlan 典型应用拓扑图

由于局域网应用的需要，需要让两个 vlan 的终端配置为同一网段的地址，它们之间的二层流量要隔离，但是三层流量能正常转发。vlan3 用的地址范围为 1.1.1.1-10，vlan4 用的地址范围为 1.1.1.20-30，只有在这个地址范围内的终端的三层流量互通，而其他使用同网段地址的终端的三层流量不被转发。在交换机上设置 supervlan 可以实现这种需求。

配置项目	配置说明
VLAN2	Supervlan
VLAN3	A 交换机 1 端口
VLAN4	A 交换机 2 端口

配置步骤如下：

```
Switch A:
switch(Config)#vlan 2-4
switch(Config)#vlan 2
switch(Config-Vlan2)#supervlan
switch(Config-Vlan2)#subvlan 3;4
switch(Config-Vlan2)#exit
switch(Config)#interface vlan 2
switch(config-if-vlan2)#ip address 1.1.1.254 255.255.255.0
switch(config-if-vlan2)# arp-check-range subvlan 3 1.1.1.1 to 1.1.1.10
switch(config-if-vlan2)# arp-check-range subvlan 4 1.1.1.20 to 1.1.1.30
switch(config-if-vlan2)#arp-proxy subvlan all
switch(config-if-vlan2)#exit
```

2.20.4 Super VLAN 排错帮助

☞ supervlan 功能与 VRRP、动态 VLAN、私有 VLAN、组播 VLAN 等功能互斥，不要同

时使用。

- ☞ subvlan 的 arp-proxy 功能只对一个 subvlan 生效，配置 arp-proxy 的 VLAN 收到的流量可以向其他 VLAN 转发。当处于两个不同 subvlan 的设备互相发送流量通信时，请注意将两个 subvlan 的 arp-proxy 功能均开启。
- ☞ subvlan 上不能建立三层接口。
- ☞ 创建\删除 supervlan 时需要保证该 VLAN 没有三层接口，否则会报错。
- ☞ 当 supervlan 的接口上指定了接口的 IP 地址范围，未指定 subvlan 地址范围，则以接口上规定的地址范围为准；当接口和 subvlan 上都指定了 IP 地址范围时，按顺序先检查是否在 subvlan 地址范围内，然后再检查是否在接口地址范围内，通过上述两次检查后的数据包才可以进入后续处理环节。
- ☞ 设置 supervlan 或 subvlan 时必须是在存在的 VLAN 才可以设置。

将端口模式设置为 trunk 时，会自动将 supervlan 从 allow-vlan 中去除。

2.21 MAC 地址表

2.21.1 MAC 地址表介绍

MAC 地址表是标识目的 MAC 地址与交换机端口之间映射关系的表，其 MAC 地址分为静态 MAC 地址和动态 MAC 地址。静态 MAC 地址由用户配置，具有最高优先级（不能被动态 MAC 地址覆盖）且永久生效；动态 MAC 地址由交换机在转发数据帧的过程中学习，且在有限时间内生效。当交换机接收到需要转发的数据帧时，首先学习数据帧的源 MAC 地址，与接收端口建立映射关系；然后根据目标 MAC 地址查询 MAC 地址表，如果命中相关表项，交换机将数据帧从相应端口转发；否则，交换机将数据帧在其所属广播域内广播。如果动态 MAC 地址长时间没有从转发数据帧中学习到，交换机就将其从 MAC 地址表中删除。

对于 MAC 地址表的操作可分为两步：

1. MAC 地址的获取；
2. 根据 MAC 地址表转发或过滤。

2.21.1.1 MAC 地址表的获取

MAC 地址表的获取可分为静态配置和动态学习。静态配置即由用户人为的建立 MAC 地址与端口的映射关系；动态学习即由交换机动态的发现 MAC 地址与端口的映射关系，并定期的更新 MAC 地址表。下面我们将重点介绍 MAC 地址表的动态学习过程。

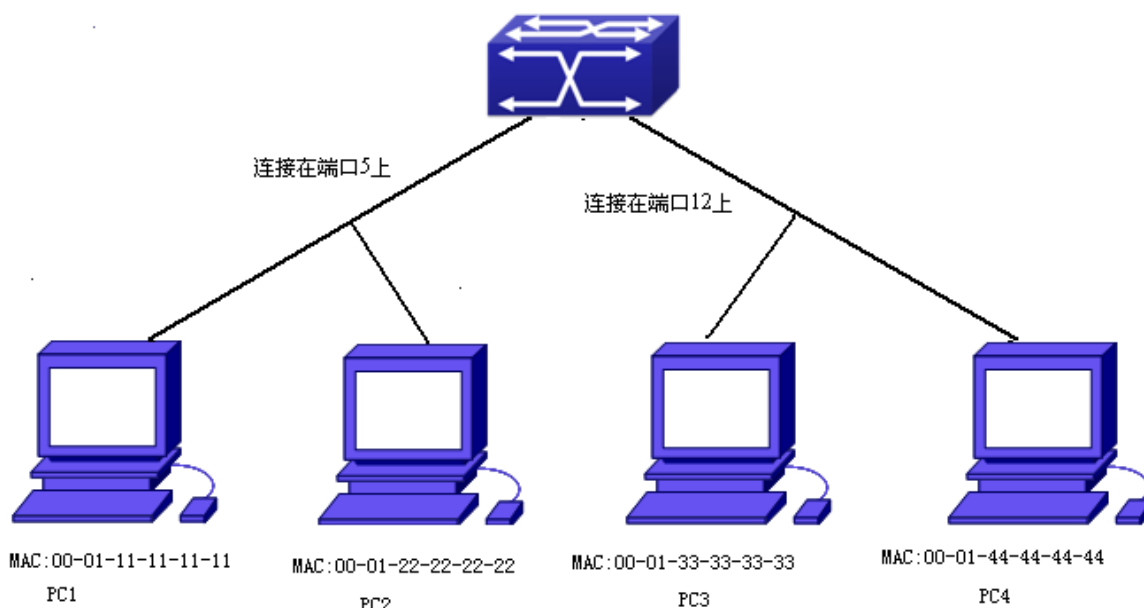


图 2-38 MAC 地址表动态学习

上图的拓扑环境为：4 台主机连接在交换机上，其中主机 1、2 在同一个物理分段中（即相同的冲突域），该物理分段与交换机的端口 1/5 相连；主机 3、4 在同一个物理分段，该物理分段与交换机的端口 1/12 相连。

初始状态下MAC地址表中没有任何学习到的地址映射表项，以主机1和主机3的相互通信为例，MAC地址表的学习过程如下：

1. 当主机1向主机3传输信息时，交换机在端口1/5处收到该信息的源MAC地址00-01-11-11-11-11，交换机的MAC地址表中就会增加MAC地址00-01-11-11-11-11和端口1/5映射表项；
2. 同时交换机会检查到该信息的目标MAC地址00-01-33-33-33-33，此时交换机中只有MAC地址00-01-11-11-11-11和端口1/5的映射表项，没有00-01-33-33-33-33对应的端口映射，因此交换机只能将该信息广播给交换机的每个端口（假设交换机的所有端口都属于缺省VLAN）；
3. 位于端口1/12的主机3、4均收到主机1发出的信息，但主机4不会给主机1回应，因为目标MAC地址为00-01-33-33-33-33，只有主机3会给主机1回应。这时交换机的1/12号端口收到主机3的发出的信息，交换机的MAC地址表中就又增加了MAC地址00-01-33-33-33-33和端口1/12映射表项；
4. 目前MAC地址表的内容为MAC地址00-01-11-11-11-11动态对应着端口1/5，MAC地址00-01-33-33-33-33动态对应着端口1/12；
5. 经过一段时间的主机1和主机3的通信之后，交换机再也没有接收到从主机1和主机3发出的信息，300到2*300秒（即一到两倍的老化时间内）内交换机的MAC地址表将删除上面保存的MAC地址映射表项。这里的300秒是交换机缺省的MAC地址的老化时间，交换机提供老化时间的修改。

2.21.1.2 转发或过滤

交换机会根据MAC地址表对接收到的数据帧做出转发或过滤的决定。以上图为例，假设当前交换机 MAC地址表动态学习到了主机1和主机3的MAC地址，用户又手工配置了主机2和主机4与端口的映射关系。交换机的MAC地址表为：

MAC地址	端口号	获取方式
00-01-11-11-11-11	1/5	动态
00-01-22-22-22-22	1/5	静态
00-01-33-33-33-33	1/12	动态
00-01-44-44-44-44	1/12	静态

1. 根据MAC地址表转发的情况

如果主机1向主机3发送信息时，交换机根据MAC地址表，将从端口1/5接收到的数据从端口1/12发出。

2. 根据MAC地址表过滤的情况

如果主机1向主机2发送信息，交换机根据MAC地址表，检查到主机2和主机1在同一个物理分段中，交换机将对该信息进行过滤，即不发送帧信息。

另外交换机能转发三种类型的帧：

- ☞ 广播帧；
- ☞ 组播帧；
- ☞ 单播帧。

下面简单介绍交换机对三种帧的处理：

1. 广播帧：交换机能阻隔冲突域，但不能阻隔广播域，在没有设置VLAN的情况下连接在交换机的所有设备是处在同一个广播域中的，当交换机接收到广播帧时，它会向所有的端口转发该广播帧。当交换机设置了VLAN后，MAC地址表也会做相应的调整，会增加VLAN的信息，此时交换机接收到广播帧后，不会将该广播帧转发给交换机内的所有端口，而改变为只向属于同一个VLAN的所有端口转发。
2. 组播帧：对于未知组播，交换机会在同一个vlan内广播，如果交换机设置了IGMP Snooping功能或配置了静态组播组时，交换机只会向属于该组播组的端口转发该组播帧。
3. 单播帧：在没有设置VLAN的情况下，当交换机接收到的单播帧的目标MAC地址在MAC表中存在，交换机会直接将该单播帧转发到相应的端口；当接收到单播帧的目标MAC地址在MAC地址表中不存在时，交换机会对该单播帧进行广播。当交换机设置了VLAN，交换机只会同一VLAN内转发单播帧；当转发单播帧的目标MAC地址在MAC地址表中存在，但不属于同一VLAN，此时交换机只能将该单播帧在它属于的VLAN内进行广播。

2.21.2 MAC 地址表配置

Mac 地址表的配置任务序列如下：

1. 配置 mac 地址老化时间，

2. 配置静态 mac 地址转发/过滤表项
3. 删除动态地址表项
4. 配置 MAC 地址的学习功能
5. 配置 hash 冲突表

1. 配置 MAC 地址老化时间

命令	解释
全局配置模式	
mac-address-table aging-time <0 aging-time> no mac-address-table aging-time	配置 mac 地址老化时间。

2. 配置静态 mac 地址转发/过滤表项

命令	解释
全局配置模式	
mac-address-table {static static-multicast blackhole} address <mac-addr> vlan <vlan-id> [interface [ethernet portchannel] <interface-name>] [source destination both] no mac-address-table {static static-multicast blackhole dynamic} [address <mac-addr>] [vlan <vlan-id>] [interface [ethernet portchannel] <interface-name>]	配置静态 mac 地址转发表项，配置静态组播 MAC 地址表项，配置地址过滤表项。

3. 删除动态地址表项

命令	解释
特权用户配置模式	
clear mac-address-table dynamic [address <mac-addr>] [vlan <vlan-id>] [interface [ethernet portchannel] <interface-name>]	删除动态地址表项。

4. 配置 MAC 软学习

命令	解释

全局配置模式	
mac-address-learning cpu-control	Disable 是禁止端口 mac 地址学习功能；
no mac-address-learning cpu-control	enable 命令恢复端口 mac 地址学习功能。

5. 配置 hash 冲突表

命令	解释
全局配置模式	
show collision-mac-address-table	打印 hash 冲突 mac 表。
特权用户配置模式	
clear collision-mac-address-table	清空 hash 冲突表。

2.21.3 典型配置举例

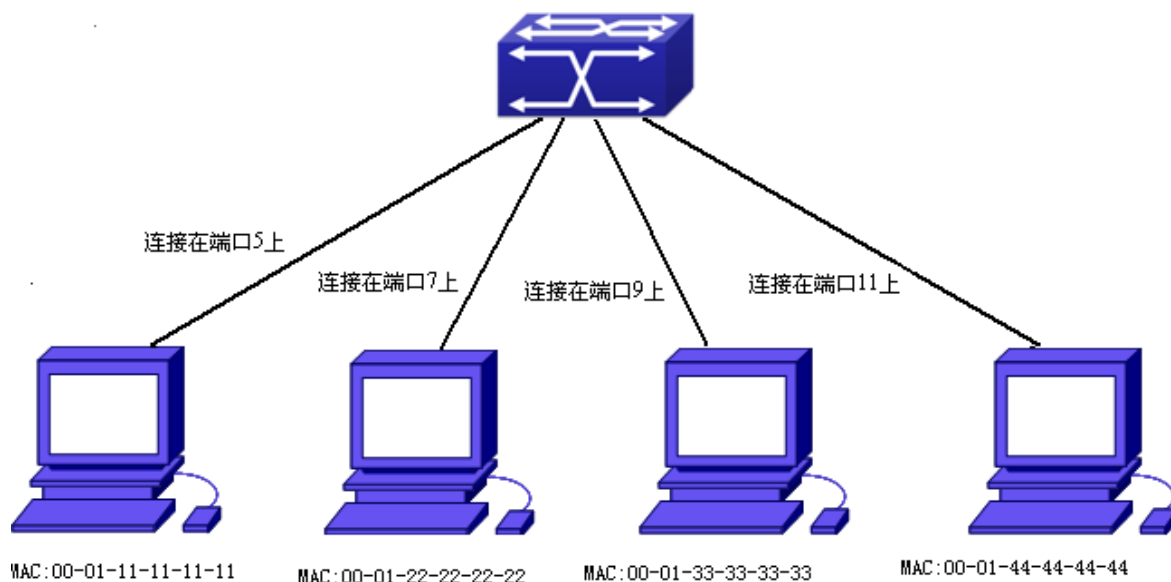


图 2-39 MAC 地址表典型配置举例

案例：

如上图所示四台主机分别连接在交换机的 1/5、1/7、1/9、1/11 端口，这四台主机都属于缺省 VLAN1。根据实际网络的需要，动态学习功能是打开的；主机 1（连接在端口 1/5）保存有机密资料，任何与它不在一个物理分段的主机不能访问它；主机 2（连接在端口 1/7）和主机 3（连接在端口 1/9）分别与端口 1/7 和端口 1/9 建立静态映射关系。

配置步骤如下：

1. 设置主机 1 的 MAC 地址 00-01-11-11-11-11 为过滤地址；

Switch(config)#mac-address-table blackhole address 00-01-11-11-11-11 vlan 1

2. 主机 2 和主机 3 分别与端口 1/7 和端口 1/9 建立静态映射关系。

```
Switch(config)#mac-address-table static address 00-01-22-22-22-22 vlan 1 interface
ethernet 1/7
```

```
Switch(config)#mac-address-table static address 00-01-33-33-33-33 vlan 1 interface
ethernet 1/9
```

2.21.4 排错帮助

输入 `show mac-address-table` 命令时，发现某端口没有学习到该端口连接的设备的 MAC。可能的原因：

- ✎ 用于连接的以太网线损坏，更换以太网线；
- ✎ 交换机启动 `SpanningTree`，且端口处于 `discarding` 状态；或者端口刚连接上设备，`Spanning Tree` 还在计算中，等 `Spanning Tree` 计算完毕，端口就可以学习 MAC 地址了；
- ✎ 若不是上述的问题时，请查看是否是端口损坏，若是请找技术支持解决。

2.22 MAC Notification

2.22.1 MAC Notification 介绍

Mac notification 功能主要在于通知，mac 地址的新增或删除，就是当有 device 的加入或 device 移除时，通过 snmp 的 trap 功能通知网管发生的变化。

2.22.2 MAC Notification 配置

Mac notification 配置任务序列如下：

1. 配置全局 snmp mac notification 功能
2. 配置全局 mac notification 功能
3. 配置全局发送 mac notification 通知的时间间隔
4. 配置 history table 的大小
5. 配置端口支持 mac notification 的 trap 类型
6. mac notification 配置情况和报文数据
7. 清除 mac notification trap 的计数值

1. 配置全局 snmp mac notification 功能

命令	解释
全局配置模式	
snmp-server enable traps mac-notification no snmp-server enable traps mac-notification	配置或取消 snmp 全局 mac notification 功能

2. 配置全局 mac notification 功能

命令	解释
全局配置模式	
mac-address-table notification no mac-address-table notification	配置或取消全局 mac notification 功能

3. 配置全局发送 mac notification 通知的时间间隔

命令	解释
全局配置模式	
mac-address-table notification interval <0-86400> no mac-address-table notification interval	配置 mac notification 的时间间隔或者恢复默认时间间隔

4. 配置 history table 的大小

命令	解释
全局配置模式	
mac-address-table notification history-size <0-500> no mac-address-table notification history-size	配置 history table 的大小或者恢复该 table 大小为默认值

5. 配置端口支持 mac notification 的 trap 类型

命令	解释
端口配置模式	
mac-notification {added all removed moved} no mac-notification	配置或取消端口支持 mac notification 的 trap 类型

6. 显示 mac notification 配置情况和报文数据

命令	解释
特权模式	
show mac-notification summary	显示 MAC 通知的配置情况和通知报文数据

7. 清除 mac notification trap 的计数值

命令	解释
特权模式	
clear mac-notification statistics	清除 mac notification trap 的计数值

2.22.3 MAC Notification 举例

管理站（NMS）的 IP 地址是 1.1.1.5；交换机（Agent）的 IP 地址是 1.1.1.9。

管理站要接收来自交换机的 Trap 消息（注：管理站可能设置了对 Trap 的团体字符串的验证，那么此例假设管理站的 Trap 验证团体字符串为 usertrap）。

配置步骤如下：

```
Switch(config)#snmp-server enable
```

```
Switch(config)#snmp-server enable traps mac-notification
```

```
Switch(config)# mac-address-table notification
```

```
Switch(config)# mac-address-table notification interval 5
```

```
Switch(config)# mac-address-table notification history-size 100
```

```
Switch(Config-If-Ethernet1/4)#mac-notification both
```

2.22.4 MAC Notification 排错帮助

通过 show 和 snmp 部分的 debug 命令可以查看 trap 发送是否成功。

第3章 IP 业务操作

3.1 三层接口

3.1.1 三层接口介绍

在交换机上可以创建三层接口。三层接口并不是实际的物理接口，它是一个虚拟的接口。三层接口是在 VLAN 的基础上创建的。三层接口可以包含一个或多个二层端口（它们同属于一个 VLAN），但也可以不包含任何二层端口。三层接口包含的二层端口中，需要至少有一个是 UP 状态，三层接口才是 UP 状态，否则为 DOWN 状态。交换机中所有的三层接口缺省使用一个相同的 MAC 地址，此地址是在三层接口创建时从交换机保留的 MAC 地址中选取的。三层接口是三层协议的基础，在三层接口上可以配置 IP 地址，交换机可以通过配置在三层接口上的 IP 地址，与其它设备进行 IP 协议的传输。交换机也可以在不同的三层接口之间转发 IP 协议报文。Loopback 接口也属于三层接口。

3.1.2 三层接口配置

三层接口配置任务序列：

- 1. 创建三层接口
- 2. 配置三层接口的带宽
- 3. 配置 VLAN 接口描述
- 4. 打开或关闭 VLAN 接口
- 5. VRF 配置
 - (1) 创建VRF实例，并进入VPN实例视图
 - (2) 配置VRF实例的RD （可选）
 - (3) 配置VRF实例的RT （可选）
 - (4) 配置 VRF 实例与接口关联

1.创建三层接口

命令	解释
全局配置模式	
interface vlan <vlan-id> no interface vlan <vlan-id>	创建一个 VLAN 接口（VLAN 接口属于三层接口）；本命令的 no 操作为删除交换机创建的 VLAN 接口（三层接口）。
interface loopback <loopback-id> no interface loopback <loopback-id>	创建一个 Loopback 接口并进入 Loopback 接口配置模式；本命令的 no 操作为删除指定的 Loopback 接口。

2. 配置三层接口的带宽

命令	解释
VLAN 接口配置模式	
bandwidth <bandwidth> no bandwidth	配置 interface vlan 的带宽。本命令的 no 命令为恢复 interface vlan 的带宽值为默认值。

3. 配置 VLAN 接口描述

命令	解释
VLAN 接口配置模式	
description <text> no description	配置 VLAN 接口的描述信息；其 no 形式取消该 VLAN 接口的描述信息。

4. 打开或关闭 VLAN 接口

命令	解释
VLAN 接口配置模式	
shutdown no shutdown	打开或关闭 VLAN 接口。

5. VRF 配置

- (1) 创建VRF实例，并进入VPN实例视图
- (2) 配置VRF实例的RD （可选）
- (3) 配置VRF实例的RT （可选）
- (4) 配置 VRF 实例与接口关联

命令	解释
全局配置模式	
ip vrf <vrf-name> no ip vrf <vrf-name>	创建 VRF 实例；缺省情况下未创建 VRF 实例。
VRF 配置模式	
rd <ASN:nn_or_IP-address:nn>	配置 VRF 实例的 RD。缺省情况下未创建 RD。
route-target {import export both} <rt-value> no route-target {import export both} <rt-value>	配置 VRF 实例的 RT。
接口配置模式	

ip vrf forwarding <vrf-name> no ip vrf forwarding <vrf-name>	配置 VRF 实例与接口关联。
ip address <ip-address> <mask> no ip address <ip-address> <mask>	配置直连接口的私网 IP 地址。

3.2 网管端口

3.2.1 IP 网管端口介绍

网管端口位于主控卡上，标注为 Ethernet，软件配置名称为 Ethernet0，用户可以使用命令 `interface Ethernet 0` 进入到网管端口配置模式。通过以太网线连接到网管端口的用户可以通过 Telnet、Web 等方式对交换机进行配置管理。

3.2.2 IP 网管端口配置

网管端口配置任务序列如下：

1. 进入网管端口配置模式
2. 配置网管端口的属性
 - (1) 打开或关闭端口
 - (2) 配置端口速率
 - (3) 配置端口双工模式

配置端口 IP 地址

1. 进入网管端口配置模式

命令	解释
全局配置模式	
interface ethernet <num>	进入网管端口配置模式。

2. 配置网管端口的属性

命令	解释
网管端口配置模式	
shutdown no shutdown	关闭或打开网管端口。
ip address <ip-address> <mask> no ip address [<ip-address> <mask>]	配置或取消网管端口的 IP 地址。

3.3 IP 配置

3.3.1 IPv4、IPv6 介绍

IPv4 是全球通用因特网协议的当前版本。事实证明，IPv4 简单、灵活、开放、稳固耐用、便于实施，且能够和多种上下层协议良好协同工作。尽管 IPv4 自 20 世纪 80 年代初确立以来就几乎未曾改动，但是 IPv4 始终支持因特网的升迁，一直发展到目前的全球规模。然而，随着因特网基础设施与因特网应用服务的发展不断地突飞猛进，IPv4 在因特网的目前规模与复杂性面前已暴露其不足之处。

IPv6 指的是互联网协议第六版，是由 IETF 设计的下一代互联网协议，用以取代现有的互联网协议第四版（IPv4）。IPv6 是专为弥补 IPv4 不足而开发出来的，以便让因特网能够进一步发展壮大。

IPv6 所解决的最重要问题就是增加 IP 地址的数量。IPv4 地址已近枯竭，而因特网用户的数量却在不断以几何级数增长。随着需要使用 IP 地址的因特网服务与应用设备（利用因特网的信息终端、家庭与小型办公室网络、IP 电话与无线服务等）不断大量涌现，IP 地址的供给更显紧张。人们早就开始着手解决 IPv4 地址紧缺的问题，采用各种技术延长现有 IPv4 基础架构的寿命，其中包括网络地址转换（Network Address Translation，简称 NAT）和无类别域间路由（Classless Inter-Domain Routing，简称 CIDR）等技术。

虽然 CIDR、NAT 和私有编址的组合暂时缓和了 IPv4 地址空间紧缺的问题，但是 NAT 技术破坏了 IP 设计初衷的端到端模型，使作为网络中间节点的路由设备必须保持每个连接的状态，大大增加了网络延迟，降低了网络性能。而且网络数据包地址的变换阻碍了端到端的网络安全检查，如 IPSec 认证报头(AH)就是一个例子。

因此，要综合解决 IPv4 存在的诸多问题，IETF 设计的下一代互联网协议 IPv6 已经成为目前唯一可行的解决方案。

首先 IPv6 协议 128 比特的编址方案能确保在时间和空间范围内为全球 IP 网络节点提供足够的全球唯一的 IP 地址。而且，除了增加地址空间以外，IPv6 还对 IPv4 的其它许多关键设计进行了改进。

层次化编址方案有助于路由聚合，有效减少了路由器的路由表项，提高了路由选择与数据包处理的效率与可扩展性。

IPv6 的包头设计相比 IPv4 更有效率，数据字段更少，去掉了包头校验和，从而加快了基本 IPv6 包头的处理速度。在 IPv6 包头中，分片字段作为可选扩展字段出现，路由器转发过程中不用再对数据包做分片处理，通过路径 MTU 发现机制协同数据包源点工作，提高了路由器处理效率。

支持地址自动配置与即插即用。IPv6 的地址自动配置功能使大量 IP 主机能够轻松发现网络路由器，并自动获得全球唯一的 IPv6 地址，这使利用 IPv6 因特网的设备具备了即

插即用特性。自动地址配置功能还使对现有网络的重新编址变得更加简单便捷，使网络运营商能够更加方便地管理从一个提供商到另一个提供商的转换。

支持 IPSec。IPSec 在 IPv4 中为可选项，而在 IPv6 协议中则是必须实现的。IPv6 提供了安全扩展包头，能够提供诸如访问控制、机密性与数据完整性等端到端的安全服务，从而使加密、验证和虚拟专用网络 (VPN) 的实施变得更加容易。

增强对移动 IP (Mobile IP) 与移动计算设备的支持。在 IETF 标准中定义的移动 IP 协议使移动设备不必脱离其现有连接即可自由移动，这是一种日益重要的网络功能。与 IPv4 不同的是，IPv6 的移动性是使用内置自动配置获取转交地址 (Care-Of-Address)，因而无需外地代理 (Foreign Agent)。此外，这种联编过程使通信节点 (Correspondent Node) 能够与移动节点 (Mobile Node) 直接通信，从而避免了在 IPv4 中所要求的三角路由选择的额外系统开销。其结果是，在 IPv6 中，移动 IP 的处理效率大为提高。

避免网络地址转换 (NAT) 的使用。NAT 机制的引入是为了在不同的网络区段之间共享和重新使用相同的地址空间。这种机制在暂时缓解了 IPv4 地址紧缺问题的同时，却为网络设备与应用程序增加了处理地址转换的负担。由于 IPv6 的地址空间大大增加，也就无需再进行地址转换，NAT 部署带来的问题与系统开销也随之解决。

支持广泛部署的路由选择协议。IPv6 保持并扩展了对现有内部网关协议 (Interior Gateway Protocols, 简称 IGP) 与外部网关协议 (Exterior Gateway Protocols, 简称 EGP) 的支持，例如，RIPng、OSPFv3、IS-ISv6 与 MBGP4+ 等 IPv6 路由协议。

组播地址数量增加，对组播的支持有所增强。IPv6 取消了广播功能，使用组播机制来处理 IPv4 的广播功能，不仅节省了网络带宽，而且提高了网络效率。

3.3.2 IP 配置

可以将三层接口配置为 IPv4 接口或者 IPv6 接口。

3.3.2.1 IPv4 地址配置

IPv4 配置任务序列：

1. 配置三层接口的 IPv4 地址

1. 配置三层接口的 IPv4 地址

命令	解释
VLAN 接口配置模式	
ip address <ip-address> <mask> [secondary] no ip address [<ip-address> <mask>]	配置 VLAN 接口的 IP 地址；本命令的 no 操作为删除 VLAN 接口 IP 地址。

3.3.2.2 IPv6 地址配置

IPv6 配置任务序列如下：

1. IPv6 基本配置
 - (1) 配置接口 IPv6 地址
 - (2) 配置 IPv6 静态路由
2. IPv6 邻居发现配置
 - (1) 配置 DAD 邻居请求消息数目
 - (2) 配置发送邻居请求消息间隔
 - (3) 使能与禁止路由器公告
 - (4) 配置路由器公告生存期
 - (5) 配置路由器公告最小间隔时间
 - (6) 配置路由器公告最大间隔时间
 - (7) 配置前缀公告参数
 - (8) 设置静态邻居表项
 - (9) 清除邻居表项
 - (10) 设置发送路由公告的 hoplimit 值
 - (11) 设置发送路由公告的 mtu 值
 - (12) 设置发送路由公告的 reachable-time 值
 - (13) 设置发送路由公告的 retrans-timer 值
 - (14) 配置标识除地址信息之外的其他信息是否由 dhcpv6 获取
 - (15) 配置标识地址信息是否由 dhcpv6 获取
3. IPv6 隧道配置
 - (1) 创建/删除隧道
 - (2) 配置隧道描述
 - (3) 配置隧道源
 - (4) 配置隧道目的
 - (5) 配置隧道下一跳
 - (6) 配置隧道模式
 - (7) 配置隧道路由

1. IPv6 基本配置

(1) 配置接口 IPv6 地址

命令	解释
接口配置模式	
ipv6 address <ipv6-address/prefix-length> [eui-64] no ipv6 address <ipv6-address /prefix-length>	配置 IPv6 地址，包括可聚合全球单播地址，本地站点地址，本地链路地址。本命令的 no 操作为删除 IPv6 地址。

(2) 设置 IPv6 静态路由

命令	解释
全局配置模式	
ipv6 route <ipv6-prefix/prefix-length> {<nexthop-ipv6-address> <interface-type interface-number> {<nexthop-ipv6-address> <interface-type interface-number>}} [distance] no ipv6 route <ipv6-prefix/prefix-length> {<nexthop-ipv6-address> <interface-type interface-number> {<nexthop-ipv6-address> <interface-type interface-number>}} [distance]	配置 IPv6 静态路由。本命令的 no 操作 为删除 IPv6 静态路由。

2. IPv6 邻居发现配置

(1) 配置 DAD 邻居请求消息数目

命令	解释
接口配置模式	
ipv6 nd dad attempts <value> no ipv6 nd dad attempts	设置接口进行重复地址检测时，连续发出的邻居请求消息数目，本命令的 no 操作为恢复默认值（1）。

(2) 配置发送邻居请求消息间隔

命令	解释
接口配置模式	
ipv6 nd ns-interval <seconds> no ipv6 nd ns-interval	设置接口发送邻居请求消息的时间间隔。本命令的 no 操作为恢复默认值（1 秒）。

(3) 使能与禁止路由器公告

命令	解释
接口配置模式	
ipv6 nd suppress-ra no ipv6 nd suppress-ra	禁止 IPv6 路由器公告，本命令的 no 操作为开启 IPv6 路由器公告。

(4) 配置路由器公告生存期

命令	解释
接口配置模式	
ipv6 nd ra-lifetime <seconds> no ipv6 nd ra-lifetime	配置路由器公告的生存期, 本命令的 no 操作为恢复默认值 (1800 秒)。

(5) 配置路由器公告最小间隔时间

命令	解释
接口配置模式	
ipv6 nd min-ra-interval <seconds> no ipv6 nd min-ra-interval	配置用于路由器公告的最小时间间隔, 本命令的 no 操作为恢复默认值 (200 秒)。

(6) 配置路由器公告最大间隔时间

命令	解释
接口配置模式	
ipv6 nd max-ra-interval <seconds> no ipv6 nd max-ra-interval	配置用于路由器公告的最大时间间隔, 本命令的 no 操作为恢复默认值 (600 秒)。

(7) 配置前缀公告参数

命令	解释
接口配置模式	
ipv6 nd prefix <ipv6-address/prefix-length> <valid-lifetime> <preferred-lifetime> [off-link] [no-autoconfig] no ipv6 nd prefix <ipv6-address/prefix-length> <valid-lifetime> <preferred-lifetime> [off-link] [no-autoconfig]	配置路由器公告的地址前缀及其公告参数, 本命令的 no 操作为删除路由公告的地址前缀。

(8) 设置静态邻居表项

命令	解释

接口配置模式	
ipv6 neighbor <ipv6-address> <hardware-address> interface <interface-type interface-name>	设置静态邻居表项，包括邻居 IPv6 地址，MAC 地址，二层端口。
no ipv6 neighbor <ipv6-address>	删除邻居表项

(9) 清除邻居表项

命令	解释
特权配置模式	
clear ipv6 neighbors	清除所有静态邻居表项。

(10) 设置发送路由公告的 hoplimit 值

命令	解释
接口配置模式	
ipv6 nd ra-hoplimit <value>	设置发送路由公告的 hoplimit 值。

(11) 设置发送路由公告的 mtu 值

命令	解释
接口配置模式	
ipv6 nd ra-mtu <value>	设置发送路由公告的 mtu 值。

(12) 设置发送路由公告的 reachable-time 值

命令	解释
接口配置模式	
ipv6 nd reachable-time <seconds>	设置发送路由公告的 reachable-time 值。

(13) 设置发送路由公告的 retrans-timer 值

命令	解释
接口配置模式	
ipv6 nd retrans-timer <seconds>	设置发送路由公告的 retrans-timer 值。

(14) 配置标识除地址信息之外的其他信息是否由 dhcpv6 获取

命令	解释
接口配置模式	
ipv6 nd other-config-flag	标识除地址信息之外的其他信息是否由 DHCPv6 获取。

(15) 配置标识地址信息是否由 dhcpv6 获取

命令	解释
接口配置模式	
ipv6 nd managed-config-flag	标识地址信息是否由 DHCPv6 获取。

3. IPv6 隧道配置

(1) 创建/删除隧道

命令	解释
全局配置模式	
interface tunnel <tnl-id> no interface tunnel <tnl-id>	创建一条隧道。本命令的 no 操作为删除一条隧道。

(2) 配置隧道描述

命令	解释
隧道配置模式	
description <desc> no description	配置隧道描述, 本命令的 no 操作删除隧道描述。

(3) 配置隧道源

命令	解释
隧道配置模式	
tunnel source {<ipv4-address> <ipv6-address> <interface-name>} no tunnel source	配置隧道源端 IPv4/IPv6 地址, 或者允许隧道源取自接口的主 IPv4/IPv6 地址, 本命令的 no 操作为删除隧道源端的 IPv4/IPv6 地址或隧道源接口名。

(4) 配置隧道目的

命令	解释
隧道配置模式	
tunnel destination {<ipv4-address> <ipv6-address>} no tunnel destination	配置隧道目的端的 IPv4/IPv6 地址, 本命令的 no 操作为删除隧道目的端的 IPv4/IPv6 地址。

(5) 配置隧道下一跳

命令	解释

隧道配置模式	
tunnel nexthop <ipv4-address> no tunnel nexthop	配置隧道的下一跳 IPv4 地址，本命令的 no 操作作为删除隧道的下一跳 IPv4 地址。

(6) 配置隧道模式

命令	解释
隧道配置模式	
tunnel mode [[gre] ipv6ip [6to4 isatap]] no tunnel mode	配置隧道模式，本命令的 no 操作为清除隧道模式。

(7) 配置隧道路由

命令	解释
全局配置模式	
ipv6 route <ipv6-address/prefix-length> {<interface-type interface-number> tunnel <tnl-id>} no ipv6 route <ipv6-address/prefix-length> {<interface-type interface-number> tunnel <tnl-id>}	配置隧道路由，本命令的 no 操作为删除隧道路由。

3.3.3 IP 配置举例

3.3.3.1 IPv4 典型案例

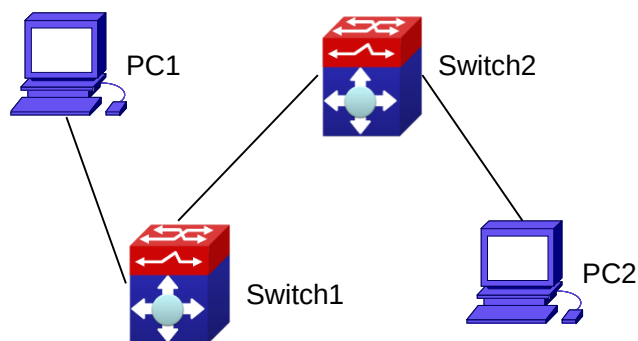


图 3-1 IPv4 典型案例

用户有如下配置需求：在 switch 1 和 switch 2 上配置不同网段的 IPv4 地址，配置静态路由，利用 ping 功能验证可达性。

配置说明：

- 1、在 Switch1 上配置两个 vlan，分别为 vlan1 和 vlan2
- 2、在 Switch1 的 vlan1 内配置 IPv4 地址 192.168.1.1 255.255.255.0，在 vlan2 内配置 IPv4 地址 192.168.2.1 255.255.255.0
- 3、在 Switch2 上配置两个 vlan，分别为 vlan2 和 vlan3
- 4、在 Switch2 的 vlan2 内配置 IPv4 地址 192.168.2.2 255.255.255.0，在 vlan3 内配置 IPv4 地址 192.168.3.1 255.255.255.0
- 5、PC1 的 IPv4 地址为 192.168.1.100 255.255.255.0，PC2 的 IPv4 地址是 192.168.3.100 255.255.255.0
- 6、在 Switch1 上配置静态路由 192.168.3.0/24，在 Switch2 上配置静态路由 192.168.1.0/24
- 7、PC 机之间互 ping

注：首先要确定 PC1 和 Switch1 可以 ping 通，PC2 和 Switch2 可以 ping 通

配置步骤如下：

```
Switch1(Config)#interface vlan 1
Switch1(Config-if-Vlan1)#ip address 192.168.1.1 255.255.255.0
Switch1(Config)#interface vlan 2
Switch1(Config-if-Vlan2)#ip address 192.168.2.1 255.255.255.0
Switch1(Config-if-Vlan2)#exit
Switch1(Config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2
```

```
Switch2(Config)#interface vlan 2
Switch2(Config-if-Vlan2)#ip address 192.168.2.2 255.255.255.0
Switch2(Config)#interface vlan 3
Switch2(Config-if-Vlan3)#ip address 192.168.3.1 255.255.255.0
Switch2(Config-if-Vlan3)#exit
```

```
Switch2(Config)#ip route 192.168.1.0 255.255.255.0 192.168.2.1
```

3.3.3.2 IPv6 典型案例

案例 1:

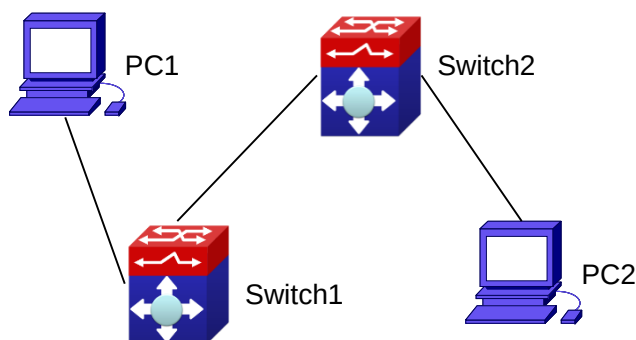


图 3-2 IPv6 典型案例

用户有如下配置需求：在 Switch 1 和 Switch 2 上配置不同网段的 IPv6 地址，配置静态路由，利用 ping6 功能验证可达性。

配置说明：

- 1、在 Switch1 上配置两个 vlan，分别为 vlan1 和 vlan2
- 2、在 Switch1 的 vlan1 内配置 IPv6 地址 2001::1/64,在 vlan2 内配置 IPv6 地址 2002::1/64
- 3、在 Switch2 上配置两个 vlan，分别为 vlan2 和 vlan3
- 4、在 Switch2 的 vlan2 内配置 IPv6 地址 2002::2/64，在 vlan3 内配置 IPv6 地址 2003::1/64
- 5、PC1 的 IPv6 地址为 2001::11/64,PC2 的 IPv6 地址是 2003::33/64
- 6、在 Switch1 上配置静态路由 2003::33/64,在 Switch2 上配置静态路由 2001::11/64
- 7、pc 机之间互相 ping6

注：首先要确定 PC1 和 Switch1 可以 ping 通，PC2 和 Switch2 可以 ping 通

配置步骤如下：

```
Switch1(Config)#interface vlan 1
Switch1(Config-if-Vlan1)#ipv6 address 2001::1/64
Switch1(Config)#interface vlan 2
Switch1(Config-if-Vlan2)#ipv6 address 2002::1/64
Switch1(Config-if-Vlan2)#exit
Switch1(Config)#ipv6 route 2003::33/64 2002::2
```

```
Switch2(Config)#interface vlan 2
```

```
Switch2(Config-if-Vlan2)#ipv6 address 2002::2/64
Switch2(Config)#interface vlan 3
Switch2(Config-if-Vlan3)#ipv6 address 2003::1/64
Switch2(Config-if-Vlan3)#exit
Switch2(Config)#ipv6 route 2001::33/64 2002::1
```

```
Switch1#ping6 2003::33
```

配置结果:

```
Switch1#show run
interface Vlan1
    ipv6 address 2001::1/64
!
interface Vlan2
    ipv6 address 2002::2/64
!
interface Loopback
    mtu 3924
!
ipv6 route 2003::/64 2002::2
!
no login
!
end
```

```
Switch2#show run
interface Vlan2
    ipv6 address 2002::2/64
!
interface Vlan3
    ipv6 address 2003::1/64
!
interface Loopback
    mtu 3924
!
ipv6 route 2001::/64 2002::1
!
no login
!
```

End

案例 2:

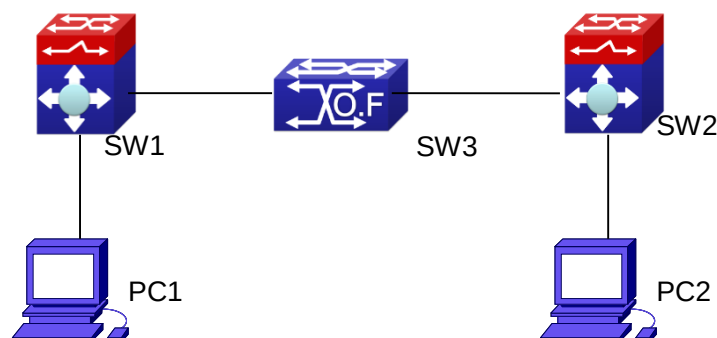


图 3-3 IPv6 隧道

该案例为 IPv6 隧道，用户有如下配置需求：SW1 和 SW2 为隧道的节点，支持双栈。SW3 只是运行 IPv4，PC1 和 PC2 进行通信

配置说明：

- 1、在 SW1 上配置两个 vlan，分别为 vlan1 和 vlan2。Vlan1 是 IPv6 域，vlan2 连接 IPv4 域
- 2、在 SW1 的 vlan1 中配置 IPv6 地址 2002:caca:ca01:2::1/64，并且打开 RA 功能，在 vlan2 配置 IPv4 地址 202.202.202.1
- 3、在 SW2 上配置两个 vlan，分别是 vlan3 和 vlan4，vlan4 是 IPv6 域，vlan3 是连接 IPv4 域
- 4、在 SW2 的 vlan4 中配置 IPv6 地址 2002:cbcb:cb01:2::1/64，并且打开 RA 功能，在 vlan3 上配置 IPv4 地址 203.203.203.1
- 5、在 SW1 上配置手工隧道，隧道的源 IPv4 地址为 202.202.202.1，配置默认 IPv6 路由指向该隧道
- 6、在 SW2 上配置手工隧道，隧道的源 IPv4 地址为 203.203.203.1，配置默认 IPv6 路由指向该隧道
- 7、在 SW3 上配置两个 vlan，分别为 vlan2 和 vlan3，在 vlan2 上配置 IPv4 地址 202.202.202.202 在 vlan3 上配置 IPv4 地址 203.203.203.203
- 8、PC1 和 PC2 通过 SW1 和 SW2 获得 2002 的前缀自动配置 IPv6 地址
- 9、在 PC1 上 ping PC2 的 IPv6 地址

配置步骤如下：

```
SW1(Config-if-Vlan1)#ipv6 address 2002:caca:ca01:2::1/64
SW1(Config-if-Vlan1)#no ipv6 nd suppress-ra
SW1(Config-if-Vlan1)#interface vlan 2
SW1(Config-if-Vlan2)#ipv4 address 202.202.202.1 255.255.255.0
SW1(Config-if-Vlan1)#exit
SW1(config)# interface tunnel 1
```

```
SW1(Config-if-Tunnel1)#tunnel source 202.202.202.1
SW1(Config-if-Tunnel1)#tunnel destination 203.203.203.1
SW1(Config-if-Tunnel1)#tunnel mode ipv6ip
SW1(config)#ipv6 route ::/0 tunnel1

SW2(Config-if-Vlan4)#ipv6 address 2002::cbcb:cb01::2/64
SW2(Config-if-Vlan4)#no ipv6 nd suppress-ra
SW2 (Config-if-Vlan3)#interface vlan 3
SW2 (Config-if-Vlan2)#ipv4 address 203.203.203.1 255.255.255.0
SW2 (Config-if-Vlan1)#exit
SW2(Config)#interface tunnel 1
SW2(Config-if-Tunnel1)#tunnel source 203.203.203.1
SW2(Config-if-Tunnel1)#tunnel destination 202.202.202.1
SW2(Config-if-Tunnel1)#tunnel mode ipv6ip
SW2(config)#ipv6 route ::/0 tunnel1
```

3.3.4 IPv6 排错帮助

- ☞ 配置路由器的生存期时不应该小于发送路由器公告的时间间隔。如果相连 PC 未获得 IPv6 地址时，应注意 RA 公告的开关（默认为关闭）。

3.4 IP 转发

3.4.1 IP 转发介绍

网关设备可以将 IP 协议报文从一个子网转发到另一个子网中，这种转发是通过路由寻径的。交换机的 IP 转发是由硬件协助完成的，可以达到端口的线速转发；同时还可以提供各种灵活的控制，对转发行为进行调整和监控。交换机可以支持禁止或允许优化的聚合算法以调整交换芯片中网段路由表项的生成，查看 IP 转发的统计信息，监视 IP 报文的收发状况，以及查看硬件转发芯片的状况。

3.4.2 IP 路由聚合配置

IP 路由聚合配置任务序列：

1. 配置是否使用优化 IP 路由聚合算法

1.配置是否使用优化 IP 路由聚合算法

命令	解释
全局配置模式	

ip fib optimize no ip fib optimize	配置交换机使用优化 IP 路由聚合算法；本命令的 no 操作为交换机不使用优化 IP 路由聚合算法。

3.5 URPF

3.5.1 URPF 介绍

URPF（Unicast Reverse Path Forwarding，单播逆向路径转发）将组播中使用的RPF技术应用到单播中，用于防止基于源地址欺骗的网络攻击行为。

交换机接收报文，同时获取报文的源地址和入接口，然后把该源地址作为目的地址在路由表中查找路由，如果查到的路由出接口和接收该报文的入接口不匹配，交换机则认为该报文的源地址是伪造的，并丢弃该报文。

源地址欺骗攻击为入侵者构造出一系列带有伪造源地址的报文，对于使用基于IP 地址验证的应用来说，此攻击方法可以导致未被授权用户以他人身份获得访问系统的权限，甚至是以管理员权限来访问。即使响应报文不能达到攻击者，同样也会造成对被攻击对象的破坏。

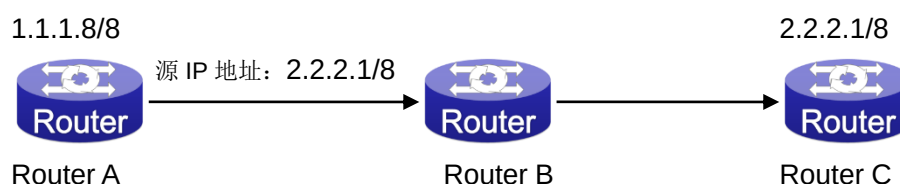


图 3-4 URPF 应用环境

如上图所示，在Router A上伪造源地址为2.2.2.1/8 的报文，向服务器Router B发起请求，Router B响应请求时将向真正的“2.2.2.1/8”发送报文。这种非法报文对Router B和Router C都造成了攻击。URPF 技术可以应用在上述环境中，阻止基于源地址欺骗的攻击。

3.5.1.1 IPv6 URPF 运行机制

目前URPF的运行机制依赖于交换机芯片提供的ACL功能。

首先全局使能URPF功能监视路由表变化：将路由表FIB中的每条路由生成对应的URPF permit ACL规则。URPF严格模式下，ACL规则的形式为：入数据包源地址网段+入数据包入接口VID。入数据包源地址网段对应FIB表路由表项的目的网段地址，入数据包入接口VID对应FIB表路由表项的出接口VID。URPF松散模式下，ACL规则的形式为入数据包源地址网段，入数据包源地址网段对应FIB表路由表项的目的网段地址。

在端口上启动URPF后：将端口绑定URPF规则，并且生成默认的DENY ALL规则下发硬件。

通过以上操作，就可以确保，在数据到达此端口时，只有符合上述规则的数据才能进入端口，其它数据都会被丢掉。

目前对应URPF的ACL规则优先级较低，不会阻挡各种协议包数据，因此启动该功能不影响该交换机路由协议的正常运行。

3.5.2 URPF 配置任务序列

- 1) 启动 URPF
- 2) 端口启动 URPF
- 3) 显示和调试 URPF 相关信息

1) 全局启动 URPF

命令	解释
全局配置模式	
urpf enable no urpf enable	全局启动和关闭 URPF。

2) 端口启动 URPF

命令	解释
端口配置模式	
ip urpf enable {loose strict} {allow-default-route } no ip urpf enable	端口启动和关闭 URPF。

3) 显示和调试 URPF 相关信息

命令	解释
特权配置模式	
debug urpf no debug urpf	关闭/打开 URPF 调试功能，在 URPF 规则安装失败时提示错误。
特权和配置模式	
show urpf	显示哪些端口启动了 URPF。
show urpf rule ipv4 num interface ethernet IFNAME	显示端口绑定的 IPv4 规则数目。
show urpf rule ipv6 num interface ethernet IFNAME	显示端口绑定的 ipv6 规则数目。
show urpf rule ipv4 interface ethernet IFNAME	显示端口绑定的 ipv4 规则明细。
show urpf rule ipv6 interface ethernet IFNAME	显示端口绑定的 ipv6 规则明细。

3.5.3 URPF 典型案例

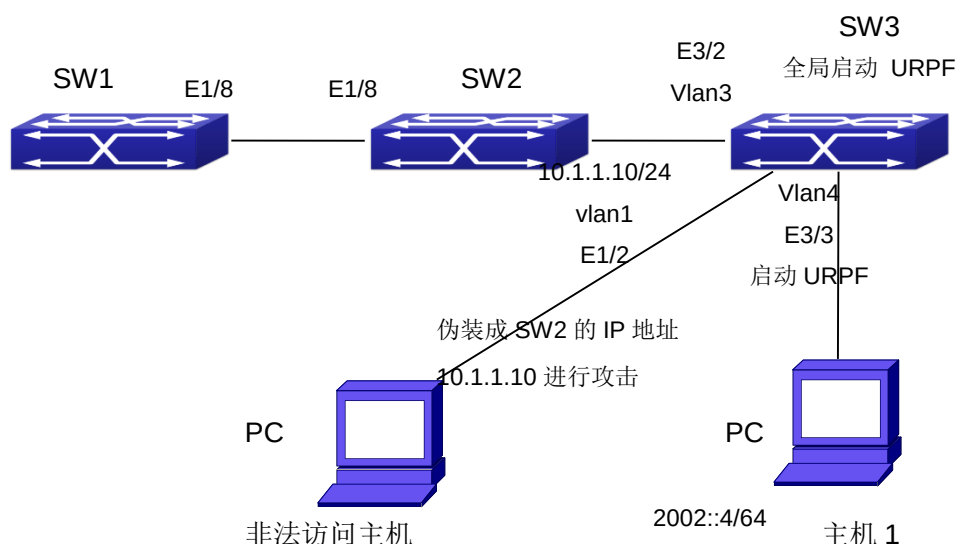


图 3-5 URPF 典型案例

如上图所示的网络拓扑中，一台 SW3 上启动 URPF 功能。当链接的网络中有人冒充别人的 IP 地址进行恶意攻击时，交换机通过硬件 FFP 功能直接丢弃所有的攻击报文。设置 SW3 Ethernet3/3 启动 URPF 功能。

SW3 配置任务序列：

```
Switch#config
```

```
Switch(config)#urpf enable
```

```
Switch(config)#interface ethernet 3/3
```

```
Switch(Config-If-Ethernet3/3)#ip urpf enable strict
```

3.5.4 URPF 排错帮助

URPF协议的正常运行非常依赖URPF对应规则的正确下发，如果在配置URPF后，发现功能与预期不符：

- ☞ 判断是否因为交换机配置了与 URPF 冲突的规则（URPF 的优先级要小于 ACL），如果发生互斥，则 ACL 的规则起作用。
- ☞ 判断 FIB 表中是否存在相关的路由，只有 FIB 表中存在该路由后，才会在该端口下发相应的 ACL 规则。
- ☞ 判断是否硬件的 ACL 表现已满，从而导致无法对新生成的路由下发 ACL 规则。
- ☞ 如果在正常配置下，URPF 运行的情况仍然和预期不匹配，请打开 URPF 的调试功能，并且利用 `show urpf` 和相关的显示规则数目以及明细规则的命令观察生成的 URPF 规

则是否正常，并将结果发送给技术服务中心。

3.6 ARP

3.6.1 ARP 介绍

ARP（Address Resolution Protocol）地址解析协议，主要用于 IP 地址到以太网 MAC 地址的解析。交换机除了支持动态 ARP 外，也支持静态配置。此外，在某些应用中，交换机还支持配置代理 ARP。如当交换机的接口收到某 ARP 请求，该请求的 IP 地址与接口地址在一个 IP 网段，但却不在同一物理网络中，此时该接口若启动了代理 ARP 的功能，接口会将自己的 MAC 地址作为 ARP 的回应，然后将收到的实际数据报文进行转发。启动代理 ARP 功能可以使因为物理网络分离但属于同一 IP 网段的机器忽视物理网络分离的事实，经过具有代理 ARP 接口的转发像处在一个物理网络中。

3.6.2 ARP 配置

ARP 配置任务序列：

1. 配置静态 ARP
2. 配置代理 ARP
3. 清除动态 ARP
4. 选择 hash 算法
5. 清除 ARP 报文统计信息

1. 配置静态 ARP

命令	解释
VLAN 接口模式	
arp <ip_address> <mac_address> {interface [ethernet] <portName> } no arp <ip_address>	配置静态 ARP 表项；本命令的 no 操作为删除指定 IP 地址的 ARP 表项。

2. 配置代理 ARP

命令	解释
VLAN 接口模式	
ip proxy-arp no ip proxy-arp	打开以太网代理 ARP 的功能；本命令的 no 操作为关闭代理 ARP 的功能。

3.清除动态 ARP

命令	解释

特权用户模式	
clear arp-cache	清除交换机学习到的动态 ARP。

4. 选择 hash 算法

命令	解释
全局配置模式	
l3 hashselect [<crc16l crc16u crc32l crc32u lsb>]	设置 L3 表的 hash 算法，该命令的使用涉及到 ARP 表项在硬件中的存储，必须在厂商技术人员的指导下使用，该命令的详细信息请参考命令手册的相应部分。

5. 清除 ARP 报文统计信息

命令	解释
特权用户模式	
clear arp traffic	清除交换机 ARP 报文统计信息。

3.6.3 ARP 转发排错帮助

交换机无法 PING 通直接相连的网络设备，检查可能存在的情况和建议的解决方法：

- ☞ 首先检查交换机是否学习到相应的 ARP。
- ☞ 若 ARP 未能学习到，那么使用 ARP 的调试信息，观察 ARP 协议报文的收发情况。
- ☞ 用户比较容易遇到的现象是线缆有问题，导致 ARP 不能学习。

3.7 I3 station movement

3.7.1 I3 station movement 介绍

正常情况下当出现 arp/nd 切换端口的时候，根据 arp/nd 报文重新学习 arp/nd 表项的端口信息。如果当 PC 或其他网络节点切换端口的时候，没有发送或者接收 ARP 报文的情况我们认为是非安全切换，因此不予重新学习。新的“I3 station movement”的功能是为了满足特殊条件下的 arp/nd 切换端口的功能，实现原理为，当打开本功能时候的，如果出现 MAC 切换端口的时候，认为这是安全的网络节点切换，任何从新端口收到的网络报文（src mac 为发生切换的网络节点）触发 arp/nd 切换，重新学习 arp/nd 到新的端口。

3.7.2 I3 station movement 配置

1. 启动/关闭 I3-station-move 功能

命令	解释
全局配置模式	
l3-station-move no l3-station-move	启动或者关闭 l3 station move 功能。

3.8 防 ARP 扫描

3.8.1 防 ARP 扫描功能介绍

ARP 扫描是一种常见的网络攻击方式。为了探测网段内的所有活动主机，攻击源将会产生大量的 ARP 报文在网段内广播，这些广播报文极大的消耗了网络的带宽资源；攻击源甚至有可能通过伪造的 ARP 报文而在网络内实施大流量攻击，使网络带宽消耗殆尽而瘫痪。而且 ARP 扫描通常是其他更加严重的攻击方式的前奏，如病毒自动感染，或者继而进行端口扫描、漏洞扫描以实施如信息窃取、畸形报文攻击，拒绝服务攻击等。

由于 ARP 扫描给网络的安全和稳定带来了极大的威胁，所以防 ARP 扫描功能将具有重大意义。交换机防 ARP 扫描的整体思路是若发现网段内存在具有 ARP 扫描特征的主机或端口，将切断攻击源头，保障网络的安全。

有两种方式来防 ARP 扫描：基于端口和基于 IP。基于端口的 ARP 扫描会计算一段时间内从某个端口接收到的 ARP 报文的数量，若超过了预先设定的阈值，则会 down 掉此端口。基于 IP 的防 arp 扫描设置限速和隔离两级阈值，当超过一级阈值（限速阈值）后，硬件对该主机的 ARP 报文（包括 ARP 请求和应答）仍然正常转发，只是对送 CPU 的速率进行限制，并产生 trap 告警，通知管理员可能存在攻击；当报文速率达到二级阈值（隔离阈值）时，再采取隔离措施，记录日志，产生 trap 告警。一级阈值限速和二级阈值隔离在全局使能基于 IP 的防 ARP 扫描后同步开启，一级阈值必须小于二级阈值才能生效。此两种防 arp 扫描功能可以同时启用，端口被禁掉之后可以通过配置自动恢复功能自动恢复其状态。IP 被禁掉之后，只要接收到 arp 报文的速率降到二级阈值之下即可自动恢复。

为了提高交换机的效率，可以配置受信任的端口和 IP，交换机不检测来自受信任的端口或 IP 的 ARP 报文，这样可以有效地减少交换机的负担。

3.8.2 防 ARP 扫描配置任务序列

- 1) 启动防 ARP 扫描功能
- 2) 配置基于端口和基于 IP 的防 ARP 扫描的阈值
- 3) 配置信任端口
- 4) 配置信任 IP
- 5) 配置自动恢复时间

6) 显示和调试防 ARP 扫描相关信息

7) 配置超出阈值之后的动作。

1) 启动防 ARP 扫描功能

命令	解释
全局配置模式	
anti-arpscan enable [ip port] no anti-arpscan enable [ip port]	全局启动或关闭防 ARP 扫描功能。

2) 配置基于端口和基于 IP 的防 ARP 扫描的阈值

命令	解释
全局配置模式	
anti-arpscan port-based threshold <threshold-value> no anti-arpscan port-based threshold	设置基于端口的防 ARP 扫描阈值。
anti-arpscan ip-based level1 level2 threshold <threshold-value> no anti-arpscan ip-based level1 level2 threshold	设置基于 IP 的防 ARP 扫描的接收 ARP 报文的一级或二级阈值，一级阈值默认为 4p/s，二级默认为 8p/s。二级阈值必须大于一级阈值。

3) 配置信任端口

命令	解释
端口配置模式	
anti-arpscan trust <port supertrust-port iptrust-port > no anti-arpscan trust <port supertrust-port iptrust-port >	设置端口的信任属性。

4) 配置信任 IP

命令	解释
全局配置模式	
anti-arpscan trust ip <ip-address> [<netmask>] no anti-arpscan trust ip <ip-address> [<netmask>]	设置 IP 的信任属性。

5) 配置自动恢复时间

命令	解释
全局配置模式	
anti-arpscan recovery enable no anti-arpscan recovery enable	启动或关闭自动恢复功能。
anti-arpscan recovery time <seconds> no anti-arpscan recovery time	设置自动恢复时间。

6) 显示和调试防 ARP 扫描相关信息

命令	解释
全局配置模式	
anti-arpscan log enable no anti-arpscan log enable	打开或关闭防 ARP 扫描的日志功能。
anti-arpscan trap enable [level1 level2] no anti-arpscan trap enable [level1 level2]	打开或关闭防 ARP 扫描的 SNMP Trap 功能。
anti-arpscan FFP max-num <num>	防 ARP 扫描功能可占用的 FFP 表项最大数量。
show anti-arpscan [trust {ip port supertrust-port iptrust-port} prohibited {ip port}]	显示防 ARP 扫描的运行和配置情况。
show anti-arpscan ip-based attack-list [history]	显示防 ARP 扫描攻击源信息或历史攻击源信息
show anti-arpscan ip-based running-config	显示防 arp 扫描的当前配置
clear anti-arpscan speed-limit< IP Address>	手动清除对特定主机的 ARP 限速
clear anti-arpscan ip-isolate<IP Address>	手动清除对特定主机的 IP 业务隔离
clear anti-arpscan attack-list {ip < IP Address > all }	手动清除对特定或所有主机的 ARP 限制
clear anti-arpscan attack-history-list {ip < IP Address > all }	手动清除防 ARP 扫描的历史攻击源信息
特权用户配置模式	
debug anti-arpscan <port ip> no debug anti-arpscan <port ip>	打开或关闭防 ARP 扫描的调试开关。

7) 配置超出阈值之后的动作。

命令	解释
全局配置模式	
anti-arpscan ip-based level2 action<isolate discard-ARP>	超过二级阈值后的处理动作可配置为 IP 业务隔离和丢弃 ARP 报文不送 CPU。
anti-arpscan ip-based arp-to-cpu speed<pps> no anti-arpscan ip-based arp-to-cpu speed	配置一级阈值超限后 ARP 送 CPU 的速率。

3.8.3 防 ARP 扫描典型案例

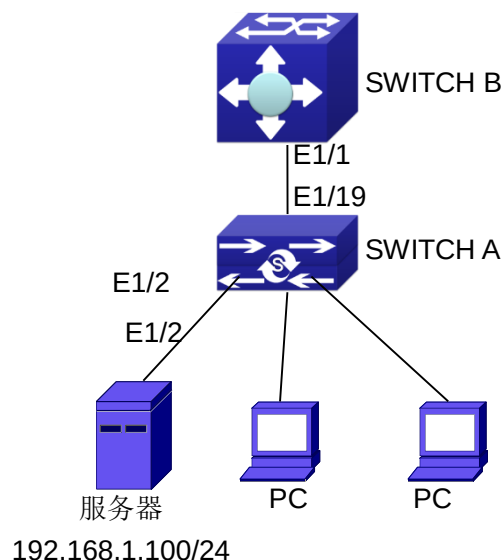


图 3-6 防 ARP 扫描典型配置案例

在上述网络拓扑图中，SWITCH B 的端口 e1/1 与 SWITCH A 的端口 e1/19 相连，SWITCH A 上的端口 e1/2 与文件服务器(IP 地址为 192.168.1.100/24)相连，其他端口都与普通 PC 相连。可通过下面的配置有效地防止 ARP 扫描，而又不影响系统的正常运行。

SWITCH A 配置任务序列：

```
SwitchA(config)#anti-arpscan enable
SwitchA(config)#anti-arpscan recovery time 3600
SwitchA(config)#anti-arpscan trust ip 192.168.1.100 255.255.255.0
SwitchA(config)#interface ethernet1/2
SwitchA (Config-If-Ethernet1/2)#anti-arpscan trust port
SwitchA (Config-If-Ethernet1/2)#exit
SwitchA(config)#interface ethernet1/19
```

```
SwitchA(Config-If-Ethernet1/1)#anti-arpscan trust supertrust-port
```

```
Switch A(Config-If-Ethernet1/1)#exit
```

SWITCH B 配置任务序列:

```
SwitchB(config)#anti-arpscan enable
```

```
SwitchB(config)#interface ethernet1/1
```

```
SwitchB(Config-If-Ethernet1/1)#anti-arpscan trust port
```

```
SwitchB(Config-If-Ethernet1/1)#exit
```

3.8.4 防 ARP 扫描排错帮助

- ☞ 防ARP扫描默认是关闭的。打开防ARP扫描后，可以同时打开调试开关debug anti-arpscan来查看调试信息。

3.9 ARP 绑定

3.9.1 概述

3.9.1.1 ARP（地址解析协议）

简单地说，ARP（RFC-826）协议主要负责将局域网中的 IP 地址转换为对应的 48 位物理地址，即网卡的 MAC 地址，比如 IP 地址为 192.168.0.1 网卡 MAC 地址为 00-03-0F-FD-1D-2B。整个转换过程是一台主机先向目标主机发送包含 IP 地址信息的广播数据包，即 ARP 请求，然后目标主机向该主机发送一个含有 IP 地址和 MAC 地址数据包，通过 MAC 地址两个主机就可以实现数据传输了。

3.9.1.2 ARP 绑定

按照 ARP 协议的设计，为了减少网络上过多的 ARP 数据通信，一台主机，即使收到的 ARP 应答并非自己请求得到的，它也会将其插入到自己的 ARP 缓存表中，这样，就造成了“ARP 绑定”的可能。如果黑客想探听同一网络中两台主机之间的通信（即使是通过交换机相连），他会分别给这两台主机发送一个 ARP 应答包，让两台主机都“误”认为对方的 MAC 地址是第三方即黑客所在的主机，这样，双方看似“直接”的通信连接，实际上都是通过黑客所在的主机间接进行的。黑客一方面得到了想要的通信内容，另一方面，只需要更改数据包中的一些信息，成功地做好转发工作即可。在这种嗅探方式中，黑客所在主机是不需要设置网卡的混杂模式的，因为通信双方的数据包在物理上都是发送给黑客所在的中转主机的。

3.9.1.3 如何进行 ARP 绑定

由于现在网络上充斥着很多基于 ARP 协议的嗅探、监听及攻击行为，而且几乎所有的攻击行为都是基于 ARP 绑定来进行的，所以如何防止 ARP 绑定就显得十分重要。ARP 绑定首先是通过伪造合法 IP 进入正常的网络环境，向交换机发送大量的伪造的 ARP 申请报文，交换机在学习到这些报文后，可能会覆盖原来学习到的正确的 IP、MAC 地址的映射关系，将一些正确的 IP、MAC 地址映射关系修改成攻击报文设置的对应关系，导致交换机在转发报文时出错，从而影响整个网络的运行。或者交换机被恶意攻击者利用，利用错误的 ARP 表，截获交换机转发的报文或者对其它服务器、主机或者网络设备进行攻击。

在网络中防止基于 ARP 的攻击和绑定交换机的方法有两种：

一、关闭交换机的自动更新功能

关闭交换机的自动更新功能以后，当交换机收到 ARP 报文时，如果是新的 ARP 报文（交换机的 ARP 表中不存在该 IP 的表项），则正常学习，这样新的用户可以正常登录网络；如果该 ARP 报文对应的 IP 地址在交换机的 ARP 表中已经存在，则判断 ARP 报文中的 MAC 地址、收到 ARP 报文的端口和交换机 ARP 表中记录的是否相同，不相同则认为是欺骗报文予以丢弃，相同则正常接收，相应的 ARP 表项老化定时器被重置。通过该机制可以防止合法的 ARP 表项被欺骗报文篡改，从而可以避免交换机遭受 ARP 绑定和攻击。

二、关闭交换机的自动学习功能

关闭交换机的自动学习功能以后，交换机不再接收 ARP 报文，适合静态配置 ARP 表项的场合。一方面可以配置静态 ARP 表项，另一方面可以将当前学习到的动态 ARP 表项转换为静态表项（可以减轻一一配置 ARP 静态表项所带来的繁重工作量。关闭交换机的自动学习功能时，如果动态表项不转换为静态表项，会正常老化掉）。在 ARP 表项均为静态配置的情况下，可以有效地避免 ARP 绑定。

3.9.2 ARP 绑定配置

配置 ARP 绑定配置序列如下：

1. 关闭 ARP 自动更新功能
2. 关闭 ARP 自动学习与更新功能
3. 将动态的 ARP 转化为静态 ARP 的功能

1. 关闭 ARP 自动更新功能

命令	解释
全局配置模式和接口配置模式	
ip arp-security updateprotect no ip arp-security updateprotect	关闭和启动 ARP 的自动更新功能。

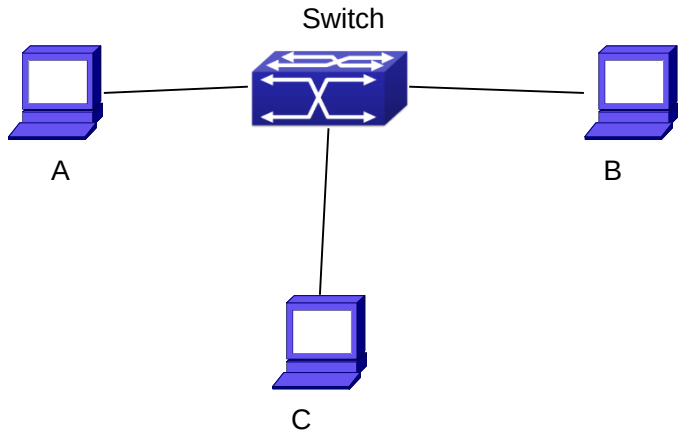
2. 关闭 ARP 自动学习与更新功能

命令	解释
全局配置模式和接口配置模式	
ip arp-security learnprotect no ip arp-security learnprotect	关闭和启动 ARP 的自动学习与更新功能。

3. 将动态的 ARP 转化为静态 ARP 的功能

命令	解释
全局配置模式和接口配置模式	
ip arp-security convert	将动态 ARP 转化静态。

3.9.3 ARP 绑定举例



设备说明：

设备	配置	数量
switch	IP:192.168.2.4; IP:192.168.1.4; mac: 00-00-00-00-00-04	1
A	IP:192.168.2.1; mac: 00-00-00-00-00-01	1
B	IP:192.168.1.2; mac: 00-00-00-00-00-02	1
C	IP:192.168.2.3; mac: 00-00-00-00-00-03	若干

在上图中首先 B 与 C 正常通信。A 想要交换机将 B 发给 C 的报文转发给自己，所以需要交换机将从 B 来的报文发给 A。首先 A 先发送 ARP 应答包给交换机，格式如：192.168.2.3，00-00-00-00-00-01。就是将自己的 MAC 地址换上 C 的 IP，这样交换机在更新 ARP 表时就会将 B 发给 IP 地址：192.168.2.3 的数据报发到 00-00-00-00-00-01 的地址(A 地址)去了。

进一步将 A 也可以将收到的数据报中的源地址，目的地址更改下，让交换机将自己发的包给 C，这样 B 与 C 相互通信的数据在不知情的状况下就都可以被 A 接收。由于 ARP 表是定时更新的，所以 A 还有 1 个任务是持续不断的向交换机发送 ARP 应答包，刷新交换机的

ARP 表。

所以重要的是将 ARP 表保护起来，可以在环境稳定后配置禁止 ARP 学习的命令，然后将所有的动态 ARP 转换为静态的，这样已经学习到的 ARP 就不会被刷新，从而为用户提供保护。

```
Switch(config)#
Switch(Config)#interface vlan 1
Switch(Config-If-Vlan1)#arp 192.168.2.1 00-00-00-00-00-01 interface eth 1/2
Switch(Config-If-Vlan1)#interface vlan 2
Switch(Config-If-Vlan2)#arp 192.168.1.2 00-00-00-00-00-02 interface eth 1/2
Switch(Config-If-Vlan2)#interface vlan 3
Switch(Config-If-Vlan3)#arp 192.168.2.3 00-00-00-00-00-03 interface eth 1/2
Switch(Config-If-Vlan3)#exit
Switch(Config)#ip arp-security learnprotect
Switch(Config)#
Switch(config)#ip arp-security convert
```

如果环境经常变动，也可以开启禁止 ARP 更新的命令，这样一旦学习到的 ARP 属性，就不会被新的 ARP 应答包所更新，保护用户数据不会被“嗅探”。

```
Switch#config
Switch(config)#ip arp-security updateprotect
```

3.10 ARP GUARD

3.10.1 ARP GUARD 的介绍

ARP 协议的设计存在严重的安全漏洞，任何网络设备都可以发送 ARP 报文通告 IP 地址和 MAC 地址的映射关系。这就为 ARP 欺骗提供了可乘之机，攻击者发送 ARP REQUEST 报文或者 ARP REPLY 报文通告错误的 IP 地址和 MAC 地址映射关系，导致网络通讯故障。ARP 欺骗的危害主要表项为两种形式：1、PC4 发送 ARP 报文通告 PC2 的 IP 地址映射为自己的 MAC 地址，将导致本应该发送给 PC2 的 IP 报文全部发送到了 PC4，这样 PC4 就可以监听、截获 PC2 的报文；2、PC4 发送 ARP 报文通告 PC2 的 IP 地址映射为非法的 MAC 地址，将导致 PC2 无法接收到本应该发送给自己的报文。特别是如果攻击者假冒网关进行 ARP 欺骗，将导致整个网络瘫痪。

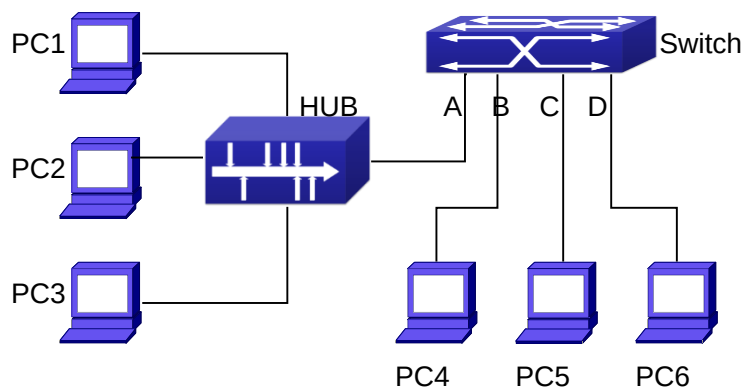


图 3-7 ARP GUARD 原理图

我们利用交换机的过滤表项保护重要网络设备的 ARP 表项不能被其它设备假冒。基本原理就是利用交换机的过滤表项，检测从端口输入的所有 ARP 报文，如果 ARP 报文的源 IP 地址是受到保护的 IP 地址，就直接丢弃报文，不再转发。

ARP GUARD 功能常用于保护网关不被攻击，如果要保护网络内的所有接入 PC 不受 ARP 欺骗攻击，需要在端口配置大量受保护的 ARP GUARD 地址，这将占用大量芯片 FFP 表项资源，可能会因此影响到其它应用功能，并不适合。此时推荐采用 FREE RESOURCE 相关接入方案，详细请参考相关文档。

3.10.2 ARP GUARD 配置任务序列

1. 配置受到保护的 IP 地址

命令	解释
端口配置模式	
arp-guard ip <addr>	配置/删除 ARP GUARD 地址。
no arp-guard ip <addr>	

3.11 Arp local proxy

3.11.1 Arp local proxy 功能简介

某种实际的应用环境中，为了防止 ARP 的欺骗，要求汇聚层的交换机实现 local arp proxy 功能，限制 ARP 报文在同一 vlan 内的转发，从而引导数据流量通过交换机进行 L3 转发。

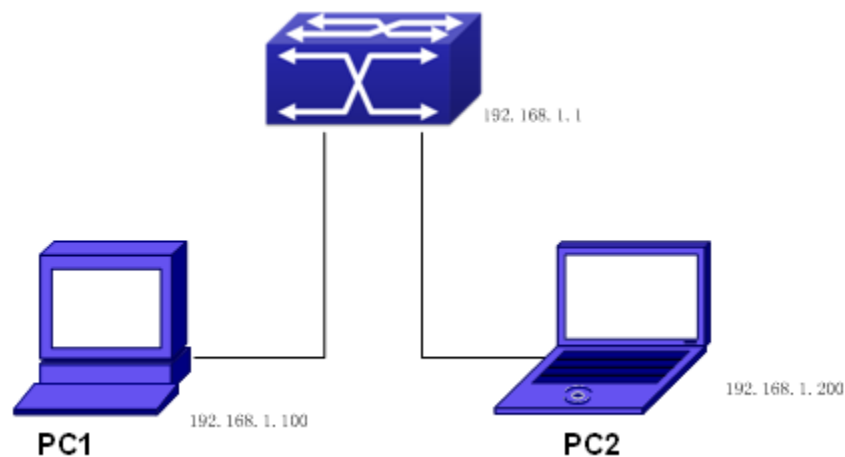


图 3-8 Arp local proxy 功能示意图

例如上图，PC1 要向 PC2 发送一个 IP 报文，整个的流程大致如下（省略了一些非 ARP 的细节）：

1. 由于 PC1 没有 PC2 的 ARP，因此 PC 发送 ARP REQUEST 并广播出去。
2. 交换机硬件收到 ARP 报文，依据新的 ARP 处理规则，芯片不会硬件转发该报文，只是把 ARP REQUEST 送 CPU。
3. 在启动 local arp proxy 的情况下，交换机向 PC1 发送 arp reply 报文（填充自己的 mac 地址）。
4. PC1 收到该 arp reply 之后，创建 ARP，发送 ip 报文，封装的以太网头的目的 MAC 为交换机的 MAC。
5. 当交换机收到该 ip 报文之后，交换机查询路由表（创建路由缓存），并下发硬件表项。
6. 当交换机有 PC2 的 ARP 的情况下，直接封装以太网头部并发送报文（目的 MAC 为 PC2）。
7. 如果交换机没有 PC2 的 ARP，那么会请求 PC2 的 ARP，然后发送 ip 报文。

该功能通常需要和其他的安全功能配合使用，例如在汇聚交换机上配置 local arp proxy，而在下连的二层交换机上配置端口隔离功能，这样将会引导所有的 ip 流量通过汇聚交换机上进行三层转发，而由于端口隔离，ARP 报文不会在 vlan 内转发，其他 PC 也不会收到。

3.11.2 Arp local proxy 功能配置任务序列

1. 启动/关闭 arp local proxy 功能

1. 启动/关闭 arp local proxy 功能

命令	解释
VLAN 接口配置模式	
ip local proxy-arp no ip local proxy-arp	启动或者关闭 arp local proxy 功能。

3.11.3 Arp local proxy 功能典型案例

如下图所示，S1 为支持 arp local proxy 的中高端三层交换机，S2 为支持端口隔离的二层接入交换机。

为了安全的考虑，在 S2 上启动端口隔离功能，这样 S2 的所有下连端口彼此隔离，所有的 ARP 报文都能通过 S1 转发。如果再在 S1 上启动 arp local proxy，则在 S1 上的所有端口隔离 ARP，同时代理 ARP，从而引导 IP 流量经过 S1 进行 3 层转发，而不是原来的 2 层转发。

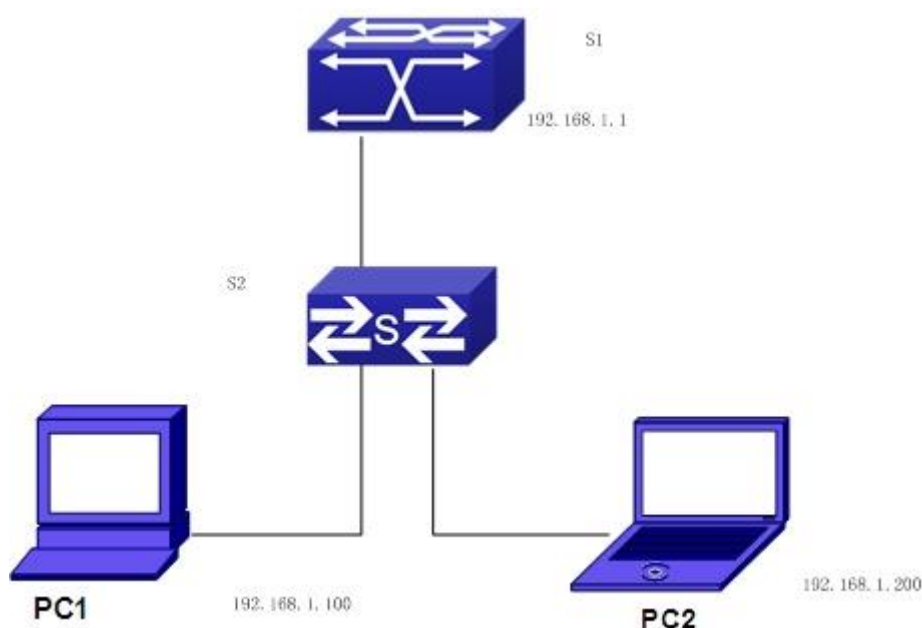


图 3-9 Arp local proxy 功能典型案例示意图

我们可以对 S1 进行如下配置：

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 192.168.1.1 255.255.255.0
```

```
Switch(Config-if-Vlan1)#ip local proxy-arp
```

```
Switch(Config-if-Vlan1)#exit
```

3.11.4 Arp local proxy 功能排错帮助

arp local proxy 功能默认是关闭的，可以使用显示命令（show running-config）看目前的配置情况。在确认配置正确的情况下，打开 ARP 的 debug，可以看到是不是成功代理 ARP，并发送代理 ARP 报文。

在操作过程中，如果有操作错误，系统将会给出具体的错误提示信息。

3.12 免费 ARP 发送

3.12.1 免费 ARP 发送功能简介

主机发送以自己 IP 地址为目标 IP 地址的 ARP 请求，这种 ARP 请求称为免费 ARP (gratuitous ARP)。

交换机基于三层接口的免费 ARP 发送功能的整体思路是：在交换机的接口模式下配置该接口定时发送免费 ARP 报文；或者在交换机的全局配置模式下配置交换机系统中所有存在的接口定时发送免费 ARP 报文。

在本系统中主要实现免费 ARP 的如下两个作用：

- 1. 减少局域网内主机向交换机网关发送 ARP 请求的次数。局域网内主机会每隔一段时间向交换机网关发送 ARP 请求，请求网关的 MAC 地址。如果交换机每隔一段时间向局域网内广播免费 ARP，这样局域网内主机就不用发送 ARP 请求了，从而减少了局域网内主机向网关发送 ARP 请求的次数。
- 2. 可以防止针对网关的 ARP 欺骗。交换机定时向局域网内主机广播免费 ARP，这样局域网内主机的 ARP 缓存会定时更新，所以在局域网内主机受到针对网关的 ARP 欺骗后，主机会在收到交换机的免费 ARP 后，更新其 ARP 缓存而得到正确网关 MAC 地址，这样就防止了针对网关的 ARP 欺骗。

3.12.2 免费 ARP 发送功能配置任务序列

- 1. 启动免费 ARP 发送功能和配置免费 ARP 报文的发送时间间隔
- 2. 显示免费 ARP 发送功能的配置信息

☞ 启动免费 ARP 发送功能和配置免费 ARP 报文的发送时间间隔

命令	解释
全局配置模式和接口配置模式	
ip gratuitous-arp <5-1200> no ip gratuitous-arp	使能免费 ARP 发送功能，并设置发送免费 ARP 报文的发送时间间隔。其中 no 操作取消使能免费 ARP 发送功能。

☞ 显示免费 ARP 发送功能的配置信息

命令	解释
特权和配置模式	

```
show ip gratuitous-arp [interface vlan
<1-4094>]
```

显示免费 ARP 发送功能的配置信息。

3.12.3 免费 ARP 发送功能典型案例

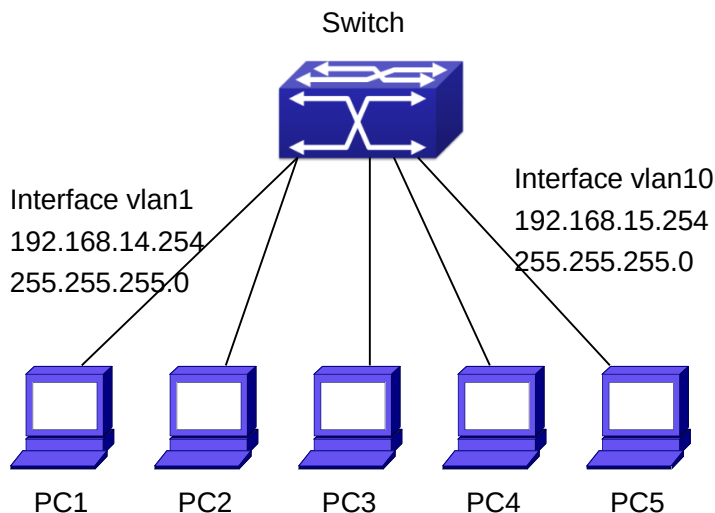


图 3-10 免费 ARP 发送功能配置案例

在上述网络拓扑图中，Switch 系统中的 interface vlan 10（IP 地址为 192.168.15.254，子网掩码为：255.255.255.0）下连接有 3 台 PC 主机（PC3、PC4 和 PC5），interface vlan 1（IP 地址为：192.168.14.254，子网掩码为：255.255.255.0）下连接有两台 PC 主机（PC1 和 PC2）。可通过如下方法配置免费 ARP 发送功能：

1. 一次设置两个接口的免费 ARP 发送功能

```
Switch(config)#ip gratuitous-arp 300
```

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

2. 一次设计一个接口的免费 ARP 发送功能

```
Switch(config)#interface vlan 10
```

```
Switch(Config-if-Vlan10)#ip gratuitous-arp 300
```

```
Switch(Config-if-Vlan10)#exit
```

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

3.12.4 免费 ARP 发送功能排错帮助

免费 ARP 发送功能默认是关闭的。打开免费 ARP 发送功能后，可以打开调试开关 debug arp send 来查看交换机发送的 arp 报文信息。

在全局配置模式下使能免费 ARP 发送功能后，必须在全局配置模式才能取消使能免费 ARP 发送功能；在接口配置模式下使能免费 ARP 发送功能后，必须在接口配置模式下才能取消使能免费 ARP 发送功能。

3.13 Keepalive gateway

3.13.1 keepalive gateway 介绍

市场需要使用以太端口参与备份或者负载均衡，由于以太端口是广播式信道，在网关 down 的情况下由于检测不到物理信号的变化，所以以太端口无法 down 下来。该功能主要用在网关使用以太端口上连上一级网关形成点到点的网络拓扑的情况下，希望可以检测到上一级网关的连通性。

例如：路由器连接光端机，路由器到光端机的线路始终是 up 的，而 moden 与远端设备之间的线路不通的时候环境下，需要使用有效手段检测对端设备是否可达。目前，使用 ARP 机制来检测网关的连通性，通过定时的给网关发送 ARP 请求来探测网关的连通性，如果 ARP 解析失败，表明网关已经 down，这时 down 掉三层接口；继续定时的发送 ARP 请求，如果 ARP 解析成功，则 up 接口。

该功能只对三层交换机支持，二层交换机不支持。

3.13.2 keepalive gateway 功能配置任务序列

1. 开启或关闭 keepalive gateway 功能和配置发送 ARP 请求报文的间隔周期和判定探测失败的重试次数
2. 显示接口 keepalive gateway 运行情况和接口 ipv4 运行状态

1. 启动 keepalive gateway 功能和配置发送 ARP 请求报文的间隔周期和判定探测失败的重试次数

命令	解释
接口配置模式	
keepalive gateway <ip-address> [{<interval-seconds> msec <interval-millisecond >}] [retry-count]	使能 keepalive gateway 功能，配置探测的网关 IP 地址、发送 ARP 请求报文的间隔周期和判定探测失败的重试次数；no 命令为关闭该功能。
no keepalive gateway	

2. 显示接口 keepalive gateway 运行情况和接口 ipv4 运行状态

命令	解释
特权和配置模式	
show keepalive gateway [interface-name]	显示指定接口的 keepalive 运行情况。如果不指定则显示所有接口的 keepalive 运行情况。
show ip interface [interface-name]	显示指定接口的 IPv4 运行状态。如果不指定则显示所有接口的 IPv4 运行状态。

3.13.3 keepalive gateway 举例

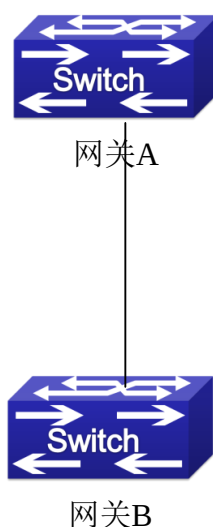


图 3-11 keepalive gateway 典型应用举例

在上述网络拓扑中，网关 A interface vlan10 接口地址为 1.1.1.1 255.255.255.0，网关 B interface vlan100 接口地址为 1.1.1.2 255.255.255.0，网关 B 支持 keepalive gate 功能，可通过如下方法配置网关 B 上的 keepalive gateway 功能：

- 1、采用默认的发送 arp 间隔和判定探测失败重复次数（缺省发送 arp 探测报文周期为 10s，缺省判定探测失败重复次数为 5 次）

```
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#keepalive gateway 1.1.1.1
Switch(config-if-vlan100)#exit
```

- 2、手动配置发送 arp 间隔和判定探测失败重复次数

```
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#keepalive gateway 1.1.1.1 3 3
Switch(config-if-vlan100)#exit
```

探测网关 A 是否可达，每隔 3 秒发送一次 ARP 探测，探测 3 次失败后认为网关 A 不可达。

3.13.4 keepalive gateway 排错帮助

当在使用 keepalive gateway 过程中遇到问题，请检查是否是如下原因：

- ☞ 确认设备为三层交换机，二层交换机不支持该功能。
- ☞ 该种探测方法仅适用于点对点的拓扑模式。
- ☞ 该方法是探测的 IPv4 可达性。因此探测结果，只能影响接口的 IPv4 工作状态。当探测失败时，只能禁止接口上不再运行 IPv4 业务；但 IPv6 等其它业务不受该探测的影响。
- ☞ 接口的物理状态仅受物理信号的控制，不受该探测功能的控制。
- ☞ 当判定网关不可达后，接口不能运行 IPv4 业务。因此所有同该接口相关的 IPv4 路由被删除，IPv4 路由协议不能在该接口上建立邻居。

3.14 UFT

3.14.1 UFT 功能介绍

交换机的数据转发依赖于交换芯片上存在的各种硬件转发表项，一般情况下交换芯片中存在MAC地址表、ARP表以及路由表这三种硬件表。由于硬件资源是有限的，这些硬件资源表初始时会由系统按照预先设定分配固定大小的表项数量，一旦表项数量确定后续就不能再进行变更。

UFT（Unified Forwarding Table）简称联合转发表，是指在交换芯片中除上述提到的三种硬件转发表外，新增的共享空间，该空间可以根据需要来扩充MAC地址表、ARP表以及路由表。相比传统的固定表项空间，UFT功能可以根据不同的业务需要，灵活调整硬件表项资源大小，更好的服务于各种业务。

UFT功能提供如下几种工作模式：

Bridge模式：增强桥模式。此模式下MAC容量大，适用于网络中存在大量MAC的场景；

Routee模式：增强路由模式。此模式下路由容量大，适用于网络中存在大量路由的场景；

Standard模式：标准模式。此模式下ARP容量大，适用于网络中存在大量主机的场景。

3.14.2 UFT 配置任务序列

1. 配置 UFT 工作模式（可选，重启后生效）
2. 查看 UFT 工作模式

1. 配置 UFT 工作模式

UFT 默认工作模式为 standard 模式，修改 UFT 工作模式后需要保存配置重启才能生效。如果只修改指定线卡的工作模式，只需要重启对应的线卡即可。

命令	解释
全局配置模式	

System working mode {bridge route standard word} [member <member-id>{slot<slot-id>}]	修改指定线卡 UFT 的工作模式。 独立模式下使用 slot 参数来指定线卡，堆叠模式下使用 member 和 slot 参数来指定线卡。 （Route 模式的路由条目数暂不支持 128K。）

2. 查看 UFT 工作模式

命令	解释
全局配置模式	
Show system working mode {config current support}	查看当前 UFT 配置的模式、当前使用的模式以及所支持的模式。（修改 UFT 工作模式后需要重启线卡，否则会出现配置模式与当前使用模式不一致的情况）

3.14.3 UFT 典型案例

案例一：

独立模式下，修改 slot 2 的 UFT 工作模式为 bridge 模式。

操作步骤如下：

Step1: 修改线卡 2 的 UFT 工作模式

```
switch(config)#system working mode bridge slot 2
```

Please save the configuration, it will take effect after the next system restart!

Step2: 保存配置，重启线卡 2

```
switch#write
```

Confirm to overwrite current startup-config configuration [Y/N]:y

Write running-config to current startup-config successful

```
switch#reset slot 2
```

Step3: 查看当前 UFT 工作模式

```
switch#show system working mode current
```

```
-----slot: 2 -----
```

```
current system working mode: mode 1(bridge)
```

```
mode 1(bridge): MAC=288K, IP=16K, ROUTER=8K
```

案例二：

堆叠模式下，修改 member2 上 slot2 的 UFT 工作模式为 bridge 模式

操作步骤如下：

Step1: 修改 member2 线卡 2 的 UFT 工作模式

```
switch(config)#system working mode bridge member 2 slot 2
```

Please save the configuration, it will take effect after the next system restart!

Step2: 保存配置，重启 member2 线卡 2

```
switch#write
```

```
Confirm to overwrite current startup-config configuration [Y/N]:y
```

```
Write running-config to current startup-config successful
```

```
switch#reset member 2 slot 2
```

Step3: 查看当前 UFT 工作模式

```
switch#show system working mode current
```

```
-----slot: member 2, slot 2 -----
```

```
current system working mode: mode 1(bridge)
```

```
mode 1(bridge): MAC=288K, IP=16K, ROUTER=8K
```

3.14.4 UFT 排错帮助

- ☞ 确认线卡支持 UFT 功能，可通过 `show system working mode support` 查看。
- ☞ 修改线卡 UFT 工作模式后，需要重启才能生效，否则按照默认 `standard` 模式工作。
- ☞ 受芯片硬件资源制约，各板卡即使工作在同一种模式，其表项数量也会存在差异，各表项支持数量可以通过 `show system working mode support` 查看。
- ☞ IP 表项的数量同时受到芯片其它资源和具体应用的影响，实际可使用的表项数量会有些差别。

3.15 DHCP

3.15.1 DHCP 介绍

DHCP[RFC2131]是 Dynamic Host Configuration Protocol 动态主机配置协议的简写，它能从地址池中把 IP 地址动态分配给请求的主机，同时也能够提供其它网络配置参数，如缺省网关、DNS 服务器、域名和网络范围内主机映像文件的位置等。DHCP 是 BOOTP 协议功能的增强，与 BOOTP 相比，DHCP 是主流技术，它不仅能为无盘工作站提供引导信息，而且在大型的网络中可以大大减轻网络管理员跟踪记录手工分配 IP 地址的负担，同时也能减轻用户的配置任务和花费。DHCP 的另外一个优点是可以部分缓解 IP 地址紧张的状况，当某一 IP 地址的用户离开使用环境时，该 IP 地址还能再次分配给其他用户使用。

DHCP 是基于 Client—Server 模式的协议，DHCP 客户机向 DHCP 服务器索取网络地址及配置参数；服务器为客户机提供网络地址及配置参数；当 DHCP 客户机和 DHCP 服务器不在同一子网时，需要由 DHCP 中继为 DHCP 客户机和 DHCP 服务器传递 DHCP 报文。协议实现的过程如下：

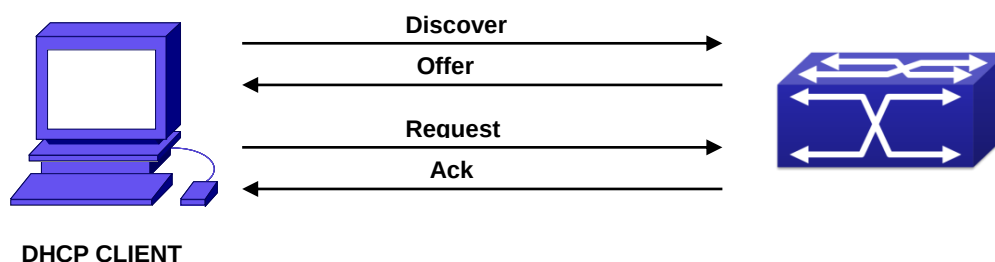


图 3-12 DHCP 协议交互过程

图解：

1. 首先 DHCP 客户机在本子网内广播 DHCPDISCOVER 包；
2. DHCP 服务器收到 DHCPDISCOVER 包后，给该 DHCP 客户机发送带有 IP 地址和其他网络参数的 DHCPOFFER 包；
3. DHCP 客户机对收到的 DHCPOFFER 包进行选择后，广播带有它要选择的 DHCP 服务器的信息的 DHCPREQUEST 包；
4. 被 DHCP 客户机选中的 DHCP 服务器向 DHCP 客户机发送 DHCPACK 包，DHCP 客户机得到 IP 地址和其他网络配置参数。

通过上面四个步骤，完成动态分配主机配置的协议过程。但如果 DHCP 服务器和 DHCP 客户机不在同一网络时，服务器是无法收到客户机发出的 DHCP 广播报文，因此服务器也不会给客户机发送任何 DHCP 报文，这时需要 DHCP 中继来转发这些 DHCP 报文，完成 DHCP 客户机和服务器之间的 DHCP 报文交互过程。

交换机实现了 DHCP 服务器和 DHCP 中继的功能。DHCP 服务器不但支持动态分配 IP 地址，还支持手工绑定 IP 地址（即为指定的硬件地址或者指定设备标识的网络设备分配一个固定的长期的 IP 地址）。动态分配 IP 地址和手工绑定 IP 地址的区别和联系是：1)采用动态方式获得的 IP 地址可以是不固定的；而采用手工绑定方式获得的 IP 地址是固定的；2)采用动态方式获得的 IP 地址租期与其地址池的租期一致，是有时间限制的；而采用手工绑定方式获得的 IP 地址的租期理论上是无限长时间，即没有时间限制；3)已经动态分配出去的地址，不允许再手工绑定；4)手工 DHCP 地址池可以继承相关网段的动态 DHCP 地址池的网络配置参数。

3.15.2 DHCP 服务器配置

DHCP 服务器配置任务序列如下：

1. 启动/关闭 DHCP 服务功能
2. 配置 DHCP 地址池
 - (1) 创建/删除 DHCP 地址池
 - (2) 配置动态 DHCP 地址池的参数
 - (3) 配置手工 DHCP 地址池的参数

3. 启动记录地址冲突的日志功能

1. 启动/关闭 DHCP 服务功能

命令	解释
全局配置模式	
service dhcp no service dhcp	启动 DHCP 服务器或中继功能。本命令的 no 操作关闭 DHCP 服务器或中继功能。

2. 配置 DHCP 地址池

(1) 创建/删除 DHCP 地址池

命令	解释
全局配置模式	
ip dhcp pool <name> no ip dhcp pool <name>	配置 DHCP 地址池。本命令的 no 操作删除 DHCP 地址池。

(2) 配置动态 DHCP 地址池的参数

命令	解释
DHCP 地址池配置模式	
network-address <network-number> [mask prefix-length] no network-address	配置地址池可分配的地址范围。本命令的 no 操作删除地址池可分配的地址。
default-router [address1[address2[...address8]]] no default-router	为 DHCP 客户机配置缺省网关。本命令的 no 操作删除缺省网关。
dns-server [address1[address2[...address8]]] no dns-server	为 DHCP 客户机配置 DNS 服务器。本命令的 no 操作删除配置的 DNS 服务器。
domain-name <domain> no domain-name	为 DHCP 客户机配置域名；本命令的 no 操作删除域名。
netbios-name-server [address1[address2[...address8]]] no netbios-name-server	配置 Wins 服务器的地址。本命令的 no 操作删除服务器的地址。
netbios-node-type {b-node h-node m-node p-node <type -number>} no netbios-node-type	配置 DHCP 客户机的节点类型。本命令的 no 操作删除 DHCP 客户机的节点类型。

bootfile <filename> no bootfile	配置 DHCP 客户机的启动时的导入文件名。 本命令的 no 操作删除此操作。
next-server [address1[address2[...address8]]] no next-server [address1[address2[...address8]]]	配置客户机导入文件存放的服务器地址。本命令的 no 操作删除客户机导入文件存放的服务器地址。
option <code> {ascii <string> hex <hex> ipaddress <ipaddress>} no option <code>	配置 option 所指定代码的网络参数的值。 本命令的 no 操作删除 option 所指定代码的网络参数值。
lease { days [hours][minutes] infinite } no lease	配置地址池中地址的租用期限。本命令的 no 操作删除地址池中地址的租用期限。
max-lease-time {[<days>] [<hours>] [<minutes>] infinite } no max-lease-time	配置地址池中地址的最大租用期限；本命令的 no 操作为恢复缺省值。
全局配置模式	
ip dhcp excluded-address <low-address> [<high-address>] no ip dhcp excluded-address <low-address> [<high-address>]	排除地址池中的不用于动态分配的地址。本命令的 no 操作删除此操作。

(3) 配置手工 DHCP 地址池的参数

命令	解释
DHCP 地址池配置模式	
hardware-address <hardware-address> [{Ethernet IEEE802}<type-number>] no hardware-address	在手工分配地址时，指定/删除用户的硬件地址。
host <address> [<mask> <prefix-length>] no host	在手工绑定地址时，配置/删除分配给指定客户机的用户的 IP 地址。
client-identifier <unique-identifier> no client-identifier	在手工绑定地址时，指定/删除用户的唯一标识。

3. 启动记录地址冲突的日志功能

命令	解释
全局配置模式	

ip dhcp conflict logging	打开/关闭 DHCP 服务器检测地址冲突的日志功能。
no ip dhcp conflict logging	
特权用户配置模式	
clear ip dhcp conflict <address all>	删除单条地址冲突的记录或全部地址的冲突记录。

3.15.3 DHCP 中继配置

当 DHCP 客户机和 DHCP 服务器不在同一个网段时,由 DHCP 中继传递 DHCP 报文。增加 DHCP 中继功能的好处是不必为每个网段都设置 DHCP 服务器,同一个 DHCP 服务器可以为很多个子网的客户机提供网络配置参数,既节约了成本又方便了管理。

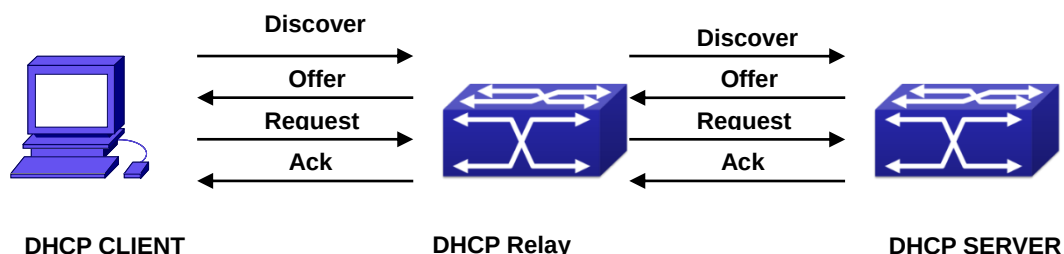


图 3-13 DHCP 中继

如上图所示, DHCP 客户机和 DHCP 服务器不在一个网络内, DHCP 客户机仍然遵循 DHCP 的四个步骤,但增加了 DHCP 中继的转发功能:

1. 首先广播 DHCPDISCOVER 报文, DHCP 中继接收到客户机广播的 DHCPDISCOVER 文后,在该报文的中继代理字段处加入自己的 IP 地址后转发给指定的 DHCP 服务器(有关 DHCP 帧格式可参看 RFC2131);
2. DHCP 服务器接收到经过 DHCP 中继转发的 DHCPDISCOVER 后,带有网络配置参数的 DHCPOFFER 报文经过 DHCP 中继发送给 DHCP 客户机;
3. DHCP 客户机选中一个 DHCP 服务器,广播 DHCPREQUEST 报文, DHCP 中继处理该报文后转发给 DHCP 服务器;
4. DHCP 服务器收到 DHCPREQUEST 后,回应 DHCPACK 报文经过 DHCP 中继发送给 DHCP 客户机。

DHCP 中继配置任务序列如下:

1. 启动 DHCP 中继
2. 配置 DHCP 中继转发 DHCP 广播报文

1.启动 DHCP 中继

命令	解释
全局配置模式	

service dhcp	在启动 DHCP 服务器功能的同时也启动了 DHCP 中继功能。
no service dhcp	

2. 配置 DHCP 中继转发 DHCP 广播报文

命令	解释
全局配置模式	
ip forward-protocol udp bootps no ip forward-protocol udp bootps	中继转发 UDP 端口号为 67 的 DHCP 广播报文。
接口配置模式	
ip helper-address <ipaddress> no ip helper-address <ipaddress>	指定 DHCP 中继转发的目标 IP 地址；本命令的 no 操作为取消该项配置。

3.15.4 DHCP 配置举例

案例 1:

为减轻网络管理员和用户的配置负担，现有某公司将交换机作为 DHCP 服务器。其 Admin VLAN 的 IP 地址为 10.16.1.2/24。其中公司局域网因为办公地点分成了 A 地、B 地两部分，A 地、B 地的网络配置如下表。

PoolA(network 10.16.1.0)		PoolB(network 10.16.2.0)	
设备	IP 地址	设备	IP 地址
缺省网关	10.16.1.200	缺省网关	10.16.2.200
	10.16.1.201		10.16.2.201
DNS 服务器	10.16.1.202	DNS 服务器	10.16.2.202
Wins 服务器	10.16.1.209	WWW 服务器	10.16.2.209
Wins 的节点类型	H-node		
Lease	3 天	Lease	1 天

其中在 A 处，因为工作的需要，特地将一台 MAC 地址为 00-03-22-23-dc-ab 的机器分配固定的 IP 地址 10.16.1.210，命名为 management。

```
Switch(config)#service dhcp
Switch(config)#interface vlan 1
Switch(Config-if-Vlan-1)#ip address 10.16.1.2 255.255.255.0
Switch(Config-if-Vlan-1)#exit
Switch(config)#ip dhcp pool A
Switch(dhcp-A-config)#network 10.16.1.0 24
Switch(dhcp-A-config)#lease 3
Switch(dhcp-A-config)#default-router 10.16.1.200 10.16.1.201
Switch(dhcp-A-config)#dns-server 10.16.1.202
```

```
Switch(dhcp-A-config)#netbios-name-server 10.16.1.209
Switch(dhcp-A-config)#netbios-node-type H-node
Switch(dhcp-A-config)#exit
Switch(config)#ip dhcp excluded-address 10.16.1.200 10.16.1.201
Switch(config)#ip dhcp pool B
Switch(dhcp-B-config)#network 10.16.2.0 24
Switch(dhcp-B-config)#lease 1
Switch(dhcp-B-config)#default-router 10.16.2.200 10.16.2.201
Switch(dhcp-B-config)#dns-server 10.16.2.202
Switch(dhcp-B-config)#option 72 ip 10.16.2.209
Switch(dhcp-config)#exit
Switch(config)#ip dhcp excluded-address 10.16.2.200 10.16.2.201
Switch(config)#ip dhcp pool A1
Switch(dhcp-A1-config)#host 10.16.1.210
Switch(dhcp-A1-config)#hardware-address 00-03-22-33-dc-ab
Switch(dhcp-A1-config)#exit
```

使用提示：

当有 DHCP/BOOTP Client 连接在交换机的属于 VLAN1 端口时，该 Client 只能获取属于 10.16.1.0/24 网段的地址而不可能获取 10.16.2.0/24 网段的地址。因为 Client 发出的广播包经过交换机的 VLAN 接口转发后，申请的是与 VLAN 接口在一个网段的 IP 地址，交换机的 VLAN 接口的 IP 地址为 10.16.1.2/24，因此 Client 申请到的 IP 地址是 10.16.1.0/24 网段的。若 DHCP/BOOTP Client 希望申请到 10.16.2.0/24 网段的地址，该 Client 发出的广播包经由转发的网关必须是 10.16.2.0/24 网段的。若要从交换机获取 10.16.2.0/24 地址池的 IP 地址，则首先要保证该 Client 的网关到交换机的可达性。

案例 2：

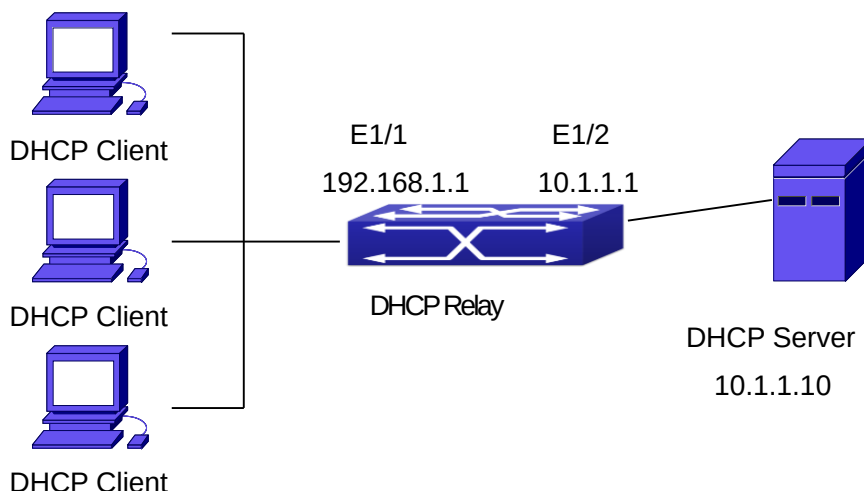


图 3-14 DHCP Relay 配置

如图所示，配置 Switch 为 DHCP 中继。DHCP 服务器的地址为 10.1.1.10，则配置如下：

```
Switch(config)#service dhcp
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 192.168.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#vlan 2
Switch(Config-Vlan2)#exit
Switch(config)#interface Ethernet 1/2
Switch(Config-If-Ethernet1/2)#switchport access vlan 2
Switch(Config-If-Ethernet1/2)#exit
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 10.1.1.1 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(Config)#ip forward-protocol udp bootps
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip helper-address 10.1.1.10
Switch(Config-if-Vlan1)#exit
```

注意: 建议配置命令 **ip forward-protocol udp bootps** 和 **ip helper-address <ipaddress>** 结合使用。**ip helper-address** 只能在三层端口下配置，不能直接在二层端口下配置。

3.15.5 DHCP 排错帮助

DHCP 客户机无法获得 IP 地址及其它网络参数，在确保 DHCP 客户机硬件、线缆等没有问题的前提下，检查可能存在的情况和建议的解决方法：

- ☞ 首先检查 DHCP 服务器是否已启动，若没有启动，建议启动相关的 DHCP 服务器；若 DHCP 客户机和服务器不在同一个物理网络中，检查中间负责转发 DHCP 报文的路由器是否具备 DHCP 中继功能。若中间路由器不具备 DHCP 中继功能，建议替换掉中间的路由器或更新版本，使其具备 DHCP 中继功能；
- ☞ 用户比较容易遇到的现象是 DHCP 客户机连接在交换机上，却获得不到 IP 地址。遇到这种情况请检测 DHCP Server 内是否有与交换机 VLAN 接口在一个网段的地址池，如没有则请添加该网段的地址池；
- ☞ 在 DHCP 服务中，动态分配 IP 地址与手工分配 IP 地址的地址池是互斥的，即如果在一个地址池执行命令 **network** 和命令 **host** 时，只能有一个生效；并且在手工地址池中一个地址池内只能配置一对 IP-MAC 的绑定，如果需要建立多对绑定，可以建立多个手工地址池，在每个地址池分别配置 IP-MAC 的绑定，否则在同一地址池内配置新的配置会覆盖旧的配置。

3.16 DHCP option 82

3.16.1 DHCP option 82 介绍

DHCP option 82 是 DHCP 报文中的中继代理信息选项（Relay Agent Information Option），其选项编号为 82。DHCP option 82 是为了增强 DHCP 服务器的安全性，改善 IP 地址配置策略而提出的一种机制。通过在网络接入设备上配置 DHCP 中继代理功能，中继代理把从客户端接收到的 DHCP 请求报文添加进 option 82 选项（其中包含了客户端的接入物理端口和接入设备标识等信息），然后再把该报文转发给 DHCP 服务器，支持 option 82 功能的 DHCP 服务器接收到报文后，根据预先配置策略和报文中 option 82 信息分配 IP 地址和其它配置信息给客户端，同时 DHCP 服务器也可以依据 option 82 中的信息识别可能的 DHCP 攻击报文并作出防范。DHCP 中继代理收到服务器应答报文后，剥离其中的 option 82 选项并根据选项中的物理端口信息，把应答报文转交到网络接入设备的指定端口。DHCP option82 的应用对客户端是透明的。

3.16.1.1 DHCP option 82 报文结构

DHCP 报文可以由多个 option 字段组成，option 82 就是其中的一种 option，它必须在其它选项之后，option255 选项之前。其格式如下图所示：

Code	Len	Agent Information Field					
82	N	i1	i2	i3	i4	...	iN

Code: 表示中继代理信息选项的序号, RFC3046 定义为 82, option 82 即由此得名。

Len: 为代理信息域 (Agent Information Field) 的字节个数, 不包括 Code 和 Len 字段的两个字节。

option 82 可以由多个 sub-option 组成, 每个 option 82 选项至少要有一个子选项, RFC3046 定义了以下两个子选项, 其格式如下图所示:

SubOpt	Len	Sub-option Value					
1	N	s1	s2	s3	s4	...	sN

SubOpt	Len	Sub-option Value					
2	N	i1	i2	i3	i4	...	iN

SubOpt: 为子选项编号, 其中代理电路 ID (即 Circuit ID) 子选项编号为 1, 代理远程 ID (即 Remote ID) 子选项编号为 2。

Len: 为 Sub-option Value 的字节个数, 不包括 SubOpt 和 Len 字段的两个字节。

3.16.1.2 option 82 工作机制

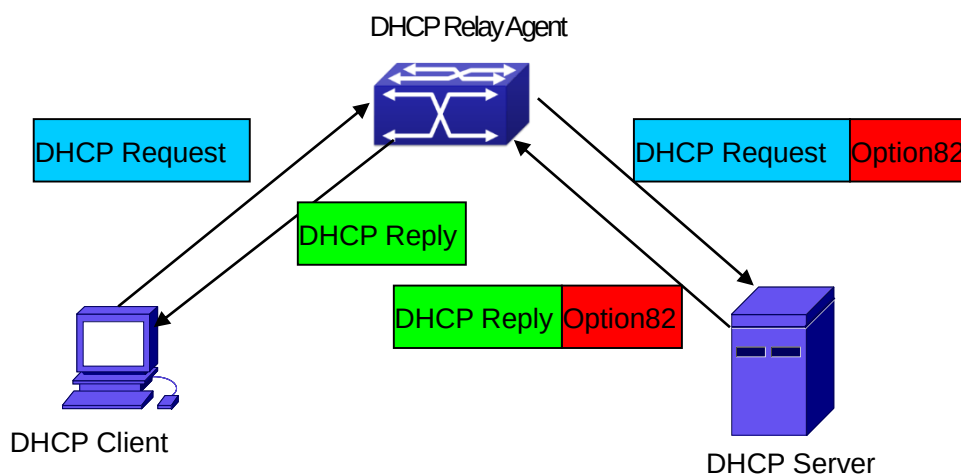


图 3-15 DHCP option 82 流程图

在 DHCP 中继代理支持 option 82 的情况下, DHCP 客户端通过 DHCP 中继从 DHCP 服务器获取 IP 地址同样要经历发现、提供、选择和确认四个阶段。这时 DHCP 协议按如下过程进行:

1) DHCP 客户端在初始化时广播发送请求报文, 这时的请求报文并不包含 option 82 选项。

2) DHCP 中继代理将 option 82 选项添加到接收到的请求报文尾部后中继转发给 DHCP 服务器。option 82 选项的子选项 1(代理电路 ID)默认是 DHCP 客户端所连接的交换机的接口信息(VLan 名加物理端口名),也可以由用户自己配置代理电路 ID, option 82 选项的子选项 2(代理远程 ID)是 DHCP 中继设备本身的 MAC 地址。

3) DHCP 服务器收到 DHCP 中继设备转发的 DHCP 请求报文后,根据报文中 option 选项所携带的信息和预定策略分配 IP 地址和其它信息给客户端,然后将带着 DHCP 配置信息以及 option 82 信息的应答报文发给 DHCP 中继代理。

4) DHCP 中继代理收到 DHCP 服务器的应答报文后将剥离报文中的 option 82 信息,然后将带有 DHCP 配置信息的报文转发给 DHCP 客户端。

3.16.2 DHCP option 82 的配置任务序列

与 DHCP option 82 相关的配置内容

- 1) 配置中继代理的 DHCP option 82 使能开关
- 2) 配置接口的 DHCP option 82 属性
- 3) 配置服务器的 DHCP option 82 使能开关
- 4) 配置中继代理的 DHCP option 82 默认格式
- 5) 配置分隔符
- 6) 配置 option82 的生成方式
- 7) DHCP option 82 的维护与诊断

1) 配置中继代理的 DHCP option 82 使能开关

命令	解释
全局配置模式	
ip dhcp relay information option no ip dhcp relay information option	设置本命令启用交换机中继代理的 option82 功能, 本命令的 no 操作关闭交换机中继代理的 option82 功能。

2) 配置接口的 DHCP option 82 属性

命令	解释
接口配置模式	

ip dhcp relay information policy {drop keep replace} no ip dhcp relay information policy	本命令用于设置系统对于接收到的含有 option82 选项的 DHCP 请求报文的重转发策略，其中 drop 模式表示如果报文中含有 option82 选项，则系统丢弃该 DHCP 报文不作处理；keep 模式表示系统保持现有报文中的 option82 选项不变转发给服务器处理；replace 模式表示系统使用自己的 option82 选项替换现有报文中的 option82 选项，然后转发给服务器处理。本命令的 no 操作设置 option82 选项 DHCP 报文的重转发策略为 replace。
ip dhcp relay information option subscriber-id {standard <circuit-id>} no ip dhcp relay information option subscriber-id	本命令用于设置从接口接收的 DHCP 请求报文添加 option 82 子选项 1(电路 ID 选项)的形式， standard 表示标准的 vlan 名加物理端口名形式，如 “Vlan2+Ethernet1/12”，<circuit-id>为用户自己指定的 option 82 的 circuit-id 内容，它是一个长度小于 64 的字符串。本命令的 no 操作把添加 option 82 子选项 1(电路 ID 选项)的形式设置为 standard 形式。
全局配置模式	
ip dhcp relay information option remote-id {standard <remote-id>} no ip dhcp relay information option remote-id	本命令用于设置从接口接收的 DHCP 请求报文添加 option 82 子选项 2(远程 ID 选项)的具体内容。本命令的 no 操作把添加 option 82 子选项 2(远程 ID 选项)的形式设置为 standard。

3) 配置服务器的 DHCP option 82 使能开关

命令	解释
全局配置模式	
ip dhcp server relay information enable no ip dhcp server relay information enable	本命令用于设置交换机 DHCP 服务器支持对 option82 选项的识别。本命令的 no 操作使服务器忽略处理 option 82 选项。

4) 配置中继代理的 DHCP option 82 默认格式

命令	解释
全局配置模式	

ip dhcp relay information option subscriber-id format {hex acsii vs-hp}	本命令设置交换机中继代理的 option82 功能 subscriber-id 默认格式。
ip dhcp relay information option remote-id format {default vs-hp}	本命令设置交换机中继代理的 option82 功能 remote-id 默认格式。

5) 配置分隔符

命令	解释
全局配置模式	
ip dhcp relay information option delimiter [colon dot slash space] no ip dhcp relay information option delimiter	配置用户在全局定义的用来生成 option82 子选项的各参数的分隔符，该命令的 no 形式恢复分隔符为默认值，即 slash。

6) 配置 option82 的生成方式

命令	解释
全局配置模式	
ip dhcp relay information option self-defined remote-id {hostname mac string WORD} no ip dhcp relay information option self-defined remote-id	配置 option82 的生成方式，用户可以自定义用来生成 option82 子选项 remote-id 的参数（集）。
ip dhcp relay information option self-defined remote-id format [ascii hex]	配置 relay option82 中 remote-id 的生成格式。
ip dhcp relay information option self-defined subscriber-id {vlan port id (switch-id (mac hostname) remote-mac) string WORD } no ip dhcp relay information option self-defined subscriber-id	配置 option82 的生成方式，用户可以自定义用来生成 option82 子选项 circuit-id 的参数（集）。
ip dhcp relay information option self-defined subscriber-id format [ascii hex]	配置 relay option82 中 circuit-id 的生成格式。

7) DHCP option 82 的维护与诊断

命令	解释
特权配置模式	

show ip dhcp relay information option	本命令显示系统 DHCP option 82 的状态信息，包括 option82 使能开关，接口重转发策略，接口电路 ID 模式，以及交换机 DHCP 服务器 option82 使能开关。
debug ip dhcp relay packet	使用本命令显示 DHCP 中继代理处理数据包的信息，包括 option 82 选项的添加和剥离动作信息。

3.16.3 DHCP option 82 应用举例

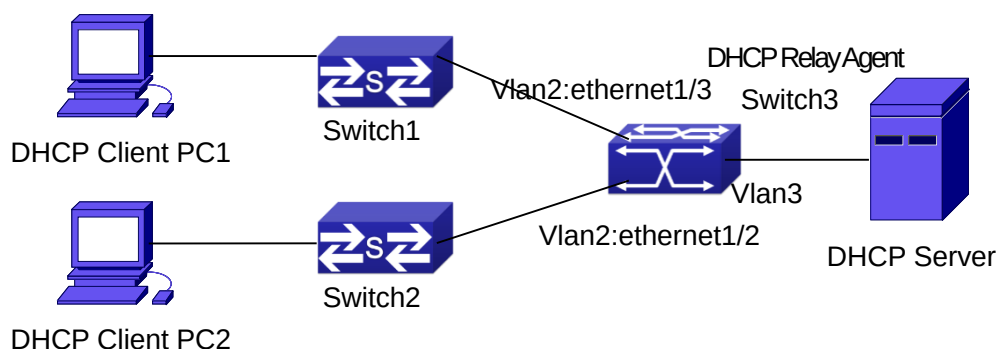


图 3-16 DHCP option 82 典型应用案例

在上面应用案例中，二层交换机 Switch1 和 Switch2 都接在三层交换机 Switch3 上，Switch3 作为 DHCP 中继代理向 DHCP 服务器转交来自 DHCP 客户端的请求报文，并把服务器的应答报文转交给 DHCP 客户端，完成 DHCP 协议过程。在不启用 DHCP option 82 功能时，DHCP 服务器无法区分 DHCP 客户端是来自 Switch1 还是 Switch2 下连的网络，因此 Switch1 和 Switch2 连接的 PC 终端都从 DHCP 服务器中公用地址池中分配地址。启用 DHCP option 82 功能以后，由于 Switch3 在来自客户端的请求报文中附加了接入 Switch3 的端口信息，这时服务器可以区分客户端是来自 Switch1 还是 Switch2 的网络，因此可以为这两个网络分配互相隔离的地址空间，便于网络管理。

Switch3(MAC 地址为 00:03:0f:02:33:01)配置如下

```
Switch3(Config)#service dhcp
Switch3(Config)#ip dhcp relay information option
Switch3(Config)#ip forward-protocol udp bootps
Switch3(Config)#interface vlan 3
Switch3(Config-if-vlan3)#ip address 192.168.10.222 255.255.255.0
Switch3(Config-if-vlan2)#ip address 192.168.102.2 255.255.255.0
Switch3(Config-if-vlan2)#ip helper 192.168.10.88
```

Linux 的 ISC DHCP Server 支持 option 82 选项，其配置文件/etc/dhcpd.conf 为

```
ddns-update-style interim;  
ignore client-updates;
```

```
class "Switch3Vlan2Class1" {  
match if option agent.circuit-id = "Vlan2+Ethernet1/2" and option  
agent.remote-id=00:03:0f:02:33:01;  
}
```

```
class "Switch3Vlan2Class2" {  
match if option agent.circuit-id = "Vlan2+Ethernet1/3" and option  
agent.remote-id=00:03:0f:02:33:01;  
}
```

```
subnet 192.168.102.0 netmask 255.255.255.0 {  
option routers 192.168.102.2;  
option subnet-mask 255.255.255.0;  
option domain-name "example.com.cn";  
option domain-name-servers 192.168.10.3;  
authoritative;
```

```
pool {  
range 192.168.102.21 192.168.102.50;  
default-lease-time 86400; #24 Hours  
max-lease-time 172800; #48 Hours  
allow members of "Switch3Vlan2Class1";  
}
```

```
pool {  
range 192.168.102.51 192.168.102.80;  
default-lease-time 43200; #12 Hours  
max-lease-time 86400; #24 Hours  
allow members of "Switch3Vlan2Class2";  
}  
}
```

这时 DHCP 服务器会为经 Switch3 中继来自 Switch1 的网络节点分配 192.168.102.21~192.168.102.50 之间的地址，为来自 Switch1 的网络节点分配 192.168.102.51~192.168.102.80 之间的地址。

3.16.4 DHCP option 82 排错帮助

- ☞ DHCP option 82 是作为 DHCP 中继代理的一个子功能模块实现的，在使用 option 82 功能之前，要确保 DHCP 中继代理正确配置。
- ☞ DHCP option 82 是 DHCP 中继代理和 DHCP 服务器共同配合完成 IP 地址分配任务，DHCP 服务器要针对中继代理的网络拓扑，正确设置分配策略，否则即使中继代理的 option 82 工作正常，地址分配也不会成功进行。在存在多种中继代理情况时，注意设置好接口 DHCP 请求报文的重转发策略。
- ☞ 对于 DHCP 中继代理的 option 82 功能实现，运行过程中可以使用 `debug ip dhcp relay packet` 命令显示中继代理的数据包处理过程，包括添加 option 82 的具体内容，采取的重转发策略，中继代理剥离服务器的 option 82 内容等，这些信息有助于用户排错诊断。
- ☞ 对于 DHCP 服务器的 option 82 功能实现，运行过程中可以使用 `debug ip dhcp server packet` 命令显示服务器的数据包处理过程，包括显示识别出请求包的 option 82 信息，在应答包中返回的 option 82 信息。

3.17 DHCP Snooping

3.17.1 DHCP Snooping 介绍

DHCP Snooping 功能指交换机监测 DHCP CLIENT 通过 DHCP 协议获取 IP 的过程。它通过设置信任端口和非信任端口，来防止 DHCP 攻击及私设 DHCP SERVER。从信任端口接收的 DHCP 报文无需校验即可转发。典型的设置是将信任端口连接 DHCP SERVER 或者 DHCP RELAY 代理。非信任端口连接 DHCP CLIENT，交换机将转发从非信任端口接收的 DHCP 请求报文，不转发从非信任端口接收的 DHCP 回应报文。如果从非信任端口接收 DHCP 回应报文，除了发出告警信息外，并可根据设置对该端口执行相应的动作，比如 shutdown，下发 blackhole。如果启用了 DHCP Snooping 绑定功能，则交换机将会保存非信任端口下的 DHCP CLIENT 的绑定信息，每一条绑定信息包含该 DHCP CLIENT 的 MAC 地址、IP 地址、租期、VLAN 号和端口号，这些绑定信息存放于 DHCP Snooping 的绑定表中。利用绑定信息，DHCP Snooping 可以结合 dot1x，arp 等模块或独立实现用户的接入控制。

防伪装 DHCP Server: 一旦交换机截获了从非信任端口收到的 DHCP Server 回应包（包括 DHCPOFFER, DHCPACK 和 DHCPNAK），将发出警告，并相应做出反应（shutdown 端口或下发 BlackHole）。

防 DHCP 过载攻击: 要对信任端口和非信任端口做 DHCP 收包限速，防止过多的 DHCP 报文攻击 CPU。

记录 DHCP 绑定数据: DHCP SNOOPING 在转发 DHCP 报文的同时，记录 DHCP SERVER 分配的绑定数据，并可以把绑定数据上载到指定的服务器进行备份。这些绑定数据主要用于

配置 dot1x userbased 端口的动态用户。有关 dot1x userbased 模式的用法请参考《dot1x 配置》的章节。

添加绑定 ARP： DHCP Snooping 捕获到绑定数据之后，可以根据绑定数据中的参数添加静态绑定 ARP，这样可以防止 ARP 欺骗。

添加信任用户： DHCP Snooping 捕获到绑定数据之后，可以根据绑定数据中的参数添加信任用户表项，这样这些用户就可以不经过 DOT1X 认证而访问所有资源。

自动恢复： 交换机 shutdown 端口或者下发 blockhole 一段时间后，交换机应主动恢复该端口或源 Mac 的通讯，同时通过 syslog 发送信息到 Log Server。

LOG 功能： 交换机检测发现异常收包时；或者自动恢复时，应自动发送 syslog 信息到 Log Server。

私有报文的加密： 交换机与内网安全管理系统 TrustView 之间使用私有报文进行通讯，用户允许配置版本 2 的私有报文进行加密处理。

添加认证 option82 功能： 与 dot1x dhcption82 认证方式配合使用，根据用户认证状态在 DHCP 报文中添加不同 option 82 内容。

3.17.2 DHCP Snooping 的配置任务序列

1. 启动 DHCP Snooping
2. 启动 DHCP Snooping 绑定功能
3. 启动 DHCP Snooping option82 功能
4. 配置私有报文版本号
5. 配置私有报文 DES 加密密钥
6. 配置 helper server 地址
7. 启动 DHCP Snooping 绑定 ARP 功能
8. 设置信任端口
9. 启动 DHCP Snooping 绑定 DOT1X 功能
10. 启动 DHCP Snooping 绑定 USER 功能
11. 添加静态表项功能
12. 设置防御动作
13. 设置指定 vlan 的 dhcp snooping 功能
14. 打开调试开关
15. 设置 DHCP Snooping option82 属性

1. 启动 DHCP Snooping

命令	解释
全局配置模式	
ip dhcp snooping enable no ip dhcp snooping enable	开启或关闭 DHCP snooping 功能。

2. 启动 DHCP Snooping 绑定功能

命令	解释
全局配置模式	
ip dhcp snooping binding enable no ip dhcp snooping binding enable	开启或关闭 DHCP snooping 绑定功能。

3. 启动 DHCP Snooping option82 功能

命令	解释
全局配置模式	
ip dhcp snooping information enable no ip dhcp snooping information enable	开启或关闭 DHCP snooping option82 功能。

4. 配置私有报文版本号

命令	解释
全局配置模式	
ip user private packet version two no ip user private packet version two	配置或删除私有报文版本号。

5. 配置私有报文 DES 加密密钥

命令	解释
全局配置模式	
enable trustview key 0/7 <password> no enable trustview key	配置或删除私有报文 DES 加密密钥。

6. 配置 helper server 地址

命令	解释
全局配置模式	

ip user helper-address A.B.C.D [port <udpport>] source <ipAddr> (secondary)) no ip user helper-address (secondary))	配置或删除 helper address 地址。

7. 启动 DHCP Snooping 绑定 ARP 功能

命令	解释
全局配置模式	
ip dhcp snooping binding arp no ip dhcp snooping binding arp	开启或关闭 DHCP snooping 绑定 ARP 功能。

8. 设置信任端口

命令	解释
端口配置模式	
ip dhcp snooping trust vlan WORD no ip dhcp snooping trust vlan WORD ip dhcp snooping trust no ip dhcp snooping trust	设置或删除端口在指定 vlan 上的 DHCP snooping 信任属性。 设置或删除端口的 DHCP snooping 信任属性。

9. 启动 DHCP SNOOPING 绑定 DOT1X 功能

命令	解释
端口配置模式	
ip dhcp snooping binding dot1x no ip dhcp snooping binding dot1x	开启或关闭 DHCP snooping 绑定 dot1x 功能。

10. 启动 DHCP SNOOPING 绑定 USER 功能

命令	解释
端口配置模式	

ip dhcp snooping binding user-control no ip dhcp snooping binding user-control	开启或关闭 DHCP snooping 绑定 user 功能。

11. 添加静态绑定信息

命令	解释
全局配置模式	
ip dhcp snooping binding user <mac> address <ipAddr> vlan <vid> interface (ethernet) <ifname> no ip dhcp snooping binding user <mac> interface (ethernet) <ifname>	添加/删除 DHCP snooping 静态绑定表项。

12. 设置防御动作

命令	解释
端口配置模式	
ip dhcp snooping action {shutdown blackhole} [recovery <second>] no ip dhcp snooping action	设置或删除端口的 DHCP snooping 自动防御动作。

13. 设置指定 vlan 的 dhcp snooping 功能

命令	解释
全局配置模式	
ip dhcp snooping vlan {WORD <cr>} no ip dhcp snooping vlan {WORD <cr>}	开启 ip dhcp snooping vlan 功能后，只针对指定的 vlan 开启 dhcp snooping 功能，未指定的 vlan 不再开启 dhcp snooping 功能。关闭 dhcp snooping vlan 功能后，对所有已创建的 vlan 都开启 dhcp snooping 功能。

14. 打开调试开关

命令	解释

特权配置模式	
debug ip dhcp snooping packet debug ip dhcp snooping event debug ip dhcp snooping update debug ip dhcp snooping binding	请参照系统排错章节。

15. 设置 DHCP Snooping option82 属性

命令	解释
全局配置模式	
ip dhcp snooping information option subscriber-id format {hex acsii vs-hp}	本命令设置 DHCP snooping option82 功能 subscriber-id 默认格式。
ip dhcp snooping information option remote-id {standard <remote-id>} no ip dhcp snooping information option remote-id	本命令用于设置从端口接收的 DHCP 请求报文添加 option 82 子选项 2(远程 ID 选项)的具体内容。本命令的 no 操作把添加 option 82 子选项 2(远程 ID 选项)的形式设置为 standard。
ip dhcp snooping information option allow-untrusted (replace) no ip dhcp snooping information option allow-untrusted (replace)	本命令用于设置允许从 DHCP snooping 的非信任端口接收含有 option82 选项的 DHCP 报文，在设置参数 replace 时，允许替换 option82 选项；关闭此命令时，所有非信任端口将丢弃含有 option82 选项的 DHCP 报文。
端口配置模式	
ip dhcp snooping information option subscriber-id {standard <circuit-id>} no ip dhcp snooping information option subscriber-id	本命令用于设置从端口接收的 DHCP 请求报文添加 option 82 子选项 1(电路 ID 选项)的具体内容。本命令的 no 操作把添加 option 82 子选项 1(电路 ID 选项)的形式设置为 standard。

3.17.3 DHCP Snooping 典型应用

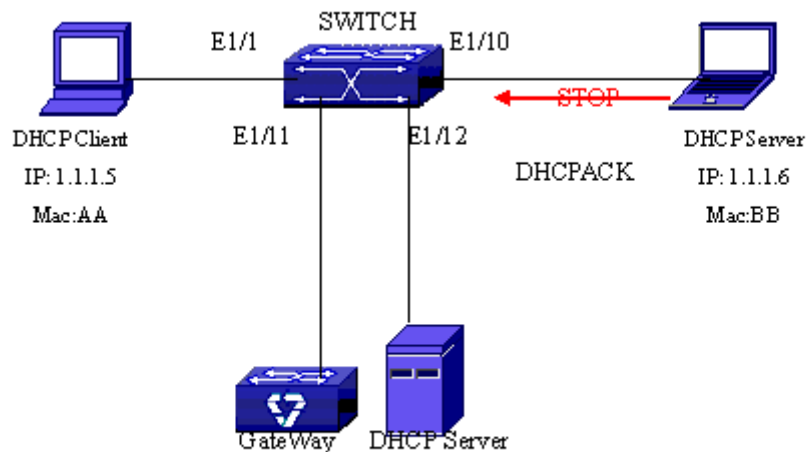


图 3-17 DHCP Snooping 防止非法 dhcp server 示意图

如上图，Mac-AA 设备为正常用户，连接在交换机非信任端口 1/1 上，其通过 DHCP Client 获得 IP 1.1.1.5；DHCP Server 和 GateWay 分别连接在 DCN 交换机的信任端口 1/11；1/12 上；恶意用户 Mac-BB 连接在非信任端口 1/10 上，试图伪装 DHCP Server（发送 DHCPACK）。在交换机上设置 DHCP snooping 将能有效发现并阻止这种网络攻击。

配置序列为：

```
switch#
switch#config
switch(config)#ip dhcp snooping enable
switch(config)#interface ethernet 1/11
switch(Config-Ethernet1/11)#ip dhcp snooping trust
switch(Config-Ethernet1/11)#exit
switch(config)#interface ethernet 1/12
switch(Config-Ethernet1/12)#ip dhcp snooping trust
switch(Config-Ethernet1/12)#exit
switch(config)#interface ethernet 1/1-10
switch(Config-Port-Range)#ip dhcp snooping action shutdown
switch(Config-Port-Range)#
```

3.17.4 DHCP Snooping 排错帮助

3.17.4.1 监控和调试信息

可以使用 `debug ip dhcp snooping` 命令来监控调试信息。

3.17.4.2 DHCP Snooping 排错帮助

当在使用 DHCP Snooping 功能出现问题时，请检查是否是如下原因：

- ☞ 检查全局 DHCP Snooping 开关是否打开；
- ☞ 如果配置 DHCP Snooping，而 DHCP 客户端没有获取 IP 时，请检查连接 dhcp server/relay 的端口是否已配置成信任端口。

3.18 DHCP option 60 和 option 43

3.18.1 DHCP option 60 和 option 43 介绍

DHCP 服务器解析来自 DHCP 客户端的 DHCP 报文，若报文中包含 option 60，则根据该 option 60 的内容以及 DHCP 服务器地址池中对 option 60 和 option 43 的配置来判断返回 option 43 给 DHCP 客户端。

在 DHCP 服务器地址池模式下配置相应的 option 60 和 option 43。

1. 地址池中同时配置了 option 60 和 option 43，若收到来自 DHCP 客户端的 DHCP 报文中可以解析到 option 60，与 DHCP 服务器地址池中的 option 60 比较，若匹配则返回该地址池中配置的 option 43 给 DHCP 客户端，若不匹配则不返回 option 43 给 DHCP 客户端
2. 地址池中仅配置了 option 43 选项，则默认匹配任何内容的 option 60，若收到的来自 DHCP 客户端的 DHCP 报文中可以解析到 option 60，直接返回地址池中配置的 option 43 给 DHCP 客户端
3. 地址池中仅配置了 option 60 选项，则不会返回 option 43 给 DHCP 客户端

3.18.2 DHCP option 60 和 option 43 配置任务序列

1. DHCP option 60 和 option 43 基本功能配置

命令	解释
地址池配置模式下	
option 60 ascii LINE	在 ip dhcp pool 模式下以 ascii 格式配置 option 60 字符串。
option 43 ascii LINE	在 ip dhcp pool 模式下以 ascii 格式配置 option 43 字符串。
option 60 hex WORD	在 ip dhcp pool 模式下以 hex 格式配置 option 60 字符串。
option 43 hex WORD	在 ip dhcp pool 模式下以 hex 格式配置 option 43 字符串。
option 60 ip A.B.C.D	在 ip dhcp pool 模式下以 IP 格式配置 option 60 字符串。
option 43 ip A.B.C.D	在 ip dhcp pool 模式下以 IP 格

	式配置 option 43 字符串。
no option 60	在地址池模式下删除配置的 option 60。
no option 43	在地址池模式下删除配置的 option 43。

3.18.3 DHCP option 60 和 option 43 举例

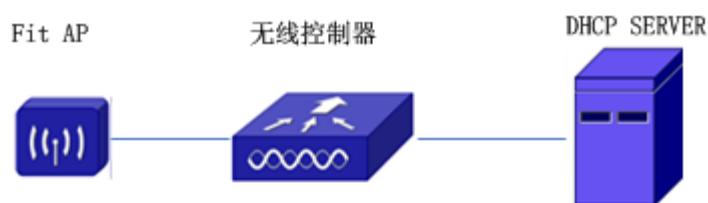


图 3-18 DHCP option 60 和 option 43 功能典型组网图

Fit AP 通过 DHCP server 获取 IP 地址、option 43 属性(此属性中携带无线控制器的 IP 地址信息)，然后向无线控制器发送单播发现请求。DHCP SERVER 通过配置匹配 fit ap 中的 option 60 选项来返回 option 43 属性给 FIT AP。DHCP option 43 携带的无线控制器地址为 192.168.10.5 和 192.168.10.6。

DHCP Server 配置

配置步骤：

配置 DHCP server

```
switch(config)#ip dhcp pool a
```

```
switch (dhcp-a-config)#option 60 ascii AP1000
```

```
switch (dhcp-a-config)#option 43 hex 0104C0A80A050104C0A80A06
```

3.18.4 DHCP option 60 和 option 43 排错帮助

当配置 DHCP option 60 和 option 43 功能出现问题时，请检查是否是如下原因：

- ◇ 是否开启 service dhcp 功能
- ◇ 地址池中若配置了 option 60，是否与报文中的 option 60 匹配

第4章 路由协议相关操作

4.1 路由协议概述

在 Internet 中，一台主机为了访问远端的另一台主机，必须通过一系列路由器或三层交换机选择一条合适的路径。

路由器或三层交换机都是通过 CPU 来计算路径，不同的是三层交换机将计算出来的路径添加到交换芯片中，由芯片进行线速转发；而路由器是将计算出来的路径保存在路由表和路由缓存区中，由 CPU 负责数据转发。可见，路由器和三层交换机都可以进行路径的选择，而三层交换机在数据转发上有比较强大的优势。下面简单描述一下三层交换机的路径选取的基本原理和方法。

在路径选取过程中，每台三层交换机只负责根据收到数据包的目的地址来选择一条合适的中间路径，然后将数据包传送给下一个三层交换机，直至路径上的最后一台三层交换机将数据包传送给目的主机。每台三层交换机所完成的将数据包传送到下一个三层交换机而选择的路径，就被称作路由。路由可以分为直连路由、静态路由和动态路由等几种。

直连路由是指到与该三层交换机直接相连网络的路径，三层交换机不用计算就可以获得。

静态路由是指人为指定的到某个网络或特定主机的路径，静态即不能随意更改。静态路由的优点是简单易配、稳定、限制非法的路由改变，便于实现负载分担，便于实现路由备份。但是，因为是人为设置，对于大型网络需要设置的路由过于庞大和复杂，因此不适用于中大型网络。

动态路由是指三层交换机根据启动的路由协议动态计算到某个网络或特定主机的路径。如果路径中下一跳三层交换机不可达，三层交换机能够自动丢弃通过该三层交换机的路径，选择通过其它三层交换机的路径。

动态路由协议一般分为两类：内部网关路由协议（IGP）和外部网关路由协议（EGP）。内部网关路由协议（IGP）是用来计算到某个自治系统内目的路由的协议。三层交换机支持的内部网关动态路由协议有 RIP 和 OSPF 路由协议，可以根据需要配置 RIP 和 OSPF 路由协议。三层交换机支持同时运行多个内部网关动态路由协议，也可以在某个动态路由协议中重新引入其它动态路由协议和静态路由从而将多个路由协议联系起来。

外部网关路由协议是用来在不同自治系统之间交换路由信息，比如 BGP 协议。目前，本公司三层交换机支持的外部网关路由协议有 BGP-4、BGP4+等。

4.1.1 路由表

如前所述，三层交换机主要用于建立当前三层交换机到达某个网络或特定主机的路由，并根据路由转发数据包。每台三层交换机都建立有一张路由表，记录着该三层交换机使用的所有路由。路由表中每条路由项都指明了数据包到某子网或主机应该通过三层交换机的哪个

物理端口发送，方可达到目的主机或到目的主机路径的下一台三层交换机。

路由表中包含下列一些主要的内容：

- ☞ 目的地址：用于标识 IP 数据包的目的地址或目的网络
- ☞ 网络掩码：与目的地址一起标识目的主机或三层交换机所在网段的网段地址。网络掩码由若干个连续的“1”构成，通常以点分十进制标识（一般为 1 到 4 个 255 组成的地址）。将目的地址和网络掩码“与”后，可以得到到目的主机或三层交换机所在网段的网络地址。例如，目的地址为 200.1.1.1，掩码为 255.255.255.0 的主机或三层交换机所在网段的网络地址为 200.1.1.0。
- ☞ 输出接口：指明 IP 数据包将从该三层交换机的哪个接口转发。
- ☞ 下一台三层交换机（下一跳）IP 地址：指明 IP 数据包将经由的下一台三层交换机。
- ☞ 路由项优先级：针对同一目的地，可能存在不同下一跳的若干条路由。这些路由可能是不同的动态路由协议发现的，也可能是人为配置的静态路由。优先级高的（数值小）将成为当前的最优路由。用户可以配置到同一目的地优先级不同的多条路由，三层交换机将按优先级顺序选取唯一的一条路由用于 IP 数据包转发。

为了不使路由表过于庞大，可以设置一条缺省路由。一旦查找路由表失败后，就选择缺省路由转发数据包。

三层交换机支持的各种路由协议及其发现路由的缺省优先级如下表表示：

路由协议或路由种类	优先级缺省值
直连路由	0
OSPF	110
静态路由（static）	1
RIP	120
OSPF ASE	150
IBGP	200
EBGP	20
未知路由	255

4.1.2 IP 路由策略

4.1.2.1 IP 路由策略介绍

路由器在发布与接收路由信息时，可能需要实施一些策略，以便对路由信息进行过滤，比如只接收或发布一部分满足给定条件的路由信息；一种路由协议（如 RIP）可能需要引入（redistribute）其它的路由协议如（OSPF）发现的路由信息，从而丰富自己的路由知识；路由器在引入其它路由协议的路由信息时，可能需要只引入一部分满足条件的路由信息，并对所引入的路由信息的某些属性进行设置，以使其满足本协议的要求。

为实现路由策略，首先要定义将要实施路由策略的路由信息的特征，即定义一组匹配规则，可以以路由信息中的不同属性作为匹配依据进行设置，如目的地址、发布路由信息的路由器地址等。匹配规则可以预先设置好，然后再将它们应用于路由的发布接收和引入等过程

的路由策略中。

在 DCNOS 中提供了 route-map、acl、as-path、community-list 和 ip-prefix 五种过滤器供路由协议引用，下面对各种过滤器逐个进行介绍：

1. route-map

用于匹配给定路由信息的某些属性，并在条件满足后对该路由信息的某些属性进行设置。

route-map 用于控制和改变路由信息，也可以控制路由之间的重新分配。一个 route-map 包含一系列 match 和 set 命令，match 命令指定需要满足的条件，set 命令则指明了当 match 条件匹配时要采取的相应动作。route-map 也用于控制不同路由进程之间的路由发布。

route-map 还可用于策略路由，使报文选择不同的路径而非最短路径。

一组 match 和 set 子句组成一个节点，一个 route-map 可以由多个节点构成，每个节点是进行匹配测试的一个单元。节点间依据序列号 sequence-number 进行匹配。match 子句定义匹配规则，匹配对象是路由信息的一些属性。同一节点中的不同 match 子句是“与”的关系，只有满足节点内所有 match 子句指定的匹配条件，才能通过该节点的匹配测试。set 子句指定动作，也就是在通过节点的匹配测试后所执行的动作——对路由信息的一些属性进行设置。

一个 route-map 的不同节点间是“或”的关系，系统依次检查 route-map 的各个节点，如果通过了 route-map 的某一节点，就意味着通过该 route-map 的匹配测试，不进入下一个节点的测试。

2. 访问控制列表acl

ACL (Access Control Lists)是交换机实现的一种数据包过滤机制，通过允许或拒绝特定的数据包进出网络，交换机可以对网络访问进行控制，有效保证网络的安全运行。用户可以基于报文中的特定信息制定一组规则（rule），每条规则都描述了对匹配一定信息的数据包所采取的动作：允许通过（permit）或拒绝通过（deny）。用户可以把这些规则应用到特定交换机端口的入口或出口方向，这样特定端口上特定方向的数据流就必须依照指定的 ACL 规则进出交换机。请参考《ACL 配置》一章。

3. 前缀列表ip-prefix

前缀列表 ip-prefix 的作用类似于 acl，但比它更为灵活，且更易于为用户理解。ip-prefix 在应用于路由信息的过滤时，其匹配对象为路由信息的目的地址信息域。

一个 ip-prefix 由前缀列表名标识。每个前缀列表可以包含多个表项，每个表项可以独立指定一个网络前缀形式的匹配范围，并用一个 sequence-number 来标识，sequence-number 指明了在 ip-prefix 中进行匹配检查的顺序。

在匹配的过程中，交换机按升序依次检查由 sequence-number 标识的各个表项，只要有某一表项满足条件，就意味着通过该 ip-prefix 的过滤（不会进入下一个表项的测试）。

4. 自治系统路径信息访问列表as-path

自治系统路径信息访问列表 as-path 仅用于 BGP。BGP 的路由信息包中，包含有一自治系统路径域（在 BGP 交换路由信息的过程中，路由信息经过的自治系统路径会记录在这个域中）。as-path 就是针对自治系统路径域指定匹配条件。

as-path 的有关配置请用户参考 BGP 配置中的 ip as-path 命令。

5. 团体属性列表community-list

团体属性列表 community-list, 仅用于 BGP。BGP 的路由信息包中包含一个 community 属性域, 用来标识一个团体。community-list 就是针对团体属性域指定匹配条件。

community-list 有关的配置请用户参考 BGP 中的 ip community-list 命令。

4.1.2.2 IP 路由策略配置

IP 路由策略配置任务序列:

- 1、定义 route-map
- 2、定义 route-map 的 match 语句
- 3、定义 route-map 的 set 语句
- 4、定义地址前缀列表

1.定义 route-map

命令	解释
全局配置模式	
route-map <map_name> {deny permit} <sequence_num> no route-map <map_name> [{deny permit} <sequence_num>]	配置 route-map; 本命令的 no 操作为删除 route-map。

2.定义 route-map 的 match 语句

命令	解释
route-map 配置模式	
match as-path <list-name> no match as-path [<list-name>]	匹配 BGP 路由经过的自制系统 AS 路径访问列表; 本命令的 no 操作为删除匹配。
match community <community-list-name community-list-num > [exact-match] no match community [<community-list-name community-list-num > [exact-match]]	匹配一个团体属性访问列表; 本命令的 no 操作为删除匹配。
match interface <interface-name > no match interface [<interface-name >]	按接口进行匹配; 本命令的 no 操作为删除匹配。

match ip <address next-hop> <ip-acl-name ip-acl-num prefix-list list-name> no match ip <address next-hop> [<ip-acl-name ip-acl-num prefix-list [list-name]>]	对地址或下一跳进行匹配；本命令的 no 操作为删除匹配。
match metric <metric-val > no match metric [<metric-val >]	对路由度量值进行匹配；本命令的 no 操作为删除匹配。
match origin <egp igp incomplete > no match origin [<egp igp incomplete >]	对路由源进行匹配；本命令的 no 操作为删除匹配。
match route-type external <type-1 type-2 > no match route-type external [<type-1 type-2 >]	对路由类型进行匹配；本命令的 no 操作为删除匹配。
match tag <tag-val > no match tag [<tag-val >]	对路由 tag 进行匹配；本命令的 no 操作为删除匹配。

3. 定义 route-map 的 set 语句

命令	解释
route-map 配置模式	
set aggregator as <as-number> <ip_addr> no set aggregator as [<as-number> <ip_addr>]	为 BGP 聚合者分配一个 AS 号；本命令的 no 操作为删除配置。
set as-path prepend <as-num> no set as-path prepend [<as-num>]	在 BGP 路由信息的 as-path 系列前加入指定的 AS 号；本命令的 no 操作为删除配置。
set atomic-aggregate no set atomic-aggregate	设置 BGP 原子聚合属性；本命令的 no 操作为删除配置。
set comm-list <community-list-name community-list-num > delete no set comm-list <community-list-name community-list-num > delete	删除 BGP 团体属性值；本命令的 no 操作为删除配置。
set community [AA:NN] [internet] [local-AS] [no-advertise] [no-export] [none] [additive] no set community [AA:NN] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]	设置 BGP 团体属性值；本命令的 no 操作为删除配置。

set extcommunity <rt soo> <AA:NN> no set extcommunity <rt soo> [<AA:NN>]	设置 BGP 扩展团体属性；本命令的 no 操作为删除配置。
set ip next-hop <ip_addr> no set ip next-hop [<ip_addr>]	设置下一跳 IP 地址；本命令的 no 操作为删除配置。
set local-preference <pre_val> no set local-preference [<pre_val>]	设置本地优先级；本命令的 no 操作为删除配置。
set metric < +/- metric_val metric_val> no set metric [< +/- metric_val metric_val>]	设置路由度量值；本命令的 no 操作为删除配置。
set metric-type <type-1 type-2> no set metric-type [<type-1 type-2>]	设置 OSPF 度量类型；本命令的 no 操作为删除配置。
set origin <egp igp incomplete > no set origin [<egp igp incomplete >]	设置 BGP 路由源；本命令的 no 操作为删除配置。
set originator-id <ip_addr> no set originator-id [<ip_addr>]	设置路由起源者 ID；本命令的 no 操作为删除配置。
set tag <tag_val> no set tag [<tag_val>]	设置 OSPF 路由 tag 值；本命令的 no 操作为删除配置。
set vpnv4 next-hop <ip_addr> no set vpnv4 next-hop [<ip_addr>]	设置 BGP VPNv4 下一跳地址；本命令的 no 操作为删除配置。
set weight <weight_val> no set weight [<weight_val>]	设置 BGP 路由权重；本命令的 no 操作为删除配置。

4.定义地址前缀列表

命令	解释
全局配置模式	
ip prefix-list <list_name> description <description> no ip prefix-list <list_name> description	对前缀列表进行描述；本命令的 no 操作为删除配置。
ip prefix-list <list_name> [seq <sequence_number>] <deny permit> < any ip_addr/mask_length [ge min_prefix_len] [le max_prefix_len]> no ip prefix-list <list_name> [seq <sequence_number>] [<deny permit> < any ip_addr/mask_length [ge min_prefix_len] [le max_prefix_len]>]	配置前缀列表；本命令的 no 操作为删除配置。

4.1.2.3 配置案例

下图是由四台三层交换机组成的一个网络，本例子示范如何通过 route-map 进行 BGP 的 as_path 属性的设置。各三层交换机之间运行 BGP 协议。对 Switch3 而言，网络 192.68.11.0/24 可以通过两条路径得知，一是以 AS_PATH 1 经 IBGP 得知(经过 Switch4)，二是以 AS_PATH 2 经 EBGP 得知(经过 Switch2)。BGP 优选最短路径，因此 AS_PATH 1 经 IBGP 的路径被优选。如果希望 BGP 优选路径 2，走 EBGP 路径，则可以将两个额外的 AS 路径号添加到从 Switch1 送到 Switch4 的 AS_PATH 信息中，以改变 Switch3 到达 192.68.11.0/24 的决策。

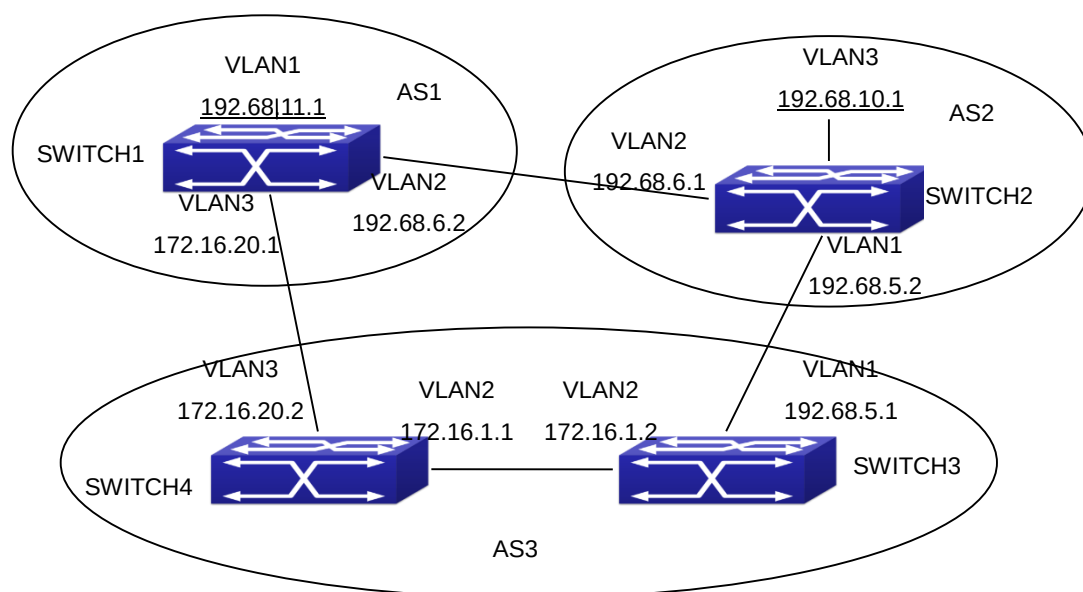


图 4-1 策略路由配置示意图

配置步骤：（此处仅列出 Switch1 的配置，其它交换机的配置省略）

三层交换机 Switch1 的配置

```
Switch1#config
Switch1(config)#router bgp 1
Switch1(config-router)#network 192.68.11.0 mask 255.255.255.0
Switch1(config-router)#neighbor 172.16.20.2 remote-as 3
Switch1(config-router)#neighbor 172.16.20.2 route-map AddAsNumbers out
Switch1(config-router)#neighbor 192.68.6.1 remote-as 2
Switch1(config-router)#exit
Switch1(config)#route-map AddAsNumbers permit 10
Switch1(config-route-map)#set as-path prepend 1 1
```

4.1.2.4 排错帮助

故障：路由协议运行正常的情况下无法实现路由信息学习

故障排除：检查如下几种错误

route-map 的各个节点中至少应该有一个节点的匹配模式是 permit 模式。当一个 route-map 用于路由信息过滤时，如果某路由信息没有通过任一节点的过滤，则认为该路由信息没有通过该 route-map 的过滤。当 route-map 的所有节点都是 deny 模式时，所有路由信息都不会通过该 route-map 的过滤。

地址前缀列表的各个表项中至少应该有一个表项的匹配模式是 permit 模式。deny 模式的表项可以先被定义，以快速的过滤掉不符合条件的路由信息。但如果所有表项都是 deny 模式，则任何路由都不会通过该地址前缀列表的过滤。可以在定义了多条 deny 模式的表项后定义一条 permit 0.0.0.0/0 le 32 的表项，以允许其它所有路由信息通过。如果不指定 less-equal 32 将只匹配缺省路由。

4.2 静态路由

4.2.1 静态路由介绍

如前所述，静态路由是指人为指定的到某个网络或特定主机的路径。静态路由的优点是简单易配、稳定、能够禁止非法的路由改变，同时便于实现负载分担和路由备份。但是，它也存在许多缺点。静态路由是静态的，一旦网络发生故障不能自动修改路由，必须人为参与配置，不适用于中大型网络。

静态路由主要应用于两种情况：1) 对于稳固的网络，为减少路由选择和路由数据流的负载，可使用静态路由。例如，到 STUB 网络的路由可采用静态路由。2) 为实现路由备份，可以使用静态路由（在备份线路配置静态路由，路由优先级低于主线路）。

静态路由与动态路由可以同时存在，三层交换机根据路由协议优先级的不同，选用优先级最高的路由。同时，在动态路由中也可以通过重新引入静态路由的方式（redistribute），将静态路由加入到动态路由中，并根据需要改变引入静态路由的优先级。

4.2.2 缺省路由介绍

缺省路由也是一种静态路由，它是在没有找到任何匹配路由时才使用的路由。在路由表中，缺省路由的表示形式为目的地址为 0.0.0.0，网络掩码也为 0.0.0.0 的路由。如果路由表中没有数据包所要到达的目的地，也没有缺省路由，则丢弃该数据包，并向源地址返回一个 ICMP 数据报指出此目的地址或网络的不可达。

4.2.3 静态路由配置

静态路由配置任务序列：

1. 静态路由配置
2. VRF 配置

1. 静态路由配置

命令	解释
全局配置模式	
<pre>ip route {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} {<gateway-address> <gateway-interface>} [<distance>] no ip route {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} {<gateway-address> <gateway-interface>} [<distance>]</pre>	配置静态路由；本命令的 no 操作为删除静态路由。

2. VRF 配置

命令	解释
全局配置模式	
<pre>ip route vrf <name> {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} {<gateway-address> <gateway-interface>} [<distance>] no ip route vrf <name> {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} [<gateway-address> <gateway-interface>] [<distance>]</pre>	配置静态路由；本命令的 no 操作为删除静态路由。

4.2.4 配置案例

下图是由三台三层交换机组成的一个简单的网络，各三层交换机和 PC 机 IP 地址的网络掩码均为 255.255.255.0， Switch-1 与 Switch-3 之间都通过配置静态路由以使 PC1 和 PC3 之间进行通信，PC3 到 PC2 之间的通信通过在 Switch-3 上配置到 Switch-2 的静态路由来实现，PC2 到 PC3 之间的通信通过在 Switch-2 上配置缺省路由来实现。

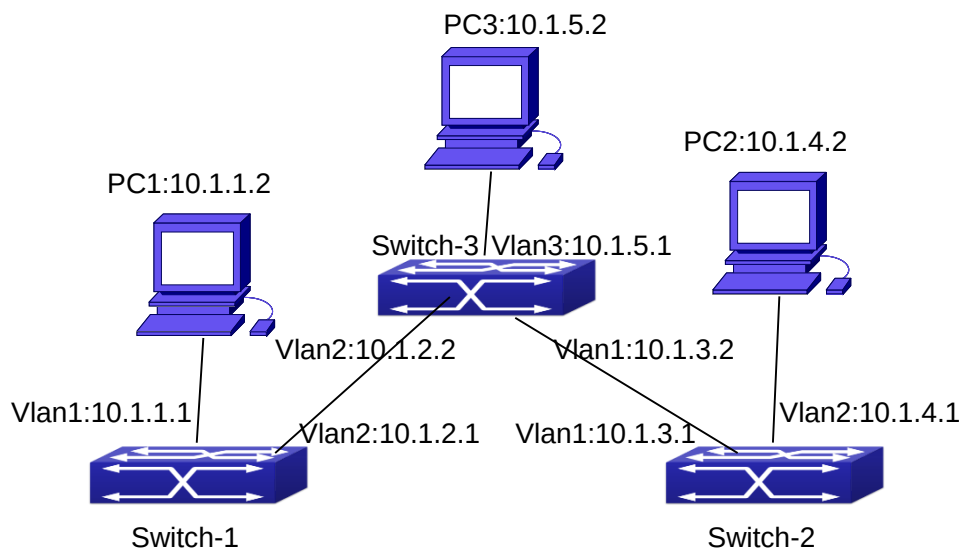


图 4-2 静态路由示意图

配置步骤:

三层交换机 Switch-1 的配置

```
Switch#config
```

```
Switch(config)#ip route 10.1.5.0 255.255.255.0 10.1.2.2
```

三层交换机 Switch-3 的配置

```
Switch#config
```

下一跳采用对端 IP 地址

```
Switch(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1
```

下一跳采用对端 IP 地址

```
Switch(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1
```

三层交换机 Switch-2 的配置

```
Switch#config
```

```
Switch(config)#ip route 0.0.0.0 0.0.0.0 10.1.3.2
```

这样，PC1 与 PC3、PC2 与 PC3 之间都可以 PING 通。

4.3 RIP

4.3.1 RIP 介绍

RIP协议最早在ARPANET网络中使用，专门用于小型简单网络中。RIP协议是基于Bellman-Ford算法的距离向量路由协议。运行距离向量协议的网络设备定期向相邻设备发送

两种信息：

- 到达目的网络所经过的跳数，即使用的度（metric），或者通过网络的数量。
- 下一跳是什么，或者达到目的网络要使用的方向（向量）。

距离向量三层交换机定期向相邻的三层交换机发送它们的整个路由选择表。三层交换机在从相邻三层交换机接收到的信息的基础之上建立自己的路由选择信息表。然后，将信息传递到它的相邻三层交换机。结果是路由选择表是在第二手信息的基础上建立的，距离代价超过15跳的路由将被视为不可达。

RIP协议是可选路由协议，它是基于UDP的协议，使用RIP的每个主机在UDP的520端口上发送和接收数据报。所有运行RIP协议的三层交换机，每隔30秒向所有邻居三层交换机发送路由表更新信息。如果180秒内没有收到来自对端的信息，那么认为该设备崩溃或相连的网络不可达。但到该三层交换机的路由还将在路由表中保持120秒，然后才被删除。

由于运行RIP协议的三层交换机使用第二手信息建立路由表，因此至少会遇到一个问题——无穷记数问题。对于一个运行RIP路由协议的网络，当某条RIP路由变为不可达，RIP三层交换机通常不会立刻发送路由更新报文，而是等待到达周期更新时间间隔（每30秒）时才发送有该路由信息的更新数据报。如果在没有收到更新报文之前，邻居向该三层交换机发送了一个包含邻居三层交换机自身路由表信息的数据报，那么会引起“无穷记数”现象，即出现到不可达三层交换机的路由选择度量固定递增的现象。这大大影响了路由选择、路由汇聚时间。

为了避免“无穷记数”现象，RIP协议提供了“水平分割”和“触发更新”等机制来解决路由循环问题。“水平分割”的原理是避免向网关发送从该网关学习到的路由，它包括“简单水平分割”——删除从邻居网关学习到的、将发送给该邻居网关的路由，和“逆向毒性水平分割”——不但在更新包中删除上述路由，而且将这些路由的代价设置为无穷。“触发更新”机制定义无论网关何时改变路由度量，都立即更新数据报并以广播形式发送出去，而不考虑30秒更新定时器的状态。

RIP协议包含版本1和版本2两个版本：RFC1058中介绍了RIP-I协议；RFC2453中介绍了RIP-II协议，同时兼容RFC1723和RFC1388。RIP-I采用发送广播数据报的方式发送路由更新数据报，它不支持子网掩码和认证。RIP-I的数据报中有一些域是不使用的，要求保证是全“0”，因此如果使用RIP-I应该进行全“0”域检查，如果这些域非“0”则丢弃此RIP-I数据报。RIP-II版本比RIP-I版本完善，它采用发送组播数据报的方式发送路由更新数据报（组播地址为224.0.0.9），它增加了子网掩码域和RIP验证域（支持简单明文密码和MD5密码验证），支持可变长子网掩码。RIP-II使用了RIP-I中的部分全“0”域，因此无需进行全“0”域检查。三层交换机缺省情况下是组播发送RIP-II数据报，接收RIP-I和RIP-II数据报。

每个运行RIP协议的三层交换机都有一个路由数据库，数据库中包含了该三层交换机所有可达目的地的路由项，并以此建立路由表。当RIP三层交换机向其相邻设备发送路由更新数据报时，路由更新数据报中包含了该三层交换机依据路由数据库建立的整个路由表。因此，对于较大的网络系统，每台三层交换机需要传输和处理的路由数据量很大、负担重，从而大大影响网络性能。

同时，RIP 协议支持将其它路由协议发现的路由信息引入到路由表中。也可以在 PE 路由器上作为与 CE 交换路由信息的协议，支持 VPN 路由/转发实例。

RIP 协议运行过程描述如下：

- 启动 RIP，以广播形式向其相邻三层交换机发送请求数据报，相邻设备收到请求数据报后，响应该请求，并回送包含本地路由信息的响应数据报。
- 三层交换机接收到响应数据报后，修改本地路由表，同时向相邻设备发送触发更新数据报，广播路由更新信息。相邻三层交换机接收到触发更新数据报之后，向其相邻的三层交换机发送触发更新数据报。经过一连串触发更新报的广播之后，各三层交换机都得到并保持最新的路由信息。

同时，RIP 三层交换机每隔 30 秒向其相邻设备广播本地路由表。相邻设备在接收到数据报之后，对本地路由进行维护，选择出最佳路由并向其各自的设备广播更新信息，使更新的路由最终达到全局有效。另外，RIP 采用超时机制对过时的路由进行超时处理，即三层交换机在一定时间间隔内（invalid 定时器间隔）没有收到来自某邻居的周期更新数据报，则认为来自该三层交换机的路由为无效路由，然后该路由再在路由表中保留一定时间间隔（holddown 定时器间隔），最后删除该路由。

4.3.2 RIP 配置

RIP 配置任务序列：

1. 启动 RIP 协议（必须）
 - （1）启动 RIP 模块/关闭 RIP 模块
 - （2）配置运行 RIP 协议的网段
2. 配置 RIP 协议参数（可选）
 - （1）配置 RIP 发包机制
 - 1) 配置 RIP 数据报的定点发送
 - 2) 配置 RIP 接口广播
 - （2）配置 RIP 路由参数
 - 1) 配置引入路由（缺省路由权值、配置 RIP 中引入其它协议的路由）
 - 2) 配置接口的验证模式及密码
 - 3) 配置路由偏移
 - 4) 配置应用路由过滤
 - 5) 配置水平分割
 - （3）配置 RIP 协议其它参数
 - 1) 配置 RIP 路由管理距离
 - 2) 配置路由表中 RIP 路由的数目限制
 - 3) 配置 RIP 更新、超时、抑制等计时器时间
 - 4) 配置 RIP UDP 接收缓冲区大小
3. 配置 RIP-I/RIP-II 模式切换

- (1) 配置所有接口使用的 RIP 版本
- (2) 配置接口发送/接收的 RIP 版本
- (3) 配置接口是否发送/接收 RIP 数据报
- 4. 删除 RIP 路由表中的特定路由
- 5. 配置 RIP VPN 命令
- 6. 配置 RIP 路由聚合
 - (1) 配置 IPv4 路由器模式的聚合路由
 - (2) 配置 IPv4 接口配置模式的聚合路由
 - (3) 显示 IPv4 聚合路由信息
- 7. 配置 RIP 引入不同进程 OSPF 路由
 - (1) 启动 RIP 引入不同进程 OSPF 路由功能
 - (2) 显示和调试 RIP 引入不同进程 OSPF 路由功能相关信息
- 8. 配置 RIP 的 VRF 地址族模式
 - (1) 启动 RIP 模块/关闭 RIP 模块
 - (2) 引用 VRF 地址族

1. 启动 RIP 协议

在系统中运行 RIP 路由协议的基本配置很简单，通常只需打开 RIP 开关、并且配置运行 RIP 的网段，即按 RIP 缺省配置发送和接收 RIP 数据报。如果需要可以切换发送、接收 RIP 数据报的版本，允许/禁止发送、接收 RIP 数据报，参考 3。

命令	解释
全局配置模式	
router rip no router rip	打开 RIP 协议；本命令的 no 操作关闭 RIP 协议。
路由器与地址族配置模式	
network <A.B.C.D/M ifname vlan> no network <A.B.C.D/M ifname vlan>	设定运行 RIP 协议的网段；本命令的 no 操作为删除运行 RIP 协议的网段。

2. 配置 RIP 协议参数

(1) 配置 RIP 发包机制

- 1) 配置 RIP 数据报的定点发送
- 2) 配置 RIP 接口广播

命令	解释
路由器配置模式	
neighbor <A.B.C.D> no neighbor <A.B.C.D>	指定需要定点发送的邻居路由器的 IP 地址；本命令的 no 操作取消指定的路由器。

passive-interface<ifname vlan> no passive-interface<ifname vlan>	指示 RIP 三层交换机在指定的接口上阻塞 RIP 广播，只能向配置了 neighbor 的三层交换机之间发送 RIP 数据包。本命令的 no 操作取消该功能。

(2) 配置 RIP 路由参数

1) 配置引入路由（缺省路由权值、配置 RIP 中引入其它协议的路由）

命令	解释
路由器配置模式	
default-metric <value> no default-metric	设定引入路由的缺省路由权值；本命令的 no 操作恢复缺省设置 1。
redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>] no redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>]	在 RIP 数据报中引入从其它路由协议引入的路由；本命令的 no 操作取消引入的相应协议的路由。
default-information originate no default-information originate	产生一条默认路由到 RIP 协议中；no 操作取消该特性。

2) 配置接口的认证模式及密码

命令	解释
接口配置模式	
ip rip authentication mode { text md5} no ip rip authentication mode [text md5]	设置认证使用的类型；本命令的 no 操作设置为取消该认证操作。
ip rip authentication string <text> no ip rip authentication string	设置认证使用的密钥；本命令的 no 操作不使用密钥。
ip rip authentication key-chain <name-of-chain> no ip rip authentication key-chain [<name-of-chain>]	设置认证使用的密钥链，本命令的 no 操作不使用密钥链。
全局模式	

accept-lifetime <start-time> {<end-time> duration<seconds> infinite} no accept-lifetime	配置 key chain 上的一个 key 接受为合法的时间；no 操作删除它。
send-lifetime <start-time> {<end-time> duration<seconds> infinite} no send-lifetime	配置 key chain 上的一个 key 可以发送的时间；no 操作删除 send-lifetime。

3) 配置路由偏移

命令	解释
路由器配置模式	
offset-list <access-list-number access-list-name> {in out }<number >[<ifname>] no offset-list <access-list-number access-list-name> {in out }<number >[<ifname>]	设定接口发送和接收 RIP 数据报时给路由的度量一个偏移量；本命令的 no 操作移去偏移列表。

4) 配置应用路由过滤

命令	解释
路由器配置模式	
distribute-list {< access-list-number access-list-name > prefix<prefix-list- name>}{in out} [<ifname>] no distribute-list {< access-list-number access-list-name > prefix<prefix-list- name>}{in out} [<ifname>]	配置应用访问列表和前缀列表来过滤路由。本命令的 no 操作配置不使用访问列表和前缀列表。

5) 配置水平分割

命令	解释
接口配置模式	
ip rip split-horizon [poisoned] no ip rip split-horizon	配置接口发送数据报时进行水平分割操作，poisoned 表示带逆向毒化。No 操作取消水平分割操作

(3) 配置 RIP 协议其它参数

- 1) 配置 RIP 路由优先级
- 2) 配置路由表中 RIP 路由的数目限制
- 3) 配置 RIP 更新、超时、抑制等计时器时间
- 4) 配置 RIP UDP 接收缓冲区大小

命令	解释
路由器配置模式	
distance <number> [<A.B.C.D/M>] [<access-list-name access-list-number>] no distance [<A.B.C.D/M>]	指定 RIP 协议的路由管理距离；本命令的 no 操作恢复缺省值 120。
maximum-prefix <maximum-prefix>[<threshold>] no maximum-prefix <maximum-prefix> no maximum-prefix	配置路由表中 RIP 路由的最大条数；本命令的 no 操作取消对路由条数的限制。
timers basic <update> <invalid> <garbage> no timers basic	调整 RIP 计时器更新、期满、垃圾收集的时间；本命令的 no 操作回复缺省设置。
recv-buffer-size <size> no recv-buffer-size	本命令配置 RIP 的 UDP 接收缓冲区大小；no 操作恢复系统缺省值。

3. 配置 RIP-I/RIP-II 模式切换**(1) 配置所有接口使用的 RIP 版本**

命令	解释
RIP 协议配置模式	
version { 1 2 } no version	设置所有三层交换机接口发送/接收 RIP 数据报的版本；no 操作恢复缺省设置，即版本 2。

(2) 配置接口发送/接收的 RIP 版本**(3) 配置接口是否发送/接收 RIP 数据报**

命令	解释
接口配置模式	
ip rip send version { 1 1-compatible 2 } no ip rip send version	设置接口发送 RIP 数据报版本；本命令的 no 操作设置接口使用 version 命令设置的版本。

ip rip receive version {1 2 } no ip rip receive version	设置接口接收 RIP 数据报版本; 本命令的 no 操作设置接口使用 version 命令设置的版本。
ip rip receive-packet no ip rip receive-packet	设定在接口上接收 RIP 数据报; 本命令的 no 操作关闭本接口的 RIP 数据报接收。
ip rip send-packet no ip rip send-packet	设定在接口上发送 RIP 数据报; 本命令的 no 操作关闭本接口的 RIP 数据报发送。

4. 删除 RIP 路由表中的特定路由

命令	解释
特权配置模式	
clear ip rip route {<A.B.C.D/M> kernel static connected rip ospf isis bgp all}	本命令删除 RIP 路由表中的特定路由。

5. 配置 RIP VPN 命令

命令	解释
路由器配置模式	
address-family ipv4 vrf <vrf-name> no address-family ipv4 vrf <vrf-name>	本命令为配置在 PE 路由器上的 VRF 配置 RIP 地址族。No 操作删除配置的地址族。
地址族配置模式	
exit-address-family	本命令退出地址族模式。

6. 配置 RIP 路由聚合

(1) 配置 IPv4 全局聚合路由

命令	解释
路由器配置模式	
ip rip aggregate-address A.B.C.D/M no ip rip aggregate-address A.B.C.D/M	配置或取消 IPv4 全局聚合路由。

(2) 配置 IPv4 接口聚合路由

命令	解释
接口配置模式	
ip rip aggregate-address A.B.C.D/M no ip rip aggregate-address A.B.C.D/M	配置或取消 IPv4 接口聚合路由。

(3) 显示 IPv4 聚合路由信息

命令	解释
特权模式和配置模式	
show ip rip aggregate	显示聚合路由信息。

7. 配置 RIP 引入不同进程 OSPF 路由

(1) 启动 RIP 引入不同进程 OSPF 路由功能

命令	解释
Router rip 配置模式	
redistribute ospf [<process-id>] [metric<value>] [route-map<word>] no redistribute ospf [<process-id>]	启动或者关闭 RIP 引入其他 OSPF 进程路由功能。

(2) 显示和调试 RIP 引入不同进程 OSPF 路由功能相关信息

命令	解释
特权和配置模式	
show ip rip redistribute	显示 RIP 进程引入其它外部路由的配置信息。
特权模式	
debug rip redistribute message send no debug rip redistribute message send debug rip redistribute route receive no debug rip redistribute route receive	打开或关闭 RIP 引入其他 OSPF 进程路由的命令下发调试开关。 打开或关闭 RIP 进程收到 nsm 发来的路由信息的调试开关。

8. 配置 RIP 的 VRF 地址族模式

命令	解释
Router RIP 配置模式	
address-family ipv4 vrf <vrf-name> no address-family ipv4 vrf <vrf-name>	本命令为配置在 PE 路由器上的 VRF 配置 RIP 地址族。No 操作删除配置的地址族。
地址族配置模式	
exit-address-family	本命令退出地址族模式。

4.3.3 RIP 案例

4.3.3.1 RIP 典型案例

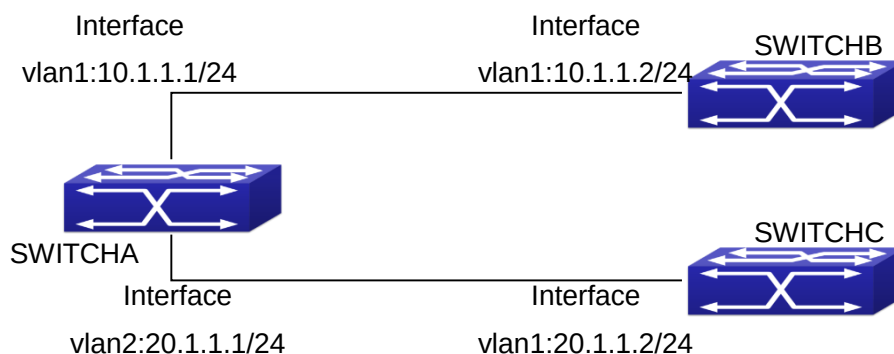


图 4-3 RIP 案例

图 3-1 为三台三层交换机组成的一个网络，三层交换机 SWITCHA 与三层交换机 SWITCHB 和三层交换机 SWITCHC 相连，三台三层交换机都运行 RIP 路由协议。设三层交换机 SWITCHA (interface vlan1: 10.1.1.1, interface vlan2: 20.1.1.1) 只与三层交换机 SWITCHB (interface vlan1: 10.1.1.2) 交流三层交换机更新信息，而不向三层交换机 SWITCHC (interface vlan1: 20.1.1.2) 交流三层交换机更新信息。

三层交换机 SWITCHA、SWITCHB、SWITCHC 的配置分别如下：

a) 三层交换机 SwitchA:

配置 interface vlan1 的 IP 地址。

```
SwitchA#config
```

```
SwitchA(config)# interface vlan 1
```

```
SwitchA(Config-if-Vlan1)# ip address 10.1.1.1 255.255.255.0
```

```
SwitchA (config-if-Vlan1)#
```

配置 interface vlan2 的 IP 地址

```
SwitchA (config)# vlan 2
```

```
SwitchA (Config-Vlan2)# switchport interface ethernet 1/2
```

```
Set the port Ethernet1/2 access vlan 2 successfully
```

```
SwitchA (Config-Vlan2)# exit
```

```
SwitchA (Config)# interface vlan 2
```

```
SwitchA (Config-if-Vlan2)# ip address 20.1.1.1 255.255.255.0
```

启动 RIP 协议，配置 RIP 网段

```
SwitchA(config)#router rip
```

```
SwitchA(config-router)#network vlan 1
```

```
SwitchA(config-router)#network vlan 2
```

```
SwitchA(config-router)#exit
```

配置接口 interface vlan 2 不向交换机 SwitchC 发送 RIP 报文

```
SwitchA(config)#router rip
```

```
SwitchA(config-router)#passive-interface vlan 2
```

```
SwitchA(config-router)#exit
SwitchA (config) #
b) 三层交换机 SwitchB:
配置 interface vlan1 的 IP 地址。
SwitchB#config
SwitchB(config)# interface vlan 1
SwitchB(Config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (Config-if-Vlan1)#exit
启动 RIP 协议，配置 RIP 网段
SwitchB(config)#router rip
SwitchB(config-router)#network vlan 1
SwitchB(config-router)#exit
c) 三层交换机 SwitchC:
配置 interface vlan1 的 IP 地址。
SwitchC#config
SwitchC(config)# interface vlan 1
SwitchC(Config-if-Vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC (Config-if-Vlan1)#exit
启动 RIP 协议，配置 RIP 网段
SwitchC(config)#router rip
SwitchC(config-router)#network vlan 1
SwitchC(config-router)#exit
```

4.3.3.2 RIP VPN 典型案例

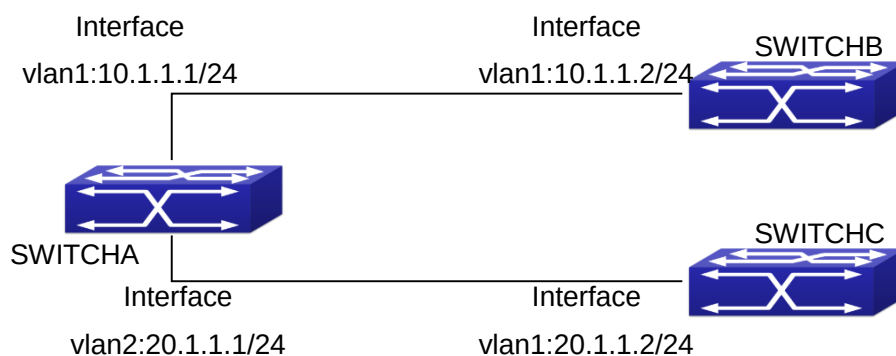


图 4-4 RIP VPN 案例

图 1-2 为三台三层交换机组成的网络。交换机 SwitchA 充当 PE，交换机 SWITCHB 和交换机 SwitchC 充当 CE1 与 CE2。PE 通过接口 vlan1 与 vlan2 分别与 CE1 和 CE2 相连。PE 与 CE 之间通过 RIP 协议交换路由信息

■ 作为 PE 的三层交换机 SwitchA:

配置 VPN 路由/转发实例 vpnb 和 vpnc

```
SwitchA#config
```

```
SwitchA(config)#ip vrf vpnb
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

```
SwitchA#(config)
```

```
SwitchA(config)#ip vrf vpnc
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

配置接口 vlan1 与 vlan2 分别与 vpnb 和 vpnc 相关联，并且配置 IP 地址

```
SwitchA(config)#in vlan1
```

```
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
```

```
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan1)#exit
```

```
SwitchA(config)#in vlan2
```

```
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc
```

```
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan2)#exit
```

分别配置与 vpnb 和 vpnc 相关联的 RIP 实例

```
SwitchA(config)#
```

```
SwitchA(config)#router rip
```

```
SwitchA(config-router)#address-family ipv4 vrf vpnb
```

```
SwitchA(config-router-af)#redistribute bgp
```

```
SwitchA(config-router-af)#network Vlan1
```

```
SwitchA(config-router-af)#exit-address-family
```

```
SwitchA(config-router)#address-family ipv4 vrf vpnc
```

```
SwitchA(config-router-af)#redistribute bgp
```

```
SwitchA(config-router-af)#network Vlan2
```

```
SwitchA(config-router-af)#exit-address-family
```

```
SwitchA(config-router)#
```

■ CE1 三层交换机 SwitchB:

配置以太网口 E1/2 的 IP 地址。

```
SwitchB#config
```

```
SwitchB(config)# interface Vlan1
```

```
SwitchB(config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
```

```
SwitchB (config-if-Vlan1)#exit
```

启动 RIP 协议，配置 RIP 网段

```
SwitchB(config)#router rip
```

```
SwitchB(config-router-rip)#network Vlan1
```

```
SwitchB(config-router-rip)#exit
```

■ CE2 三层交换机 SwitchC:

配置以太网口 E1/2 的 IP 地址。

```
SwitchC#config
```

```
SwitchC(config)# interface Vlan1
```

```
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

```
SwitchC (config-if-vlan1)#exit
```

启动 RIP 协议，配置 RIP 网段

```
SwitchC(config)#router rip
```

```
SwitchC(config-router)#network Vlan1
```

```
SwitchC(config-router)#exit
```

4.3.3.3 RIP 路由聚合功能典型案例

如下图的应用环境：

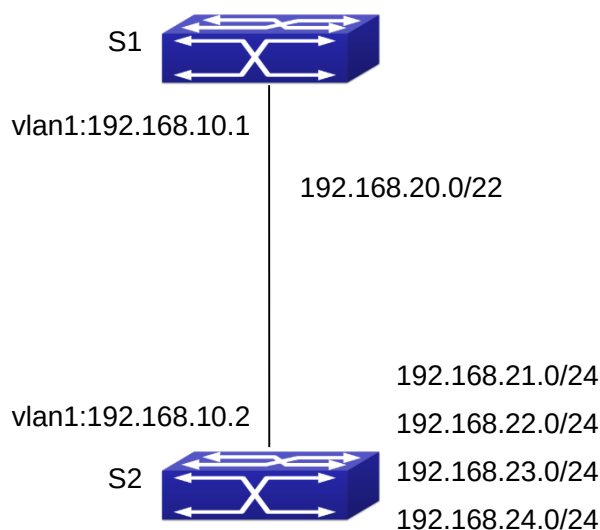


图 4-5 RIP 路由聚合典型应用环境

在上述网络拓扑图中，S2通过接口vlan1与S1相连，S2另外有四条子网路由，分别是：192.168.21.0/24，192.168.22.0/24，192.168.23.0/24，192.168.24.0/24。S2支持路由聚合，在S2的接口vlan1上配置聚合路由192.168.20.0/22后，通过vlan1给S1发送路由信息时，将四条子网路由聚合成一条聚合路由192.168.20.0/22，发送给S1，而不将子网发送给邻居。减小了S1的路由表规模，节省了内存。

S1配置序列：

```
S1(config)#router rip
```

```
S1(config-router) #network vlan 1
```

S2配置序列：

```
S2(config)#router rip
S2(config-router) #network vlan 1
S2(config-router) #exit
S2(config)#in vlan 1
S2(Config-if-Vlan1)# ip rip agg 192.168.20.0/22
```

4.3.4 RIP 排错帮助

在配置、使用 RIP 协议时，可能会由于物理连接、配置错误等原因导致 RIP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 然后，先启动 RIP 协议（使用 `router rip` 命令）并且配置网段（使用 `network` 命令）再在相应接口配置 RIP 协议参数，如使用的是 RIP-I 还是 RIP-II 等；
- ☞ 接着，注意 RIP 协议的自身特点——RIP 三层交换机每隔 30 秒向它所有邻居三层交换机发送路由表更新信息，如果 180 秒内没有收到来自某三层交换机的信息，那么认为该三层交换机崩溃或相连的网络不可达，到该三层交换机的路由还将在路由表中保持 120 秒然后被删除。因此，如果删除某个 RIP 路由，需等待 300 秒后才能保证路由表中除去此路由项；
- ☞ 如果是在 PE 路由器上使用 RIP 协议与 CE 交换路由信息。首先应该创建相应的 VPN 路由/转发实例，将相关接口与之关联。然后进入 RIP 的地址族模式配置相应的参数如果仍无法解决 RIP 路由问题，那么请使用 `debug rip` 调试命令，然后将 3 分钟内的 debug 信息拷贝下来，发送给本公司技术服务中心。

4.4 OSPF

4.4.1 OSPF 介绍

OSPF（Open Shortest Path First）协议即“开放最短路径优先协议”。它是一种基于链路状态的自治系统内部的动态路由协议，它通过三层交换机间交换链路状态信息来组成一个链路状态数据库，然后基于这个数据库用最短路径优先算法生成路由表。

自治系统（AS）是一个自我管理的互连网络。在大型网络中，例如Internet，极大的互连网络被分解为自治系统。连接到Internet上的大型公司网络是独立的自治系统，因为Internet上的其他主机并不由它来管理，而且它和Internet三层交换机并不共享内部路由选择信息。

每个链路状态三层交换机可提供与其邻居三层交换机的拓扑结构的信息：

- 三层交换机所连接的网段（链路）

- 连接链路的状态

链路状态信息在网络上泛洪（flooding），目的是使所有的三层交换机可以接收到第一手信息。链路状态三层交换机并不会广播包含在它们的路由表内的所有信息，相反链路状态三层交换机将发送关于已经改动的链路状态信息。链路状态三层交换机将向它们的邻居发送呼叫信息建立邻接关系，然后在邻接三层交换机之间发送链路状态通告（LSA）。邻接三层交换机将LSA复制到它们的路由选择表中，并传递这些信息到网络的剩余部分。这个过程称为泛洪（flooding）。这样，实现了向网络发送第一手信息，为网络建立更新路由的准确映射。链路状态路由选择协议使用称为代价的方法（而不是使用跳）来选择路径。代价是自动或人工赋值的。根据链路状态协议的算法，代价可以计算数据包必须穿越的跳数目、链路带宽、链路上的当前负载，甚至可以由管理员加入的权重来评价。

- 1) 当一个链路状态三层交换机进入链路状态互连网络时，它发送一个呼叫数据包（HELLO）以了解其邻居，并建立邻接关系。
- 2) 邻居用关于它们所连接的链路以及相关的代价度的信息进行应答。
- 3) 起始的三层交换机用这个信息来建立它的路由选择表。
- 4) 然后，作为定期更新的一部分。三层交换机向它的邻接三层交换机发送链路状态数据包。这个LSA包括了那个三层交换机的链路及相关代价。
- 5) 每个邻接三层交换机复制该LSA数据包，并且将LSA传递到下一个邻居（即泛洪）。
- 6) 因为三层交换机并没有在向前泛洪LSA之前重新计算路由选择数据库，因此收敛时间大大减少了。

链路状态路由选择协议的一个主要优点就是不可能形成路由无穷记数，原因是链路状态协议独立建立它们自己的路由选择信息表的方式。另一个优点是，在链路状态互连网络中收敛非常快，原因是一旦路由选择拓扑出现变动，则更新在互连网络上迅速泛洪。这些优点又释放了三层交换机的资源，因为对不好的路由信息所花费的处理能力和带宽消耗都很少。

OSPF协议的特点：OSPF协议支持各种规模的网络，最多可以支持几百台三层交换机；路由拓扑结构变化后OSPF能够立即发送链路状态更新数据包，收敛迅速；OSPF根据收集到的链路状态采用最短路径算法计算路由，保证了无自环路由；OSPF将自治系统划分为多个域，减小了数据库尺寸、减少了占用网络带宽、降低了计算负担（根据三层交换机在自治系统所处的位置可以划分为域内三层交换机、域边界三层交换机、自治系统边界三层交换机和骨干三层交换机）；OSPF支持负载分担，支持到同一目的地址的多条等价路由；OSPF支持4级路由机制（按域内路由、域间路由、第一类外部路由和第二类外部路由顺序分级处理路由）；OSPF协议支持IP子网、支持与其他路由协议的路由的引入、支持基于接口的报文验证；OSPF支持组播发送数据包。

每个OSPF三层交换机都维护着一份描述整个自治系统拓扑结构的数据库。每一台三层交换机都搜集本地状态的信息，如可用接口信息，可达邻居信息，然后通过使用链路状态通告（即向外发送链路状态信息），本三层交换机与其他OSPF三层交换机交换链路状态信息，从而形成描述整个自治系统的链路状态数据库。根据链路状态数据库，各三层交换机构建一棵以自己为根的最短路径树，这棵树给出了到自治系统中各节点的路由。如果存在两台或两台以上的三层交换机（即多路访问网络），该网络上要选出“指定三层交换机”和“备份指

定三层交换机”，指定三层交换机负责将网络的链路状态播出去，引入这一概念，有助于减少在多路访问网络上各三层交换机之间的数据流量。

OSPF协议要求将自治系统的网络划分成区域来管理，即将自治系统划分为0域（骨干域）和非0域，区域间传送的路由信息被进一步抽象概括，从而减少了整个网络的实用带宽。OSPF使用4类不同的路由，按优先顺序来说分别是区域内路由、区域间路由、第一类外部路由和第二类外部路由。区域内和区域间路由描述的是自治系统内部的网络结构，而外部路由则描述了应该如何选择到自治系统以外的目的地。第一类外部路由对应于OSPF从其它内部路由协议所引入的信息，这些路由的开销和OSPF自身路由的开销具有可比性；第二类外部路由对应于OSPF从外部路由协议所引入的信息，它们的开销远大于OSPF自身的路由开销，所以在计算路由开销时，不考虑OSPF自身的路由开销。

OSPF的区域以Backbone（骨干区域）作为核心，该区域标识为0域，所有的其他区域都必须与0域在逻辑上相连，0域必须保证连续。为此，骨干区域上特别引入了虚连接的概念以保证即使在物理上分割的该区域仍然在逻辑上具有连通性。在同一区域内的所有三层交换机的配置要保持一致。

如上所述，LSA只能在邻接三层交换机之间进行传递，OSPF协议包括五类LSA：路由器LSA、网络LSA、到其它域所在网络的汇总LSA、到自治系统边界三层交换机的汇总LSA和自治系统外部LSA。它们也可分别叫做类型1LSA、类型2LSA、类型3LSA、类型4LSA和类型5LSA。路由器LSA由OSPF域内的每个三层交换机产生，并发送给域内所有其它邻接三层交换机；网络LSA是由多路访问网络所在OSPF域的指定三层交换机产生的（为减少多路访问网络上各三层交换机之间的数据流量，多路访问网络中需要选出“指定三层交换机”和“备份指定三层交换机”，由指定三层交换机负责将网络的链路状态播出去），并发送给域内所有其它邻接三层交换机；汇总LSA由OSPF域边界三层交换机产生，并在域边界三层交换机之间传送；自治系统外部LSA由自治系统外部边界三层交换机产生，并在整个自治系统内传递。

对于以外部链路状态通告为主的自治系统，为减少拓扑数据库的尺寸，OSPF允许将某些域配置成STUB域。类型4LSA（ASBR汇总LSA）和类型5LSA（AS外部LSA）等两种LSA不允许泛滥进入/通过STUB域。STUB域内必须使用缺省路由，STUB域的域边界三层交换机通过汇总类型3LSA来通告到STUB域的缺省路由；这些缺省路由只在STUB内泛滥，不泛滥到STUB域之外。每个STUB域对应一条缺省路由，STUB域到AS外部目的地的路由只依赖该域的缺省路由。

OSPF协议路由的简单计算过程如下：

- 每个支持OSPF协议的三层交换机都维护着一个描述整个自治系统拓扑结构的链路状态数据库（即LS数据库）。每个三层交换机根据自己周围的网络拓扑结构生成一条链路状态通告（即路由器LSA），通过相互之间发送链路状态更新包（LSU）将各自的LSA发送给网络中的其它三层交换机。这样，每台三层交换机都收到了其它三层交换机的LSA，所有LSA在一起便形成了链路状态数据库。
- 由于一条LSA是对该三层交换机周围网络拓扑结构的描述，那么LS数据库则是对整个网络的拓扑结构的描述。三层交换机很容易根据LS数据库建立一张带权有向图，显然自治系统内的所有三层交换机都将得到一张相同的网络拓扑图。

- 每台三层交换机都使用最短路径 SPF 算法计算出一棵以自己为根的最短路径树，该树给出了到自治系统中各个节点的路由，外部路由信息为叶子节点，外部路由可根据广播它的三层交换机进行标记以记录关于自治系统的额外信息。由此可见，各个三层交换机得到的路由表是不同的。

OSPF 协议是 IETF 组织开发的，目前广泛使用的是 OSPFv2 版本是参照 RFC2328 中描述的内容实现的。

4.4.2 OSPF 配置

三层交换机的 OSPF 配置有自身的特点，简单说分两步：1、在全局使能 OSPF；2、配置接口所属的区域。配置任务序列如下：

1. 启动 OSPF 协议（必须）
 - （1）启动/关闭 OSPF 协议（必须）
 - （2）配置运行 OSPF 三层交换机的 ID 号（可选）
 - （3）配置运行 OSPF 的网络范围（可选）
 - （4）配置接口所属的域（必须）
2. 配置 OSPF 辅助参数（可选）
 - （1）配置 OSPF 发包机制参数
 - 1) 配置 OSPF 数据包的验证
 - 2) 配置 OSPF 接口为只收不发
 - 3) 配置接口发送数据包的代价
 - 4) 配置 OSPF 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）
 - （2）配置 OSPF 引入路由参数
 - 1) 配置引入外部路由的缺省参数（缺省类型、缺省标记值、缺省代价值）
 - 2) 配置在 OSPF 中引入其它协议的路由
 - （3）配置 OSPF 引入其他 OSPF 进程路由功能
 - 1) 启动 OSPF 引入其他 OSPF 进程路由功能
 - 2) 显示相关配置信息
 - 3) 调试
 - （4）配置 OSPF 协议其它参数
 - 1) 配置 OSPF 路由协议优先级
 - 2) 配置 OSPF STUB 域及缺省路由的代价
 - 3) 配置 OSPF 虚链路
 - 4) 配置接口在选举指定三层交换机 DR 中的优先级
 - 5) 配置打开/关闭记录 OSPF 邻居状态变化日志功能

- 6) 过滤 OSPF 计算得到的路由
3. 关闭 OSPF 协议

1. 启动 OSPF 协议

在三层交换机上运行 OSPF 路由协议的基本配置也很简单,通常只需打开 OSPF 开关、配置接口所在 OSPF 域即可, OSPF 协议的参数都使用缺省值。如需修改 OSPF 协议参数值, 参看 2.配置 OSPF 辅助参数。

命令	解释
全局配置模式	
[no] router ospf [process <id>] [VRF Name]	启动 OSPF 协议进程, 本命令的 no 操作关闭 OSPF 协议。(必须)
OSPF 协议配置模式	
router-id <router_id>	配置运行 OSPF 协议三层交换机的 ID
no router-id	号; 本命令的 no 操作取消三层交换机的 ID 号。缺省为选择某个接口的 IP 地址作为三层交换机 ID。(可选)
[no] network {<network> <mask> <network>/<prefix>} area <area_id>	配置某个网段属于某个区域, 本命令的 no 操作为取消该配置。(必须)

2. 配置 OSPF 辅助参数

(1) 配置 OSPF 发包机制参数

- 1) 配置 OSPF 数据包的验证
- 2) 配置 OSPF 接口为只收不发
- 3) 配置接口发送数据包的代价

命令	解释
接口配置模式	
ip ospf authentication { message-digest null} no ip ospf authentication	配置接口接受 OSPF 数据包所需要的验证方式 本命令的 no 操作恢复为缺省值。
ip ospf [<ip-address>] authentication-key <0 LINE 7 WORD LINE> no ip ospf [<ip-address>] authentication	指定接口上发送和接受 OSPF 报文所需要的验证密钥。本命令的 no 操作为取消验证密钥。

[no] passive-interface <ifname> [<ip-address>]	配置在特定接口上不发送 hello 分组，本命令的 no 操作取消该功能。
ip ospf cost <cost> no ip ospf cost	指定接口运行 OSPF 协议所需的代价；本命令的 no 操作恢复代价的缺省值。

4) 配置 OSPF 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）

命令	解释
接口配置模式	
ip ospf hello-interval <time> no ip ospf hello-interval	配置接口周期发送 HELLO 数据包的时间间隔；本命令的 no 操作恢复为缺省值。
ip ospf dead-interval <time> no ip ospf dead-interval	配置认定相邻三层交换机失效的时间间隔；本命令的 no 操作恢复为缺省值。
ip ospf transit-delay <time> no ip ospf transit-delay	设置在接口上传送链路状态广播的时延值；本命令的 no 操作恢复为缺省值。
ip ospf retransmit <time> no ip ospf retransmit	配置接口与邻接三层交换机之间传送链路状态通告时的重传间隔；本命令的 no 操作恢复为缺省值。

(2) 配置 OSPF 引入路由参数

配置在 OSPF 中引入其它协议的路由

命令	解释
OSPF 协议配置模式	
redistribute { bgp connected static rip kernel } [metric-type { 1 2 }] [tag <tag>] [metric <cost_value>] [router-map <WORD>] no redistribute { bgp connected static rip kernel }	引入其他协议发现路由和静态路由作为外部路由信息；本命令的 no 操作取消引入的外部路由信息。

(3) 配置 OSPF 引入其他 OSPF 进程路由功能

1) 启动 OSPF 引入其他 OSPF 进程路由功能

命令	解释
Router ospf 配置模式	

redistribute ospf [<process-id>] [metric<value>] [metric-type {1 2}][route-map<word>] no redistribute ospf [<process-i d>] [metric<value>] [metric-type {1 2}][route-map<word>]	启动或者关闭 OSPF 引入其他 OSPF 进程路由功能。

2) 显示相关配置信息

命令	解释
特权模式或者配置模式	
show ip ospf [<process-id>] redistribute	显示 ospf 进程引入其它外部路由的配置信息。

3) 调试

命令	解释
特权模式	
debug ospf redistribute message send no debug ospf redistribute message send debug ospf redistribute route receive no debug ospf redistribute route receive	打开或关闭 ospf 进程引入其他 ospf 进程路由的命令下发调试开关。 打开或关闭 ospf 进程收到 nsm 发来的路由信息的调试开关。

(4) 配置 OSPF 协议其它参数

- 1) 配置 OSPF spf 算法时间的计算
- 2) 配置 OSPF 链路状态数据库中 LSA 的数目限制
- 3) 配置 OSPF 各种区域参数

命令	解释
OSPF 协议配置模式	
timers spf <interval> no timers spf	配置 OSPF 的 SPF 定时器；本命令的 no 操作恢复为缺省值。

overflow database {<max-LSA> [hard soft] external <max-LSA> <recover time>} no overflow database [external <max-LSA> <recover time>]	配置当前 OSPF 进程数据库中 LSA 的限制；本命令的 no 操作恢复为缺省值。
area <id> {authentication [message-digest] default-cost <cost> filter-list {access prefix} <WORD> {in out} nssa [default-information-originate no-redistribution no-summary translator-role] range <range> stub [no-summary] virtual-link <neighbor>} no area <id> {authentication default-cost filter-list {access prefix} <WORD> {in out} nssa [default-information-originate no-redistribution no-summary translator-role] range <range> stub [no-summary] virtual-link <neighbor>}	配置 OSPF 区域下的参数（STUB 域，NSSA 域和虚链路）；本命令的 no 操作恢复为缺省值。

4) 配置接口在选举指定三层交换机 DR 中的优先级

命令	解释
接口配置模式	
ip ospf priority <priority> no ip ospf priority	配置接口在选举“指定三层交换机”时的优先级；本命令的 no 操作为恢复优先级缺省值。

5) 配置打开/关闭记录 OSPF 邻居状态变化日志功能

命令	解释
OSPF 协议配置模式	
log-adjacency-changes detail no log-adjacency-changes detail	设置是否记录 ospf 邻居变化的日志信息。

6) 过滤 OSPF 计算得到的路由

命令	解释
OSPF 协议配置模式	
filter-policy <access-list-name> no filter-policy	使用访问列表对 OSPF 计算得到的路由进行过滤。no 命令取消路由过滤。

3. 关闭 OSPF 协议

命令	解释
全局配置模式	
no router ospf [process <id>]	关闭 OSPF 路由协议。

4.4.3 OSPF 案例

4.4.3.1 OSPF 典型案例

案例一：OSPF自治系统

以五台三层交换机为例组成一个OSPF自治系统，OSPF区域划分见下图。

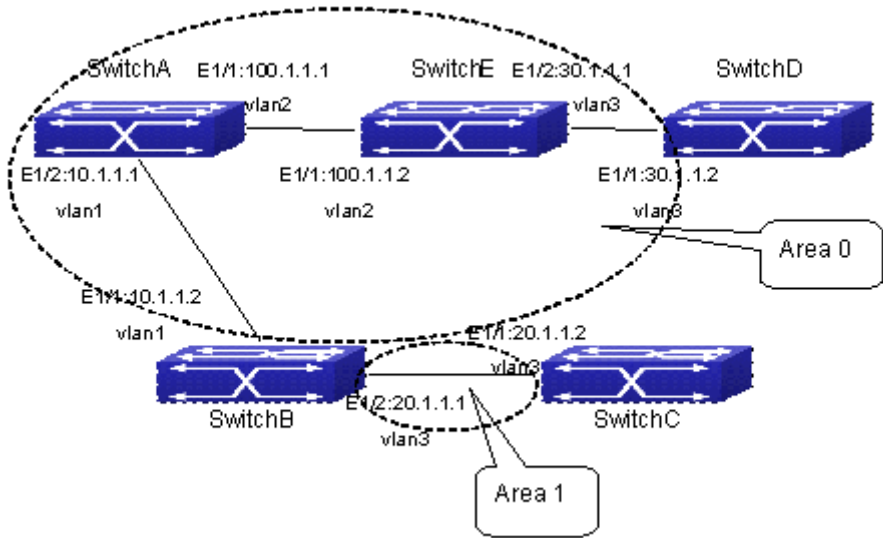


图 4-6 OSPF 自治系统网络拓扑示意图

三层交换机Switch1- Switch5的配置分别如下：

三层交换机Switch1：

配置接口vlan1的IP地址。

Switch1#config

Switch1(config)# interface vlan 1

Switch1(config-if-vlan1)# ip address 10.1.1.1 255.255.255.0

Switch1(config-if-vlan1)#exit

配置接口vlan2的IP地址

Switch1(config)# interface vlan 2

Switch1(config-if-vlan2)# ip address 100.1.1.1 255.255.255.0

```
Switch1(config-if-vlan2)#exit
```

启动OSPF协议，配置接口vlan1、vlan2所属域号

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 10.1.1.0/24 area 0
```

```
Switch1(config-router)#network 100.1.1.0/24 area 0
```

```
Switch1(config-router)#exit
```

```
Switch1(config)#exit
```

```
Switch1#
```

三层交换机Switch2:

配置接口vlan1和vlan2的IP地址。

```
Switch2#config
```

```
Switch2(config)# interface vlan 1
```

```
Switch2(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
```

```
Switch2(config-if-vlan1)#no shutdown
```

```
Switch2(config-if-vlan1)#exit
```

```
Switch2(config)# interface vlan 3
```

```
Switch2(config-if-vlan3)# ip address 20.1.1.1 255.255.255.0
```

```
Switch2(config-if-vlan3)#no shutdown
```

```
Switch2(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan1和vlan3所属OSPF区域

```
Switch2(config)#router ospf
```

```
Switch2(config-router)# network 10.1.1.0/24 area 0
```

```
Switch2(config-router)# network 20.1.1.0/24 area 1
```

```
Switch2(config-router)#exit
```

```
Switch2(config)#exit
```

```
Switch2#
```

三层交换机Switch3:

配置接口vlan3的IP地址。

```
Switch3#config
```

```
Switch3(config)# interface vlan 3
```

```
Switch3(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

```
Switch3(config-if-vlan3)#no shutdown
```

```
Switch3(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan3所属OSPF区域

```
Switch3(config)#router ospf
```

```
Switch3(config-router)# network 20.1.1.0/24 area 1
```

```
Switch3(config-router)#exit
```

```
Switch3(config)#exit
```

Switch3#

三层交换机Switch4:

配置接口vlan3的IP地址。

Switch4#config

Switch4(config)# interface vlan 3

Switch4(config-if-vlan3)# ip address 30.1.1.2 255.255.255.0

Switch4(config-if-vlan3)#no shutdown

Switch4(config-if-vlan3)#exit

启动OSPF协议，配置接口vlan3所属OSPF区域

Switch4(config)#router ospf

Switch4(config-router)# network 30.1.1.0/24 area 0

Switch4(config-router)#exit

Switch4(config)#exit

Switch4#

三层交换机Switch5:

配置接口vlan2的IP地址。

Switch5#config

Switch5(config)# interface vlan 2

Switch5(config-if-vlan2)# ip address 100.1.1.2 255.255.255.0

Switch5(config-if-vlan2)#no shutdown

Switch5(config-if-vlan2)#exit

配置接口vlan3的IP地址

Switch5(config)# interface vlan 3

Switch5(config-if-vlan3)# ip address 30.1.1.1 255.255.255.0

Switch5(config-if-vlan3)#no shutdown

Switch5(config-if-vlan3)#exit

启动OSPF协议，配置接口vlan2和vlan3所属域号

Switch5(config)#router ospf

Switch5(config-router)# network 30.1.1.0/24 area 0

Switch5(config-router)# network 100.1.1.0/24 area 0

Switch5(config-router)#exit

Switch5(config)#exit

Switch5#

案例二：OSPF协议典型复杂拓扑

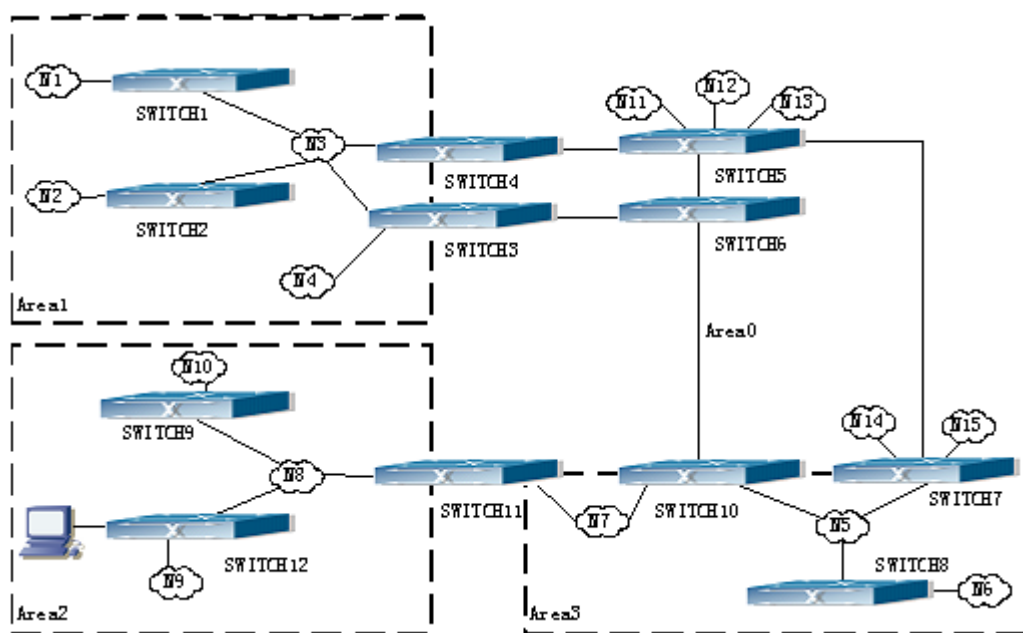


图 4-7 复杂典型 OSPF 自治系统

图 4-2 为一个典型复杂的 OSPF 自治系统网络拓扑。域 1 包括网络 N1-N4 和三层交换机 Switch1-Switch4，域 2 包括网络 N8-N10、主机 H1 与三层交换机 Switch9，域 3 包括网络 N5-N7 和三层交换机 Switch7、Switch8、Switch10 和 Switch11，且配置成网络 N8-N10 和主机 H1 使用一条汇总路由（即将域 3 定义为 STUB 域）。三层交换机 Switch1、Switch2、Switch5、Switch6、Switch8、Switch9 和 Switch12 是域内三层交换机，三层交换机 Switch3、Switch4、Switch7、Switch10 和 Switch11 是域边界三层交换机，三层交换机 Switch5 和 Switch7 是自治系统边界三层交换机。

就域 1 而言，三层交换机 Switch1 和 Switch2 是域内三层交换机，域边界三层交换机 Switch3 和 Switch4 负责向域 1 通告所有到该域外目的地的距离代价，同时 Switch3 和 Switch4 还必须向域 1 通告自治系统边界三层交换机 Switch5 和 Switch7 的位置，来自 Switch5 和 Switch7 的自治系统外部链路状态通告在整个自治系统内泛滥。当自治系统外部链路状态通告在域 1 内泛滥时，这些 LSA 被包含在域 1 数据库内，从而得到到网络 N11-N15 的路由。

另外，三层交换机 Switch3 和 Switch4 也必须向骨干域（即 0 域，所有非 0 域必须通过 0 域相连，不允许它们直接相连）汇总域 1 的拓扑结构，通告域 1 所包含的网络（即 N1-N4）以及从 Switch3 和 Switch4 到这些网络的代价。由于骨干域要求必须是连通的，所以在骨干三层交换机 Switch10 和 Switch11 之间必须建立一条虚链接。域边界三层交换机之间通过骨干三层交换机来交换汇总信息，每一个域边界三层交换机监听来自其它域边界三层交换机的汇总信息。

虚链路不仅用于保证骨干域的连通性，还用于健壮骨干域。比如，骨干三层交换机 Switch6 和 Switch10 之间的链路如果被切断，这将导致骨干域不连续。通过在骨干三层交换机 Switch7 和 Switch10 之间建立一条虚链路，骨干域会变得更健壮一些。同时，Switch7 和 Switch10 之间的虚链接提供了到域 3 和三层交换机 Switch7 的一条更短的路径。

以域1为例，假设三层交换机Switch1的接口vlan2的IP地址为10.1.1.1；三层交换机Switch2的接口VLAN2的IP地址为10.1.1.2；三层交换机Switch3的接口VLAN2的IP地址为10.1.1.3；三层交换机Switch4的接口VLAN2的IP地址为10.1.1.4。Switch1使用以太网口VLAN1与网络N1相连，IP地址为20.1.1.1；Switch2使用以太网口VLAN1与网络N2相连，IP地址为20.1.2.1；Switch3使用以太网口VLAN3与网络N4相连，IP地址为20.1.3.1；都属于域1。Switch3使用以太网口VLAN1与三层交换机Switch6相连，IP地址为10.1.5.1；Switch4使用以太网口VLAN1与三层交换机Switch5相连，IP地址为10.1.6.1；都属于域0。域1内的三层交换机之间采用简单密钥认证，域1边界三层交换机与域0骨干三层交换机之间采用MD5密钥认证。

下面仅给出域1内各三层交换机的配置，其他域三层交换机的配置省略。Switch1、Switch2、Switch3和Switch4的配置分别如下：

1)Switch1:

配置接口vlan2的IP地址

```
Switch1#config
```

```
Switch1(config)# interface vlan 2
```

```
Switch1(config-if-Vlan2)# ip address 10.1.1.1 255.255.255.0
```

```
Switch1(config-if-Vlan2)#exit
```

启动OSPF协议，配置接口vlan2所属域号

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 10.1.1.0/24 area 1
```

```
Switch1(config-router)#exit
```

配置简单密钥认证

```
Switch1(config)#interface vlan 2
```

```
Switch1(config-if-Vlan2)#ip ospf authentication
```

```
Switch1(config-if-Vlan2)#ip ospf authentication-key test
```

```
Switch1(config-if-Vlan2)#exit
```

配置接口vlan1的IP地址，配置所属域号

```
Switch1(config)# interface vlan 1
```

```
Switch1(config-if-Vlan1)#ip address 20.1.1.1 255.255.255.0
```

```
Switch1(config-if-Vlan1)#exit
```

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 20.1.1.0/24 area 1
```

```
Switch1(config-router)#exit
```

2)Switch2:

配置接口vlan2的IP地址

```
Switch2#config
```

```
Switch2(config)# interface vlan 2
```

```
Switch2(config-if-Vlan2)# ip address 10.1.1.2 255.255.255.0
```

```
Switch2(config-if-Vlan2)#exit
启动OSPF协议，配置接口vlan2所属域号
Switch2(config)#router ospf
Switch2(config-router)#network 10.1.1.0/24 area 1
Switch2(config-router)#exit
Switch2(config)#interface vlan 2
配置简单密钥认证
Switch2(config)#interface vlan 2
Switch2(config-if-Vlan2)#ip ospf authentication
Switch2(config-if-Vlan2)#ip ospf authentication-key test
Switch2(config-if-Vlan2)#exit
配置接口vlan1的IP地址，配置所属域号
Switch2(config)# interface vlan 1
Switch2(config-if-Vlan1)#ip address 20.1.2.1 255.255.255.0
Switch2(config-if-Vlan1)#exit
Switch2(config)#router ospf
Switch2(config-router)#network 20.1.2.0/24 area 1
Switch2(config-router)#exit
Switch2(config)#exit
Switch2#
3)Switch3:
配置接口vlan2的IP地址
Switch3#config
Switch3(config)# interface vlan 2
Switch3(config-if-Vlan2)# ip address 10.1.1.3 255.255.255.0
Switch3(config-if-Vlan2)#exit
启动OSPF协议，配置接口vlan2所属域号
Switch3(config)#router ospf
Switch3(config-router)#network 10.1.1.0/24 area 1
Switch3(config-router)#exit
配置简单密钥认证
Switch3(config)#interface vlan 2
Switch3(config-if-Vlan2)#ip ospf authentication
Switch3(config-if-Vlan2)#ip ospf authentication-key test
Switch3(config-if-Vlan2)#exit
配置接口vlan3的IP地址，配置所属域号
Switch3(config)# interface vlan 3
Switch3(config-if-Vlan3)#ip address 20.1.3.1 255.255.255.0
```



```
Switch3(config-if-Vlan3)#exit
Switch3(config)#router ospf
Switch3(config-router)#network 20.1.3.0/24 area 1
Switch3(config-router)#exit
配置接口vlan1的IP地址，并配置接口所属域号
Switch3(config)# interface vlan 1
Switch3(config-if-Vlan1)#ip address 10.1.5.1 255.255.255.0
Switch3(config-if-Vlan1)#exit
Switch3(config)#router ospf
Switch3(config-router)#network 10.1.5.0/24 area 0
Switch3(config-router)#exit
配置MD5密钥认证
Switch3(config)#interface vlan 1
Switch3 (config-if-Vlan1)#ip ospf authentication message-digest
Switch3 (config-if-Vlan1)#ip ospf authentication-key test
Switch3 (config-if-Vlan1)#exit
Switch3(config)#exit
Switch3#
4)Switch4:
配置接口vlan2的IP地址
Switch4#config
Switch4(config)# interface vlan 2
Switch4(config-if-Vlan2)# ip address 10.1.1.4 255.255.255.0
Switch4(config-if-Vlan2)#exit
启动OSPF协议，配置接口vlan2所属域号
Switch4(config)#router ospf
Switch4(config-router)#network 10.1.1.0/24 area 1
Switch4(config-router)#exit
配置简单密钥认证
Switch4(config)#interface vlan 2
Switch4(config-if-Vlan2)#ip ospf authentication
Switch4(config-if-Vlan2)#ip ospf authentication-key test
Switch4(config-if-Vlan2)#exit
配置接口vlan1的IP地址，并配置接口所属域号
Switch4(config)# interface vlan 1
Switch4(config-if-Vlan1)# ip address 10.1.6.1 255.255.255.0
Switch4(config-if-Vlan1)#exit
Switch4(config)#router ospf
```

```
Switch4(config-router)#network 10.1.6.0/24 area 0
Switch4(config-router)#exit
配置MD5密钥认证
Switch4(config)#interface vlan 1
Switch4(config-if-Vlan1)#ip ospf authentication message-digest
Switch4(config-if-Vlan1)#ip ospf authentication-key test
Switch4(config-if-Vlan1)#exit
Switch4(config)#exit
Switch4#
```

案例三：OSPF引入其他OSPF进程路由功能

如下图所示，一台运行 OSPF 路由协议的交换机连接两个网络，网络 A 与网络 B，出于某种原因，要求网络 A 可以学到网络 B 的路由，但网络 B 不可以学到网络 A 的路由，根据需求，可配置在 interface vlan1 与 interface vlan2 上分别起 OSPF 的两个进程，同时配置 interface vlan1 所属的 OSPF 进程引入 interface vlan2 所属的 OSPF 进程的路由，但不能配置 interface vlan2 所属的 OSPF 进程引入 interface vlan1 所属的 OSPF 进程的路由。

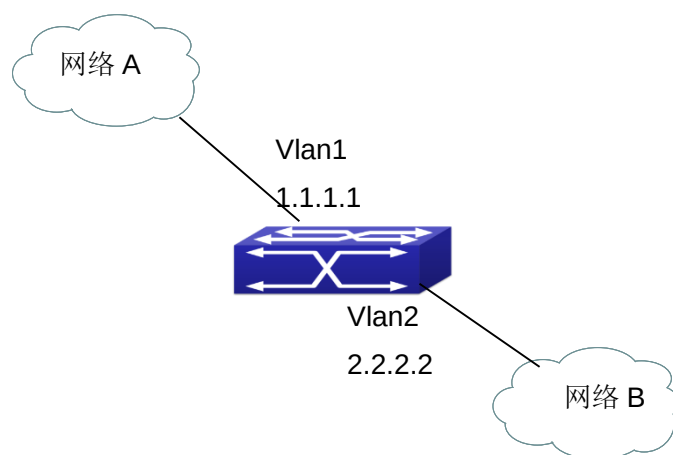


图 4-8 OSPF 引入其他 OSPF 进程路由功能案例

我们可以进行如下配置：

```
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 1.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 2.2.2.2 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(config)#router ospf 10
```

```

Switch(config-router)#network 2.2.2.0/24 area 1
Switch(config-router)#exit
Switch(config)#router ospf 20
Switch(config-router)#network 1.1.1.0/24 area 1
Switch(config-router)#redistribute ospf 10
Switch(config-router)#exit

```

4.4.3.2 OSPF VPN 典型案例

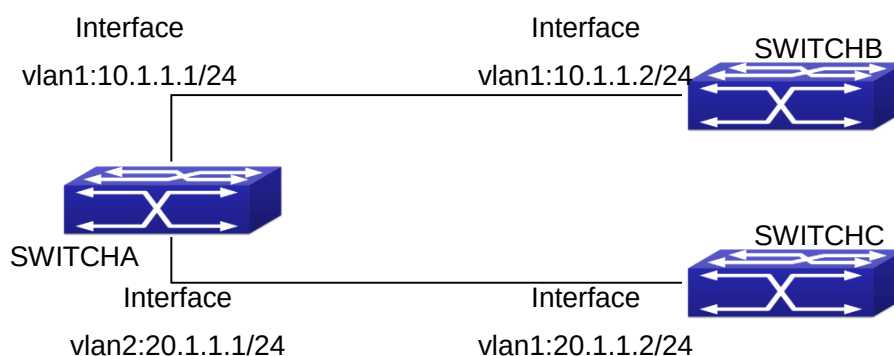


图 4-9 OSPF VPN 案例

图 4-4 为三台三层交换机组成的网络。交换机 SwitchA 充当 PE，交换机 SwitchB 和交换机 SwitchC 充当 CE1 与 CE2。PE 通过接口 vlan1 与 vlan2 分别与 CE1 和 CE2 相连。PE 与 CE 之间通过 OSPF 协议交换路由信息

a) 作为 PE 的三层交换机 SwitchA:

配置 VPN 路由/转发实例 vpnb 和 vpnc

```

SwitchA#config
SwitchA(config)#ip vrf vpnb
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit
SwitchA#(config)
SwitchA(config)#ip vrf vpnc
SwitchA(config-vrf)#
SwitchA(config-vrf)#exit

```

配置接口 vlan1 与 vlan2 分别与 vpnb 和 vpnc 相关联，并且配置 IP 地址

```

SwitchA(config)#in vlan1
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
SwitchA(config-if-Vlan1)#exit
SwitchA(config)#in vlan2
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc

```

```
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan2)#exit
```

分别配置与 vpnb 和 vpnc 相关联的 OSPF 实例

```
SwitchA(config)#
```

```
SwitchA(config)#router ospf 100 vpnb
```

```
SwitchA(config-router)#network 10.1.1.0/24 area 0
```

```
SwitchA(config-router)#redistribute bgp
```

```
SwitchA(config-router)#exit
```

```
SwitchA(config)#router ospf 200 vpnc
```

```
SwitchA(config-router)#network 20.1.1.0/24 area 0
```

```
SwitchA(config-routerf)#redistribute bgp
```

b) CE1 三层交换机 SwitchB:

配置以太网口 E1/2 的 IP 地址。

```
SwitchB#config
```

```
SwitchB(config)# interface Vlan1
```

```
SwitchB(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
```

```
SwitchB (config-if-vlan1)#exit
```

启动 OSPF 协议，配置 OSPF 网段

```
SwitchB(config)#router ospf
```

```
SwitchB(config-router-rip)#network 10.1.1.0/24 area 0
```

```
SwitchB(config-router-rip)#exit
```

c) CE2 三层交换机 SwitchC:

配置以太网口 E1/2 的 IP 地址。

```
SwitchC#config
```

```
SwitchC(config)# interface Vlan1
```

```
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

```
SwitchC (config-if-vlan1)#exit
```

启动 OSPF 协议，配置 OSPF 网段

```
SwitchC(config)#router ospf
```

```
SwitchC(config-router)#network 20.1.1.0/24 area 0
```

```
SwitchC(config-router)#exit
```

4.4.4 OSPF 排错帮助

在配置、使用 OSPF 协议时，可能会由于物理连接、配置错误等原因导致 OSPF 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；

- ☞ 然后在各接口上配置不同网段的 IP 地址；
- ☞ 然后，先启动 OSPF 协议（使用 `router ospf` 命令）再在相应接口配置所属 OSPF 域；
- ☞ 接着，注意 OSPF 协议的自身特点——OSPF 骨干域（0 域）必须保证是连续的；如果不连续，要使用虚链路（virtual link）来保证；所有非 0 域只能通过 0 域与其他非 0 域相连，不允许非 0 域直接相连；边界三层交换机是指该三层交换机的一部分接口属于 0 域，而另外一部分接口属于非 0 域；对于广播网等多访问网，需要选举指定三层交换机 DR。

4.5 BGP

4.5.1 BGP 介绍

BGP（Border Gateway Protocol）是边界网关协议。它是一种自治系统（AS，autonomous system）间的动态路由发现协议。它的基本功能是在 AS 间自动交换无环路的路由信息。BGP 通过交换带有自治系统编号的 AS 序列属性的路径可达信息，来选择最佳路由从而消除路由环路并实施用户配置的策略。通常在一个 AS 内部的交换机可以使用多种内部网关协议（IGP，Interior Gateway Protocol）来相互交换 AS 内部的路由信息，如 RIP、OSPF 都是 IGP 协议；而用外部网关协议（EGP，Exterior Gateway Protocol）来交换 AS 之间的路由信息，如 BGP 就是 EGP 中的一种。通常一个 AS 是建立在单一的管理部门之上的。BGP 经常用于 ISP 之间或是跨国公司的各个分部之间的交换机上。

BGP 从 1989 年就开始使用，它最早发布的三个版本分别是 RFC1105（BGP-1）、RFC1163（BGP-2）和 RFC1267（BGP-3），当前最流行的是使用的 RFC1771（BGP-4）。本交换机目前支持 BGP-4。

1. BGP-4 的特性

BGP-4 适用于分布式结构并支持无类域间路由（CIDR，Classless InterDomain Routing）。BGP-4 正迅速成为事实上的 Internet 外部路由协议标准，BGP-4 特性如下：

- BGP 是一种外部路由协议，与 OSPF、RIP 等的内部路由协议不同，BGP 并不能发现和计算路由，但可以控制路由的传播和选择最好的路由。
- 通过在更新路由中携带 AS 路径信息可以彻底解决路由环路的问题。
- 使用 TCP 作为其传输层协议提高了协议的可靠性。端口号为 179。
- BGP-4 支持无类域间路由（CIDR，Classless InterDomain Routing），这是较 BGP-3 的一个重要改进。CIDR 以一种全新的方法看待 IP 地址，不再区分 A 类网、B 类网及 C 类网。例如一个非法的 C 类网络地址 192.213.0.0 255.255.0.0 采用 CIDR 表示法 192.213.0.0/16 就成为一个合法的超级网络。其中 /16 表示网络号由从地址左端开始的 16 比特构成。CIDR 的引入简化了路由聚合（Routes Aggregation）。路由聚合实际上是合并几个不同路由的过程，这样从通告几条路由变为通告一条路由，减化了路由表。

- 路由更新时BGP只发送增量路由，大大减少了BGP传播路由所占用的带宽，适用于在Internet上传播大量的路由信息。
- 由于政治上和经济上的原因，每个自治系统希望对路由进行过滤选择和控制，因此BGP-4提供了丰富的路由策略，它使得BGP便于扩展以支持因特网新的发展。

2. BGP-4运行概述

不同于RIP和OSPF协议，BGP协议是面向连接的。BGP交换机之间要交换路由信息就一定需要建立连接后才能交换路由信息。BGP协议的运行是通过消息驱动的，其消息共可分为四类：

打开消息(Open Message)——是TCP连接建立后第一个发送的消息。它用于建立BGP同伴间的BGP连接关系。Open消息中的一些参数用于协商BGP邻居间的连接能否建立。

保持消息(Keepalive Message)——是用于检测连接有效性的消息。一般是周期发送用于保持BGP连接。若在holdtime时间内没有收到这个消息或者Update消息，则关闭BGP连接。

更新消息(Update Message)——是BGP系统中最重要消息，用于在同伴之间交换路由信息。除了交换机交换更新的路由信息，也交换不可用的或是已经取消的路由信息。它由三部分构成：不可达路由(unreachable)、网络可达性信息(NLRI Network Layer Reachability Information)、路径属性(Path Attributes)。

通告消息(Notification Message)——是错误通告消息。当一个BGP发言人收到这样的消息时要关闭与它的邻居的BGP连接。

BGP-4是面向连接的。BGP系统作为高层协议，运行在一个特定的网络设备上。当发现邻居后，就建立并维护一个TCP会话。建立了会话之后就要进行路由表的交换和同步。仅仅在系统初启时，通过发送整个BGP路由表交换路由信息。之后只有在有更新的路由信息时，才交换路由信息。此时只交换增量更新信息(update message)。BGP-4通过使用keepalive信息来周期性地维护链路和会话，即通过定期地接收和发送保活消息(keepalive)来检测相互之间的连接是否正常。

参与BGP会话的交换机称为BGP发言者(speaker)。它不断的接收或产生新路由信息，并将它通告(advertise)给其它的BGP发言者。当BGP发言者收到来自其它自治系统的新路由通告时，如果该路由比当前已知路由好，或者当前还没有可接受路由，那么它就把这个路由通告给自治系统内所有其它的BGP发言者。一个BGP发言者也将同它交换消息的其它的BGP发言者称为邻居(neighbor)或对等体(peer)，若干相关的邻居可以构成对等体组(peer group)。BGP在交换机上以下列两种方式运行：

- 1、IBGP: Internal BGP
- 2、EBGP: External BGP

当BGP运行于同一AS内部时被称为IBGP。当BGP运行于不同自治系统之间时称为EBGP。通常，外部邻居彼此物理上相邻，而内部邻居则可以在同一AS的任何地方。它们之间的不同最终体现在BGP路由协议对路由信息的处理方式上。设备可以通过检查邻居交换机发送的打开消息中的AS号，来决定邻居的BGP交换机是作为外部邻居还是作为内部邻

居。

IBGP在AS内部使用，它向该AS内的所有BGP邻居传递信息。IBGP在一个大的组织内交流AS路由信息。注意，AS内的交换机不必是物理介质上相邻，只要是交换机位于同一个AS内部，那么两者之间就可以互为邻居。因为BGP本身不能发现路由，因此需要其它的内部路由协议（如静态路由、直连路由、OSPF和RIP）的路由表中包含邻居的IP地址所在的网段。并用该路由进行BGP之间的信息交换。为了防止路由环路，一个BGP发言人从内部邻居收到路由通告时，不会将这些路由通告给其它内部邻居。

EBGP在AS之间使用，它向外部AS的BGP邻居传递路由信息。EBGP之间一般彼此物理上临近，即共享同一介质。因为EBGP之间要物理上相互临近，所以在两个AS之间的边界设备通常就是EBGP。一个BGP发言人从外部邻居收到路由通告时，则将这些路由通告给其它内部邻居。

3. 路由属性

BGP-4可以通过相关机制达到共享和查询内部IP路由表的目的，但它仍然拥有自己的路由表。在BGP路由表中，每一条路由都拥有：一个网络号（network number），所经过的AS列表信息（也被称为as路径），和一些路由属性（如origin）。BGP-4使用的路由属性很复杂，该属性可以在进行路径选择时提供度量的依据。

4. BGP的路由选择策略

当从多个邻居收到关于同一个路由的BGP通告时，经过路由过滤后需要考虑选择最优路由。这个过程叫做BGP的路由选择过程。BGP的路由选择过程只有下列条件满足时才会开始：

1. 交换机的路由必需是下一跳可达的。即在路由表中，有能够到达下一跳地址的路由。
2. BGP 必需和 IGP 同步（除非设置成了非同步，仅限于 IBGP）

BGP路由选择过程基于BGP的属性值。当有多条路由是指明同一个目的地时，BGP则要选择一条到达目的地的最佳路由。该决策过程如下：

1. 优先选用具有最大权重（weight）的路由；
2. 如果权重相同，优选具有最大本地优先级（local preference）的路由；
3. 如果本地优先级相同，优选本地交换机始发的路由。
4. 如果本地优先级相同，并且没有本地交换机始发的路由，优选具有最短AS路径的路由；
5. 如果 AS 路径相同，优选具有最低 origin 类型的路由（IGP<EGP<INCOMPLETE）；
6. 如果origin类型相同，优选具有最低MED属性的路由，除非激活bgp always-compare-med命令，否则这种比较只在来自同一邻居AS的路由间进行。
7. 如果 MED 属性相同，EBGP 优于联盟外部，联盟外部优于 IBGP；
8. 如果所有前面的条件都相同，则判断这些路由的 NEXT_HOP 属性，在 IGP 路由表中到达其下一跳地址对应的路由 cost 最低的路由被优选

9. 如果至此仍然相同，则用BGP交换机ID（router ID）打破平衡。最优路由来自于交换机的BGP Router-ID 最小的那个。

4.5.2 BGP 配置

BGP 的配置任务分为基本配置任务和高级配置任务。BGP 的基本的配置任务包括：

1. 启动 BGP 路由（必须）
2. 配置 BGP 邻居（必须）
3. 管理路由策略的改变
4. 配置 BGP 权重
5. 设置基于邻居的 BGP 路由过滤策略
6. 设置 BGP 的下一跳
7. 配置 EBGP 的多跳
8. 配置 BGP 的会话标识
9. 配置 BGP 的版本号

高级 BGP 配置任务包括：

1. 使用路由映射（route-map）来更改路由
2. 配置路由聚合
3. 配置路由的团体属性过滤
4. 设置 BGP 的联盟
5. 设置路由反射器
6. 设置对等体组
7. 配置邻居和对等体组参数
8. 调整 BGP 定时器
9. 调整 BGP 的通告间隔
10. 设置默认的本地优先级
11. 允许发送缺省路由
12. 设置 BGP 的 MED 值
13. 配置 BGP 的路由重分配
14. 配置 BGP 路由衰减
15. 配置 BGP 能力协商
16. 配置路由服务器
17. 配置选路规则
18. 配置 BGP 引入不同进程 OSPF 路由
 - （1）启动 BGP 引入不同进程 OSPF 路由功能
 - （2）显示和调试 BGP 引入不同进程 OSPF 路由功能相关信息

一. 基本 BGP 配置任务

1. 启动 BGP 路由

命令	解释
全局配置模式	
router bgp <as-id> no router bgp <as-id>	启动 BGP，该命令的 no 形式关闭 BGP 进程。
BGP 协议配置模式	
bgp asnotation asdot no bgp asnotation asdot	配置以分隔符（asdot）方式显示 AS 号 and 进行正则表达式匹配。
network <ip-address/M> no network <ip-address/M>	设置 BGP 要通告的网络，该命令的 no 形式取消要通告的网络。
address-family ipv4 {unicast multicast vrf <vrf-nam>} no address-family ipv4 {unicast multicast vrf <vrf-nam>}	创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4。

2. 配置 BGP 邻居

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} remote-as <as-id> no neighbor {<ip-address> <TAG>} [remote-as <as-id>]	指定一个 BGP 邻居，该命令的 no 操作删除邻居。

3. 管理路由策略的改变

(1) 配置硬重建

命令	解释
特权用户模式	
clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>}	配置 BGP 硬重建。

(2) 配置出站软重建

命令	解释
特权用户模式	

clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>} soft out	配置 BGP 出站软重建。

(3) 配置入站软重建

命令	解释
BGP 协议配置模式	
neighbor { <ip-address> <TAG> } soft-reconfiguration inbound no neighbor { <ip-address> <TAG> } soft-reconfiguration inbound	该命令可以存储从邻居或是对等体组发过来的路由信息，该命令的 no 形式取消路由信息的存储。
特权用户模式	
clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>} soft in	配置 BGP 入站软重建。

4. 配置 BGP 权重

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} weight <weight> no neighbor {<ip-address> <TAG>} weight <weight>	配置 BGP 邻居的权重值，该命令的 no 形式恢复默认的权重值。

5. 设置基于邻居的 BGP 路由过滤策略

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} distribute-list {<1-199> <1300-2699> <WORD>} {in out} no neighbor {<ip-address> <TAG>} distribute-list {<1-199> <1300-2699> <WORD>} {in out}	过滤邻居的路由更新信息，该命令的 no 形式取消路由过滤。

6. 设置 BGP 的下一跳

1) 设置下一跳为本交换机的地址

命令	解释
BGP 协议配置模式	

neighbor {<ip-address> <TAG>} next-hop-self no neighbor {<ip-address> <TAG>} next-hop-self	发送路由时下一跳设置为自己的地址。该命令的 no 形式取消设置。

2) 通过路由映射取消默认的下一跳

命令	解释
路由映射配置模式	
set ip next-hop <ip-address> no set ip next-hop	设置出去路由的下一跳属性，该命令的 no 形式取消该设置。

7. 配置 EBGp 的多跳

如果外部邻居之间的连接不是直连的，那么可以使用下面的命令配置邻居的多跳。

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>] no neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>]	配置允许同不直接相连网络上的交换机建立 EBGp 连接。该命令的 no 形式取消设置。

8. 配置 BGP 的会话标识

命令	解释
BGP 协议配置模式	
bgp router-id <ip-address> no bgp router-id	配置 router-id 标识的值，该命令的 no 形将恢复默认值。

9. 配置 BGP 的版本号

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} version <value> no neighbor {<ip-address> <TAG>} version	设置 BGP 邻居使用的版本号，该命令的 no 形式恢复缺省设置。目前仅支持版本 4。

二. 高级 BGP 配置任务

1. 使用路由映射（route-map）来更改路由

命令	解释
BGP 协议配置模式	

neighbor {<ip-address> <TAG>} route-map <map-name> {in out} no neighbor {<ip-address> <TAG>} route-map <map-name> {in out}	将路由映射应用于入站或出站的路由。no 命令取消路由映射的设置。

2. 配置路由聚合

命令	解释
BGP 协议配置模式	
aggregate-address <ip-address/M> [summary-only] [as-set] no aggregate-address <ip-address/M> [summary-only] [as-set]	在 BGP 路由表中建立一条聚合记录，该命令的 no 形式取消聚合记录。

3. 配置路由的团体属性过滤

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} send-community no neighbor {<ip-address> <TAG>} send-community	允许向 BGP 邻居发送的路由更新带有团体属性，该命令的 no 形式使向邻居发送的路由不带团体属性。

4. 设置 BGP 的联盟

命令	解释
BGP 协议配置模式	
bgp confederation identifier <as-id> no bgp confederation identifier <as-id>	配置一个 BGP 自治系统联盟标识，该命令的 no 形式删除 BGP 自治系统联盟标识。
bgp confederation peers <as-id> [<as-id>..] no bgp confederation peers <as-id> [<as-id>..]	配置属于自治系统联盟的 AS，该命令的 no 形式从自治系统联盟中删除 AS。

5. 设置路由反射器

(1) 设置路由反射器和它的客户端可以使用下面的命令：

命令	解释
BGP 协议配置模式	

neighbor <ip-address> route-reflector-client no neighbor <ip-address> route-reflector-client	配置当前的交换机做为路由反射器并且指明一个客户端，该命令的 no 形式删除一个客户端。

(2) 如果簇内有多于一个路由反射器，设置簇 ID 的命令如下所示：

命令	解释
BGP 协议配置模式	
bgp cluster-id <cluster-id> no bgp cluster-id	配置簇 ID，该命令的 no 形式簇 ID 的设置。

(3) 如果需要进行客户端到客户端的路由反射，那么可以用下面的命令：

命令	解释
BGP 协议配置模式	
bgp client-to-client reflection no bgp client-to-client reflection	配置允许客户端到客户端的路由反射，该命令的 no 形式禁止客户端到客户端的路由反射。

6. 设置对等体组

(1) 创建对等体组

命令	解释
BGP 协议配置模式	
neighbor <TAG> peer-group no neighbor <TAG> peer-group	创建对等体组，该命令的 no 形式删除对等体组。

(2) 将邻居加入对等体组内

命令	解释
BGP 协议配置模式	
neighbor <ip-address> peer-group <TAG> no neighbor <ip-address> peer-group <TAG>	配置向对等体组内添加一个成员，该命令的 no 形式取消指定的成员。

7. 配置邻居和对等体组参数

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} remote-as <as-id>	指定一个 BGP 邻居，该命令的 no 操作删除邻居。

no neighbor {<ip-address> <TAG>} remote-as <as-id>	
neighbor {<ip-address> <TAG>} description <.LINE> no neighbor {<ip-address> <TAG>} description	指定和 BGP 邻居的描述信息, 该命令的 no 形式删除描述信息。
neighbor {<ip-address> <TAG>} default-originate [route-map <NAME>] no neighbor {<ip-address> <TAG>} default-originate [route-map <NAME>]	允许发送缺省路由 0.0.0.0, 该命令的 no 形式取消发送默认路由。
neighbor {<ip-address> <TAG>} send-community no neighbor {<ip-address> <TAG>} send-community	设置向邻居发送路由的团体属性。
neighbor {<ip-address> <TAG>} timers <keepalive> <holdtime> no neighbor {<ip-address> <TAG>} timers	设置某个特定的邻居的 keepalive 和 holdtime 定时器时间, 该命令的 no 形式恢复默认值。
neighbor {<ip-address> <TAG>} advertisement-interval <seconds> no neighbor {<ip-address> <TAG>} advertisement-interval	设置发送 BGP 路由信息之间的最小间隔, 该命令的 no 形式恢复默认值。
neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>] no neighbor {<ip-address> <TAG>} ebgp-multihop	配置允许同不直接相连网络上的 EBGp 建立连接。该命令的 no 形式取消设置。
neighbor {<ip-address> <TAG>} weight <weight> no neighbor {<ip-address> <TAG>} weight	配置 BGP 邻居的权重值, 该命令的 no 形式恢复默认的权重值。
neighbor {<ip-address> <TAG>} distribute-list { <access-list-number> <name> } { in out } no neighbor {<ip-address> <TAG>} distribute-list { <access-list-number> <name> } { in out }	过滤邻居的路由更新信息, 该命令的 no 形式取消路由过滤。
neighbor {<ip-address> <TAG>} route-reflector-client no neighbor {<ip-address> <TAG>}	配置当前的交换机做为路由反射器并且指明一个客户端, 该命令的 no 形式删除一个客户端。

route-reflector-client	
neighbor {<ip-address> <TAG>} next-hop-self no neighbor {<ip-address> <TAG>} next-hop-self	发送路由时下一跳设置为自己的地址。该命令的 no 形式取消设置。
neighbor {<ip-address> <TAG>} version <value> no neighbor {<ip-address> <TAG>} version	指定和 BGP 邻居进行通信的 BGP 版本, 该命令的 no 形式恢复默认设置。
neighbor {<ip-address> <TAG>} route-map <map-name> {in out} no neighbor {<ip-address> <TAG>} route-map <map-name> {in out}	将路由映射应用于入站或出站的路由。no 命令取消路由映射的设置。
neighbor {<ip-address> <TAG>} soft-reconfiguration inbound no neighbor {<ip-address> <TAG>} soft-reconfiguration inbound	该命令可以存储从邻居或是对等体组发过来的路由信息, 该命令的 no 形式取消路由信息的存储。
neighbor {<ip-address> <TAG>} shutdown no neighbor {<ip-address> <TAG>} shutdown	关闭 BGP 邻居或对等体组, 该命令的 no 形式激活已经关闭的 BGP 邻居或对等体组。

8. 调整 BGP 定时器

(1) 设置所有邻居的 BGP 定时器。

命令	解释
BGP 协议配置模式	
timers bgp <keepalive> <holdtime> no timers bgp	设置所有邻居的 BGP 定时器, 该命令的 no 形式恢复默认值。

(2) 设置特定邻居的定时器值

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} timers <keepalive> <holdtime> no neighbor {<ip-address> <TAG>} timers	设置某个特定的邻居的 keepalive 和 holdtime 定时器时间, 该命令的 no 形式恢复默认值。

9. 调整 BGP 的通告间隔

命令	解释

BGP 协议配置模式	
neighbor {<ip-address> <TAG>} advertisement-interval <seconds> no neighbor {<ip-address> <TAG>} advertisement-interval	设置发送 BGP 路由更新信息之间的最小间隔，该命令的 no 形式恢复缺省设置。

10. 设置默认的本地优先级

命令	解释
BGP 协议配置模式	
bgp default local-preference <value> no bgp default local-preference	改变默认的本地优先级，该命令的 no 形式恢复默认值。

11. 允许发送缺省路由

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} default-originate no neighbor {<ip-address> <TAG>} default-originate	允许发送缺省路由 0.0.0.0，该命令的 no 形式取消发送默认路由

12. 设置 BGP 的 MED 值

(1) 设置 MED 值

命令	解释
路由映射配置模式	
set metric <metric-value> no set metric	配置 metric 的值，该命令的 no 形式恢复默认值

(2) 对来自不同 AS 的路径应用基于 MED 的路径选择

命令	解释
BGP 协议配置模式	
bgp always-compare-med no bgp always-compare-med	允许对来自不同 AS 的路径进行 MED 值的比较，该命令的 no 形式禁止对来自不同 AS 的路径进行 MED 值的比较。

13. 配置 BGP 的路由重分配

命令	解释
BGP 协议配置模式	

redistribute {connected static rip ospf} [metric <metric>] [route-map <NAME>] no redistribute {connected static rip ospf}	将 IGP 路由重分配到 BGP。同时可以指定再分配的 metric、路由映射，该命令的 no 形式取消重分配。

14. 配置 BGP 的路由衰减

命令	解释
BGP 协议配置模式	
bgp dampening [<1-45>] [<1-20000> <1-20000> <1-255>] [<1-45>] no bgp dampening	启动 BGP 路由衰减并使用设定的参数，其 no 形式停止路由衰减。

15. 配置 BGP 的能力协商

命令	解释
BGP 协议配置模式	

neighbor {<ip-address> <TAG>} capability {dynamic route-refresh} no neighbor {<ip-address> <TAG>} capability {dynamic route-refresh} neighbor {<ip-address> <TAG>} capability orf prefix-list {<both> <send> <receive>} no neighbor {<ip-address> <TAG>} capability orf prefix-list {<both> <send> <receive>} neighbor {<ip-address> <TAG>} dont-capability-negotiate no neighbor {<ip-address> <TAG>} dont-capability-negotiate neighbor {<ip-address> <TAG>} override-capability no neighbor {<ip-address> <TAG>} override-capability neighbor {<ip-address> <TAG>} strict-capability-match no neighbor {<ip-address> <TAG>} strict-capability-match	BGP 提供能力协商机制，在建立连接时进行能力的匹配。目前支持的能力包括路由刷新、动态能力、出路由过滤（ORF）能力以及地址族支持协商的能力。使用这些命令来使能这些能力，其 NO 形式关闭这些能力。也可以通过命令来配置不进行能力协商，进行严格的能力协商或不管协商的结果。

16. 配置路由服务器

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} route-server-client no neighbor {<ip-address> <TAG>} route-server-client	路由服务器可以在 EBGp 环境下集中配置 BGP 邻居，从而减少每个客户配置的对等体的个数，本命令配置本机为路由服务器并指定其服务的客户，其 NO 形式可删除客户。

17. 配置选路规则

命令	解释
BGP 协议配置模式	

bgp always-compare-med no bgp always-compare-med bgp bestpath as-path ignore no bgp bestpath as-path ignore bgp bestpath compare-confed-aspath no bgp bestpath compare-confed-aspath bgp bestpath compare-routerid no bgp bestpath compare-routerid bgp bestpath med {[confed] [missing-is-worst]} no bgp bestpath med {[confed] [missing-is-worst]}	BGP 可以通过设置改变一些选路规则，从而达到改变最优选择的目的，可以通过这些命令使在 EBGp 环境下比较 MED，忽视 AS-PATH 的长度，比较联盟的 as-path 长度，比较路由器标识，比较联盟 MED 等规则，通过其 NO 形式恢复默认的选路规则。

18. 配置 BGP 引入不同进程 OSPF 路由

(1) 启动 BGP 引入不同进程 OSPF 路由功能

命令	解释
Router bgp 配置模式	
redistribute ospf [<process-id>] [route-map<word>] no redistribute ospf [<process-id>]	启动或者关闭 BGP 引入其他 OSPF 进程路由功能。

(2) 显示和调试 BGP 引入不同进程 OSPF 路由功能相关信息

命令	解释
特权和配置模式	
show ip bgp redistribute	显示 BGP 进程引入其它外部路由的配置信息。
特权模式	
debug bgp redistribute message send no debug bgp redistribute message send debug bgp redistribute route receive no debug bgp redistribute route receive	打开或关闭 BGP 引入其他 OSPF 进程路由的命令下发调试开关。 打开或关闭 BGP 进程收到 nsm 发来的路由信息的调试开关。

4.5.3 BGP 典型案例

4.5.3.1 案例一：BGP 邻居配置

SwitchB、SwitchC和SwitchD位于AS200中，SwitchA位于AS100中。SwitchA和SwitchB共享一个相同的网段11.0.0.0。而SwitchB和SwitchD彼此物理上不相邻。

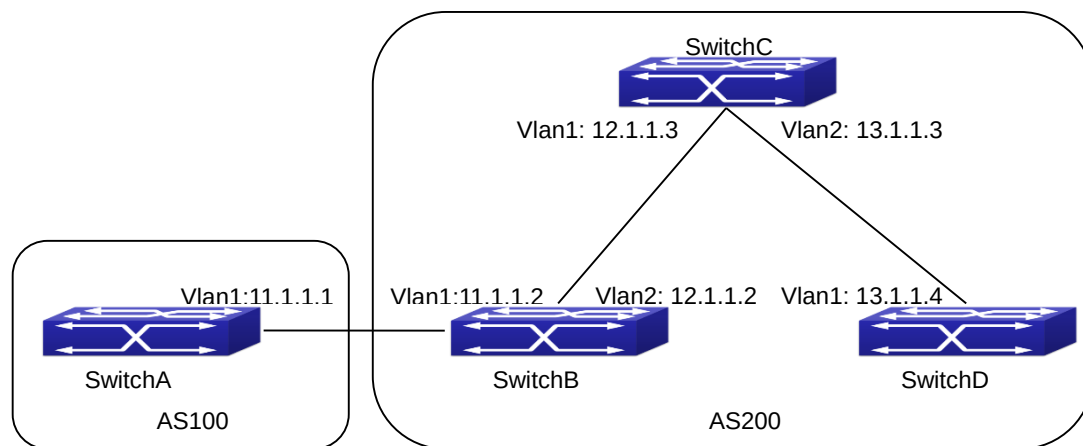


图 4-10 BGP 网络拓扑图

则SwitchA的配置如下：

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
SwitchA(config-router-bgp)#exit
```

SwitchB的配置如下：

```
SwitchB(config)#router bgp 200
SwitchB(config-router-bgp)#network 11.0.0.0
SwitchB(config-router-bgp)#network 12.0.0.0
SwitchB(config-router-bgp)#network 13.0.0.0
SwitchB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
SwitchB(config-router-bgp)#neighbor 12.1.1.3 remote-as 200
SwitchB(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchB(config-router-bgp)#exit
```

SwitchC的配置如下：

```
SwitchC(config)#router bgp 200
SwitchC(config-router-bgp)#network 12.0.0.0
SwitchC(config-router-bgp)#network 13.0.0.0
SwitchC(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
```

```
SwitchC(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchC(config-router-bgp)#exit
```

RouteD的配置如下：

```
SwitchD(config)#router bgp 200
SwitchD(config-router-bgp)#network 13.0.0.0
SwitchD(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
SwitchD(config-router-bgp)#neighbor 13.1.1.3 remote-as 200
SwitchD(config-router-bgp)#exit
```

此时SwitchB和SwitchA之间的连接是EBGP，和SwitchC、SwitchD之间的连接是IBGP。

SwitchB和SwitchD之间不必物理上相连就可以进行BGP的连接，但前提是两个交换机必须有到达对方的路由。该路由可以通过静态路由或是IGP获得。

4.5.3.2 案例二：BGP 聚合配置

在这个例子中，设置路由聚合。首先使用redistribute命令将静态路由重分配到BGP路由表中：

```
SwitchB(config)#ip route 193.0.0.0/24 11.1.1.1
SwitchB(config)#router bgp 100
SwitchB(config-router-bgp)#redistribute static
```

当路由表中至少有一条路由属于指定范围时，下面的配置就在BGP路由表中创建一个聚合路由。聚合路由将被认为来自本机产生的AS。而关于193.0.0.0的更具体的路由信息也进行通告：

```
SwitchB(config)#router bgp 100
SwitchB(config-router-bgp)#aggregate 193.0.0.0/16
```

此时也可以将上面的aggregate命令改为下面的命令，则此时该交换机只通告聚合路由193.0.0.0，而禁止把更具体的路由通告给所有邻居：

```
SwitchB(config-router-bgp)#aggregate 193.0.0.0/16 summary-only
```

4.5.3.3 案例三：配置 BGP 团体属性

下面的例子中，route map set-community用于到邻居16.1.1.6的出站更新。此时通过访问列表1的路由设置特殊的团体属性值“1111”，其它的路由进行正常通告。

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map set-community out
Switch(config-router-bgp)#exit
```

```
Switch(config)#route-map set-community permit 10
Switch(config-route-map)#match ip address 1
Switch(config-route-map)#set community 1111
Switch(config-route-map)#exit
Switch(config)#route-map set-community permit 20
Switch(config-route-map)#match ip address 2
Switch(config-route-map)#exit
Switch(config)#access-list 1 permit 11.1.0.0 0.0.255.255
Switch(config)#access-list 2 permit 0.0.0.0 255.255.255.255
Switch(config)#exit
```

```
Switch#clear ip bgp 16.1.1.6 soft out
```

下面的例子中，根据路由的团体属性值有选择性地设置来自邻居16.1.1.6的路由的MED和本地优先级值。所有同团体列表com1匹配的路由都设置MED为2000，团体列表com1允许团体值带有“100 200 300”或“900 901”的路由通过。同时该路由也可能具有其它团体属性。所有通过团体列表com2的路由都设置本地优先级值为500。而如果既不能通过“com1”，也不能通过“com2”的团体列表的路由将被拒绝。

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map match-community in
Switch(config-router-bgp)#exit
Switch(config)#route-map match-community permit 10
Switch(config-route-map)#match community com1
Switch(config-route-map)#set metric 2000
Switch(config-route-map)#exit
Switch(config)#route-map match-community permit 20
Switch(config-route-map)#match community com2
Switch(config-route-map)#set local-preference 500
Switch(config-route-map)#exit
Switch(config)#ip community-list com1 permit 100 200 300
Switch(config)#ip community-list com1 permit 900 901
Switch(config)#ip community-list com2 permit 88
Switch(config)#ip community-list com2 permit 90
Switch(config)#exit
Switch#clear ip bgp 16.1.1.6 soft out
```

4.5.3.4 案例四：BGP 联盟配置

下面是一个自治系统联盟的配置。如图所示，SWB、SWC 建立 IBGP 连接，属于自治系统 10；SWD 属于自治系统 20；SWB 与 SWD 建立自治系统联盟内部 EBGP 连接；AS10、AS20 组成自治系统联盟，自治系统号为 AS200；SWA 属于自治系统 AS100，SWB 通过 SWA 与自治系统 200 建立 EBGP 连接。

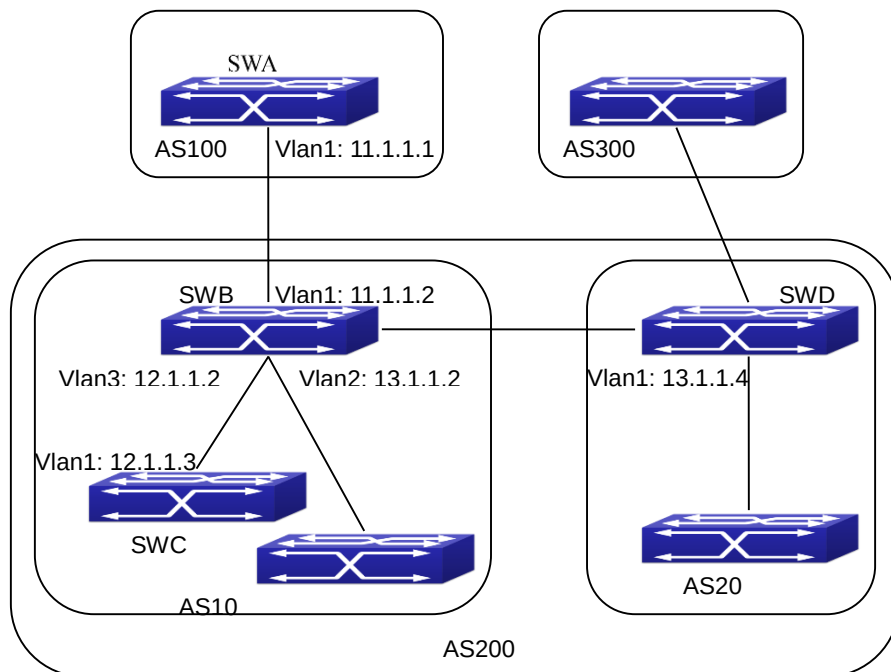


图 4-11 联盟配置拓扑图

配置如下：

SWA:

```
SWA(config)#router bgp 100
```

```
SWA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
```

SWB:

```
SWB(config)#router bgp 10
```

SWB(config-router-bgp)#bgp confederation identifier 200

```
SWB(config-router-bgp)#bgp confederation peers 20
```

```
SWB(config-router-bgp)#neighbor 12.1.1.3 remote-as 10
```

```
SWB(config-router-bgp)#neighbor 13.1.1.4 remote-as 20
```

```
SWB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
```

SWC:

```
SWC(config)#router bgp 10
```

```
SWC(config-router-bgp)#bgp confederation identifier 200
```

```
SWC(config-router-bgp)#bgp confederation peers 20
```

```
SWC(config-router-bgp)#neighbor 12.1.1.2 remote-as 10
```

SWD:

```
SWD(config)#router bgp 20
```

```
SWD(config-router-bgp)#bgp confederation identifier 200
```

```
SWD(config-router-bgp)#bgp confederation peers 10
```

```
SWD(config-router-bgp)#neighbor 13.1.1.2 remote-as 10
```

4.5.3.5 案例五：BGP 路由反射器配置

下面是一个路由反射器的配置，如图所示，SWA、SWB、SWC、SWD、SWE、SWF 和 SWG 建立 IBGP 连接，属于 AS100。SWC 和 AS200 建立 EBGP 连接。SWA 和 AS300 建立 EBGP 连接。SWC、SWD 和 SWG 作路由反射器。

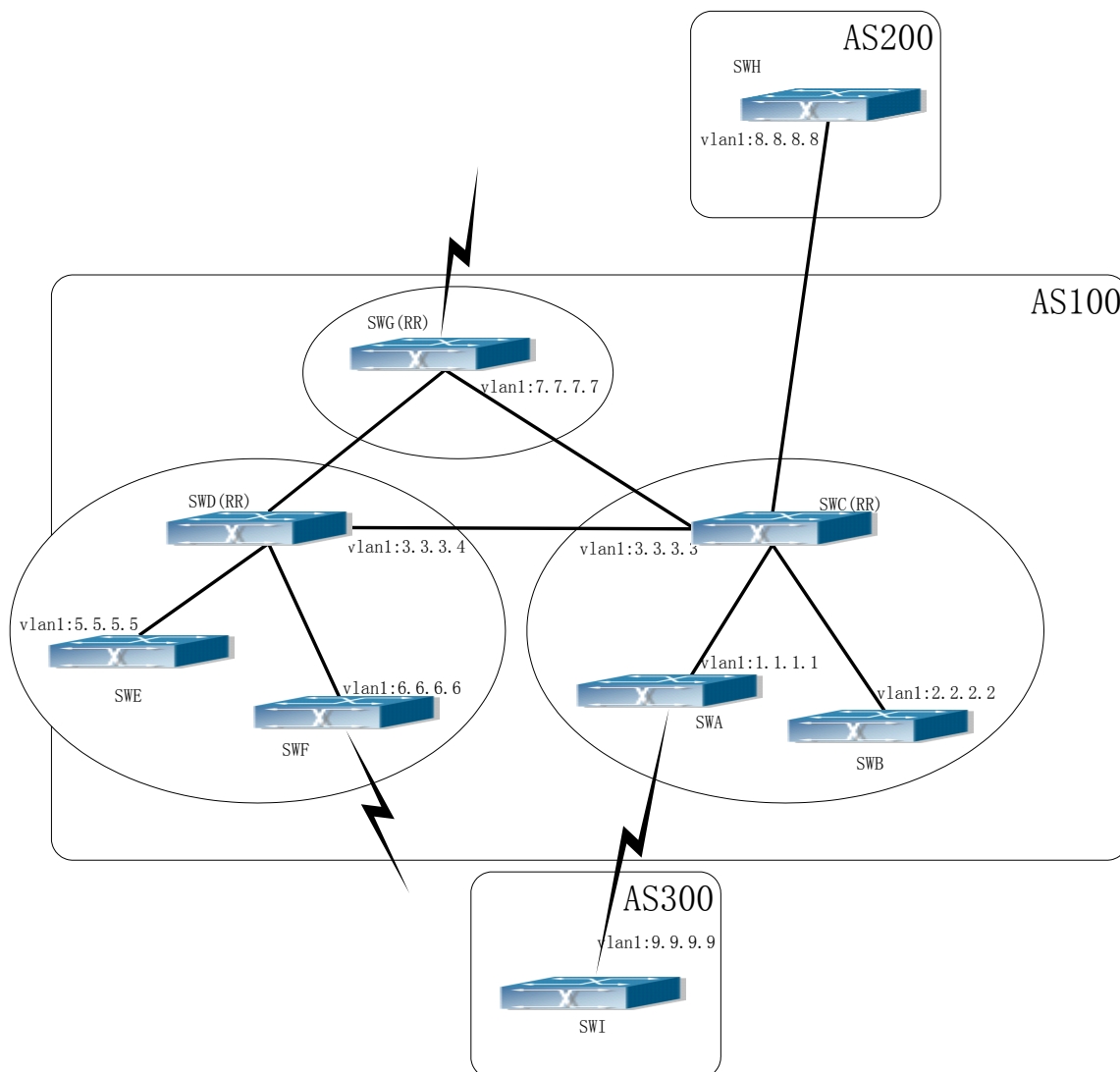


图 4-12 路由反射器拓扑图

配置如下：

SWC 的配置：

```
SWC(config)#router bgp 100
SWC(config-router-bgp)#neighbor 1.1.1.1 remote-as 100
SWC(config-router-bgp)#neighbor 1.1.1.1 route-reflector-client
SWC(config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SWC(config-router-bgp)#neighbor 2.2.2.2 route-reflector-client
SWC(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
SWC(config-router-bgp)#neighbor 3.3.3.4 remote-as 100
SWC(config-router-bgp)#neighbor 8.8.8.8 remote-as 200
```

SWD 的配置：

```
SWD(config)#router bgp 100
SWD(config-router-bgp)#neighbor 5.5.5.5 remote-as 100
SWD(config-router-bgp)#neighbor 5.5.5.5 route-reflector-client
SWD(config-router-bgp)#neighbor 6.6.6.6 remote-as 100
SWD(config-router-bgp)#neighbor 6.6.6.6 route-reflector-client
SWD(config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SWD(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
```

SWA 的配置：

```
SWA(config)#router bgp 100
SWA(config-router-bgp)#neighbor 1.1.1.2 remote-as 100
SWA(config-router-bgp)#neighbor 9.9.9.9 remote-as 300
```

此时的 SWA 不需要和所有的 AS100 的交换机建立 IBGP 连接，就可以收到本 AS 内的其它交换机的 BGP 路由。

4.5.3.6 案例六：BGP 的 MED 设置

下面是一个 MED 的配置，如图所示，SWA 属于 AS100，SWB 属于 AS400，SWC 和 SWD 属于 AS300。

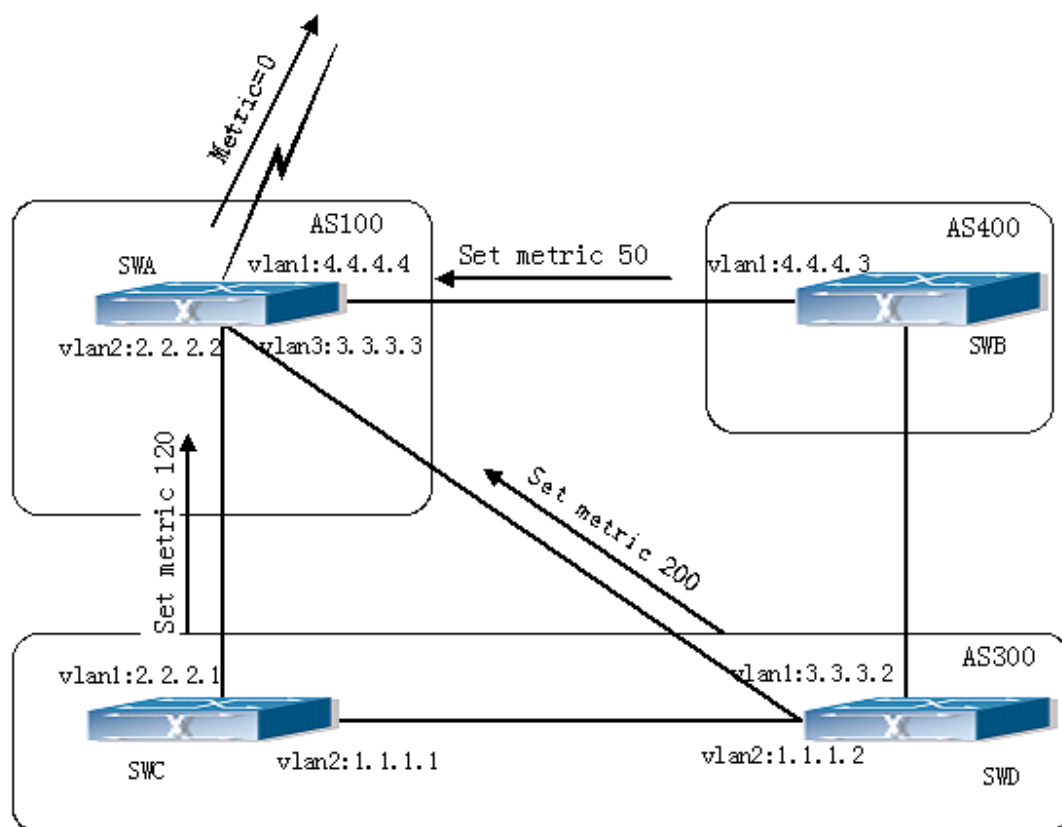


图 4-13 MED 配置拓扑图

SWA 的配置

```
SWA(config)#router bgp 100
SWA(config-router-bgp)#neighbor 2.2.2.1 remote-as 300
SWA(config-router-bgp)#neighbor 3.3.3.2 remote-as 300
SWA(config-router-bgp)#neighbor 4.4.4.3 remote-as 400
```

SWC 的配置

```
SWC(config)#router bgp 300
SWC (config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SWC (config-router-bgp)#neighbor 2.2.2.2 route-map set-metric out
SWC (config-router-bgp)#neighbor 1.1.1.2 remote-as 300
SWC (config-router-bgp)#exit
SWC (config)#route-map set-metric permit 10
SWC (Config-Router-RouteMap)#set metric 120
```

SWD 的配置

```
SWD (config)#router bgp 300
```

```
SWD (config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SWD (config-router-bgp)#neighbor 3.3.3.3 route-map set-metric out
SWD (config-router-bgp)#neighbor 1.1.1.1 remote-as 300
SWD (config-router-bgp)#exit
SWD (config)#route-map set-metric permit 10
SWD (Config-Router-RouteMap)#set metric 200
```

SWB 的配置

```
SWB (config)#router bgp 400
SWB (config-router-bgp)#neighbor 4.4.4.4 remote-as 100
SWB (config-router-bgp)#neighbor 4.4.4.4 route-map set-metric out
SWB (config-router-bgp)#exit
SWB (config)#route-map set-metric permit 10
SWB (Config-Router-RouteMap)#set metric 50
```

经过上面的配置以后,假设 SWB、SWC 和 SWD 都向 SWA 发送了一条路由 12.0.0.0。根据 BGP 路由策略的比较,假设在进行 metric 属性比较之前,从上述三个交换机发出的该条路由的所有属性都相同。此时,metric 值较小的将是更优路径。但交换机对 metric 属性的比较将只在来自同一 AS 的路由上进行比较。则此时对于 SWA 来说,通过 SWC 的路径优于通过 SWD 的路径。而由于此时 SWC 和 SWB 位于不同的 AS,所以 SWA 将不在两个交换机之间进行 metric 值。如果要使 metric 值的比较在不同的 AS 之间也进行那么需要用到“bgp always-compare-med”命令。如果配置了该命令,则对于 SWA 来说通过 SWB 的路径是最优的。此时可以在 SWA 上增加下面的配置命令:

```
SWA(config-router-bgp)# bgp always-compare-med
```

4.5.3.7 案例七: BGP VPN 典型案例

在 MPLS VPN 环境下,BGP 作为核心的路由传播工具和在边缘设备上产生 ILM 和 FTN 表项的重要工具,在 DCNOS 上,它和 LDP 协议一起,为 MPLS VPN 的应用提供了核心支持。在这里,LDP 工作在公网上,而在公网的非边缘路由器上,不需要运行 BGP 协议。

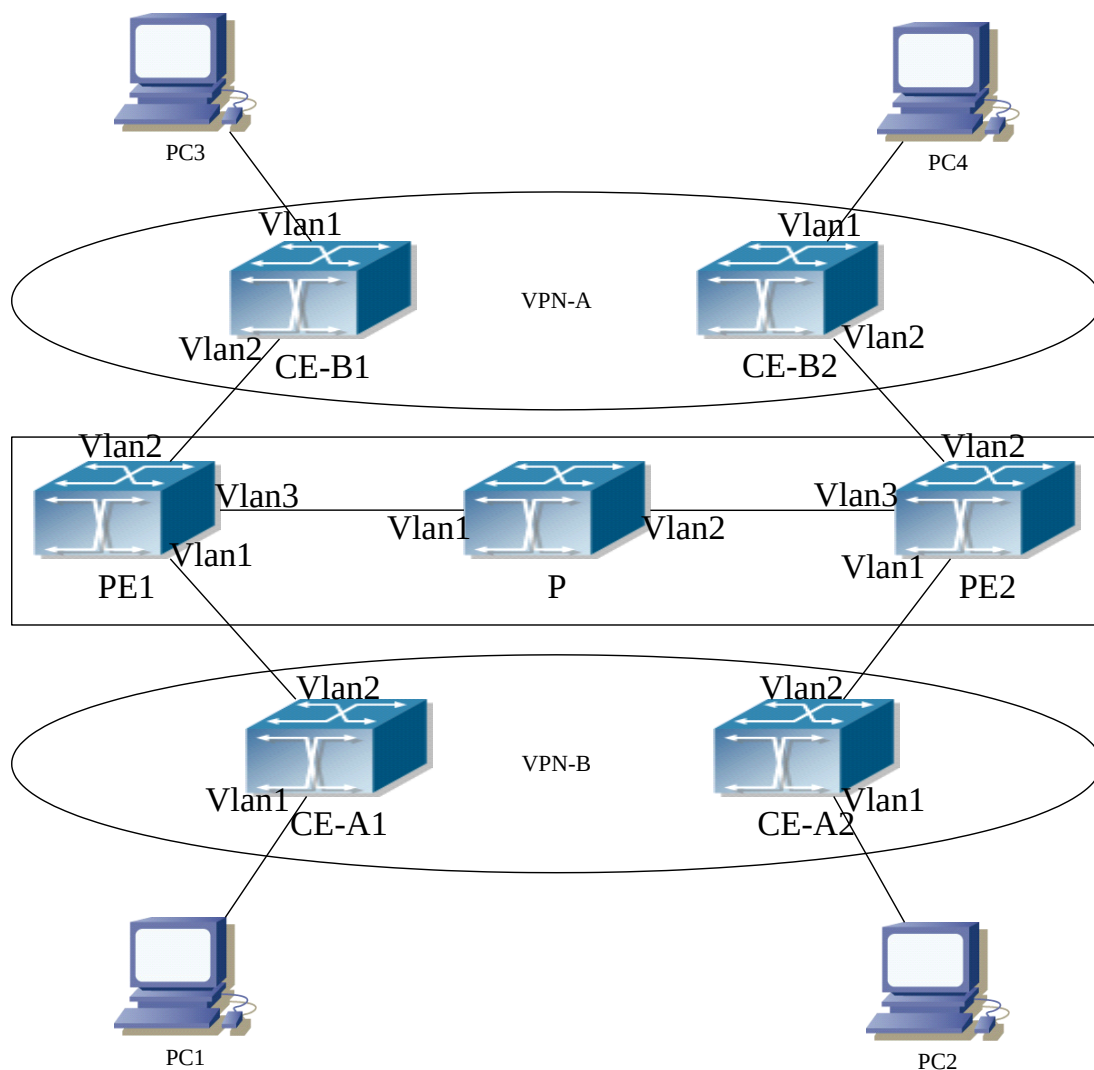


图 4-14 MPLS VPN 典型案例

如图是一个典型的 MPLS VPN 案例，其中 PE1、P、PE2 组成了公网区域，也就是使用 MPLS 交换的区域，CE-A1 与 CE-A2 组成了 VPN-A，CE-B1 与 CE-B2 组成了 VPN-B。两个 VPN 都通过公网的标签交换通信，他们互相不能通信。PE1 和 PE2 是运营商提供的边缘路由器，而 CE-A1、CE-A2、CE-B1、CE-B2 是用户端的接入交换机，PC1-PC4 表示了接入的用户。BGP 分别运行在公网区域和私网区域，其中在公网区域需要支持 VPN 路由，并且使用 LOOPBACK 接口进行连接。

各设备配置如下：

CE-A1 的配置：

```
CE-A1#config
```

```
CE-A1(config)#interface vlan 2
```

```
CE-A1(config-if-Vlan2)#ip address 192.168.101.2 255.255.255.0
```

```
CE-A1(config-if-Vlan2)#exit
```

```
CE-A1(config)#interface vlan 1
```

```
CE-A1(config-if-Vlan2)#ip address 10.1.1.1 255.255.255.0
```

```
CE-A1(config-if-Vlan2)#exit
CE-A1(config)#router bgp 60101
CE-A1(config-router)#neighbor 192.168.101.1 remote-as 100
CE-A1(config-router)#exit
```

CE-A2 的配置:

```
CE-A2#config
CE-A2(config)#interface vlan 2
CE-A2(config-if-Vlan2)#ip address 192.168.102.2 255.255.255.0
CE-A2(config-if-Vlan2)#exit
CE-A2(config)#interface vlan 1
CE-A2(config-if-Vlan2)#ip address 10.1.2.1 255.255.255.0
CE-A2(config-if-Vlan2)#exit
CE-A2(config)#router bgp 60102
CE-A2(config-router)#neighbor 192.168.102.1 remote-as 100
CE-A2(config-router)#exit
```

CE-B1 的配置:

```
CE-B1#config
CE-B1(config)#interface vlan 2
CE-B1(config-if-Vlan2)#ip address 192.168.201.2 255.255.255.0
CE-B1(config-if-Vlan2)#exit
CE-B1(config)#interface vlan 1
CE-B1(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
CE-B1(config-if-Vlan2)#exit
CE-B1(config)#router bgp 60201
CE-B1(config-router)#neighbor 192.168.201.1 remote-as 100
CE-B1(config-router)#exit
```

CE-B2 的配置:

```
CE-B2#config
CE-B2(config)#interface vlan 2
CE-B2(config-if-Vlan2)#ip address 192.168.202.2 255.255.255.0
CE-B2(config-if-Vlan2)#exit
CE-B2(config)#interface vlan 1
CE-B2(config-if-Vlan2)#ip address 20.1.2.1 255.255.255.0
CE-B2(config-if-Vlan2)#exit
CE-B2(config)#router bgp 60202
```

```
CE-B2(config-router)#neighbor 192.168.202.1 remote-as 100
```

```
CE-B2(config-router)#exit
```

PE1 的配置:

```
PE1#config
```

```
PE1(config)#ip vrf VRF-A
```

```
PE1(config-vrf)#rd 100:10
```

```
PE1(config-vrf)#route-target both 100:10
```

```
PE1(config-vrf)#exit
```

```
PE1(config)#ip vrf VRF-B
```

```
PE1(config-vrf)#rd 100:20
```

```
PE1(config-vrf)#route-target both 100:20
```

```
PE1(config-vrf)#exit
```

```
PE1(config)#interface vlan 1
```

```
PE1(config-if-Vlan1)#ip vrf forwarding VRF-A
```

```
PE1(config-if-Vlan1)#ip address 192.168.101.1 255.255.255.0
```

```
PE1(config-if-Vlan1)#exit
```

```
PE1(config)#interface vlan 2
```

```
PE1(config-if-Vlan2)#ip vrf forwarding VRF-B
```

```
PE1(config-if-Vlan2)#ip address 192.168.201.1 255.255.255.0
```

```
PE1(config-if-Vlan2)#exit
```

```
PE1(config)#interface vlan 3
```

```
PE1(config-if-Vlan3)#ip address 202.200.1.2 255.255.255.0
```

```
PE1(config-if-Vlan3)#label-switching
```

```
PE1(config-if-Vlan3)#exit
```

```
PE1(config)#interface loopback 1
```

```
PE1(Config-if-Loopback1)# ip address 200.200.1.1 255.255.255.255
```

```
PE1(config-if-Vlan3)#exit
```

```
PE1(config)#router bgp 100
```

```
PE1(config-router)#neighbor 200.200.1.2 remote-as 100
```

```
PE1(config-router)#neighbor 200.200.1.2 update-source 200.200.1.1
```

```
PE1(config-router)#address-family vpnv4 unicast
```

```
PE1(config-router-af)#neighbor 200.200.1.2 activate
```

```
PE1(config-router-af)#exit-address-family
```

```
PE1(config-router)#address-family ipv4 vrf VRF-A
```

```
PE1(config-router-af)# neighbor 192.168.101.2 remote-as 60101
```

```
PE1(config-router-af)#exit-address-family
```

```
PE1(config-router)#address-family ipv4 vrf VRF-B
```

```
PE1(config-router-af)# neighbor 192.168.201.2 remote-as 60201
```

```
PE1(config-router-af)#exit-address-family
```

PE2 的配置:

```
PE2#config
```

```
PE2(config)#ip vrf VRF-A
```

```
PE2(config-vrf)#rd 100:10
```

```
PE2(config-vrf)#route-target both 100:10
```

```
PE2(config-vrf)#exit
```

```
PE2(config)#ip vrf VRF-B
```

```
PE2(config-vrf)#rd 100:20
```

```
PE2(config-vrf)#route-target both 100:20
```

```
PE2(config-vrf)#exit
```

```
PE2(config)#interface vlan 1
```

```
PE2(config-if-Vlan1)#ip vrf forwarding VRF-A
```

```
PE2(config-if-Vlan1)#ip address 192.168.102.1 255.255.255.0
```

```
PE2(config-if-Vlan1)#exit
```

```
PE2(config)#interface vlan 2
```

```
PE2(config-if-Vlan2)#ip vrf forwarding VRF-B
```

```
PE2(config-if-Vlan2)#ip address 192.168.202.1 255.255.255.0
```

```
PE2(config-if-Vlan2)#exit
```

```
PE2(config)#interface vlan 3
```

```
PE2(config-if-Vlan3)#ip address 202.200.2.2 255.255.255.0
```

```
PE2(config-if-Vlan3)#label-switching
```

```
PE2(config-if-Vlan3)#exit
```

```
PE2(config)#interface loopback 1
```

```
PE2(Config-if-Loopback1)# ip address 200.200.1.2 255.255.255.255
```

```
PE2(config-if-Vlan3)#exit
```

```
PE2(config)#router bgp 100
```

```
PE2(config-router)#neighbor 200.200.1.1 remote-as 100
```

```
PE2(config-router)#address-family vpnv4 unicast
```

```
PE2(config-router-af)#neighbor 200.200.1.1 activate
```

```
PE2(config-router-af)#exit-address-family
```

```
PE2(config-router)#address-family ipv4 vrf VRF-A
```

```
PE2(config-router-af)# neighbor 192.168.102.2 remote-as 60102
```

```
PE2(config-router-af)#exit-address-family
```

```
PE2(config-router)#address-family ipv4 vrf VRF-B
```

```
PE2(config-router-af)# neighbor 192.168.202.2 remote-as 60202
```

```
PE2(config-router-af)#exit-address-family
```

上面的例子是一种比较典型的情况，如果需要 VRF 间可以通信，可以通过修改 route-target 的方式实现。如果两端的 BGP AS 号相同，还需要在 PE 上使用 neighbor <ip-addr>

as-override 命令来避免 AS 号的重复。

这里只列出了 BGP 相关的配置，公网上运行的 LDP 协议的相关配置请参考 LDP 典型案例。

4.5.4 BGP 排错帮助

在配置、使用 BGP 协议时，可能会由于物理连接、配置错误等原因导致 BGP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误。
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）。
- ☞ 然后，先启动 BGP 协议（使用 router bgp 命令），再配置相应的 IBGP 和 EBGP 邻居（使用 neighbor remote-as 命令）。
- ☞ 需注意 BGP 协议本身并不能发现路由，BGP 路由的产生需要引入其他的路由。只有引入了路由才能将这些路由通告到 IBGP 和 EBGP 邻居上。这些引入的路由包括直连路由、静态路由、IGP 路由（RIP 和 OSPF）。引入这些路由的方法可以通过 network 命令和 redistribute（BGP）命令。
- ☞ 对于 BGP 来说，IBGP 和 EBGP 的行为不同需要注意。
- ☞ 当配置完成后，可以使用 show ip bgp summary 命令察看邻居的连接情况，应确保所有的邻居都保持 BGP 连接状态。并使用 show ip bgp 命令察看 BGP 路由表，使用 show ip route 察看 IP 路由表。
- ☞ 如果仍无法解决 BGP 的路由问题，那么请使用 debug ip bgp 等调试命令，然后将 3 分钟内的 debug 信息拷贝下来，发送给本公司技术服务中心。

4.6 IPv4 黑洞路由

4.6.1 黑洞路由介绍

黑洞路由实际是一种特殊的静态路由，顾名思义，就是目的地址为该网段的数据报文到达设备之后，将被丢弃。

4.6.2 IPv4 黑洞路由配置任务

1. 配置IPv4黑洞路由

1. 配置IPv4黑洞路由

命令	解释

全局配置模式	
<pre>ip route {<ip-prefix> <mask> <ip-prefix> <prefix-length>} null0 [<distance>] no ip route {<ip-prefix> <mask> <ip-prefix> <prefix-length>} null0</pre>	配置静态黑洞路由。本命令的no操作作为删除配置的黑洞路由。

4.6.3 黑洞路由举例

案例：IPv4黑洞路由功能。

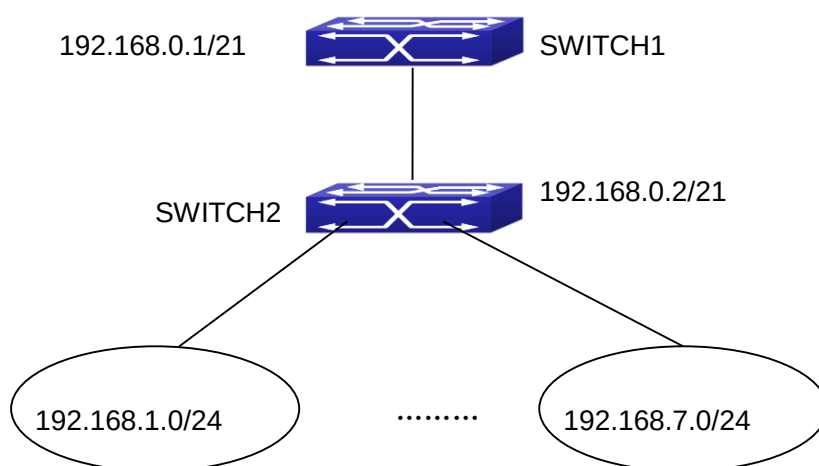


图 4-15 配置 IPv4 黑洞路由的示例图

如图所示，交换机 SWITCH2 配置了 8 个 VLAN 三层接口用于用户接入，分别是 192.168.1.0/24 ~ 192.168.7.0/24，同时 SWITCH2 配置了一条缺省路由指向路由器 SWITCH1，SWITCH1 配置回程路由 192.168.0.0/21 指回到 S。正常情况下这个网络不会出现问题，但是 SWITCH2 的一个三层接口 down 掉之后，例如到网段 192.168.1.0/24 的三层接口 down 掉了，有数据包发往某 VLAN1 时，但数据到达 SWITCH2 之后，没有到这个网段的路由，那么会根据缺省路由送到 SWITCH1，SWITCH1 在接受到 IP 报文之后，查询路由表，匹配上直连路由，然后又送到 SWITCH2，如此往复，就在 SWITCH2 和 SWITCH1 之间循环转发，形成路由环路。为了解决这个问题，可以在 SWITCH2 上配置一条“黑洞路由”：

```
ip route 192.168.0.0/21 null0 50
```

这时，当到 SWITCH2 收到 interface vlan1 报文之后，查询路由表，匹配上这条黑洞路由，那么会丢弃该报文，便不会在 SWITCH2 和 SWITCH1 上循环转发。

配置步骤如下：

Switch#config

```
Switch(config)#ip route 192.168.0.0/21 null0 50
```

4.6.4 黑洞路由排错帮助

在配置、使用黑洞路由功能时，可能会因为掩码长度不合适，以及管理距离不合适而造成无法正常使用。因此，用户应注意以下要点：

使用IPv6黑洞路由时，请首先确保全局启动了IPv6功能。

☞ 掩码长度最好比正常的路由的掩码长度长，这样可以确保当掩码长度长的正常路由有效的时候不干扰其工作。

☞ 当掩码长度与其他路由一样时，建议把黑洞路由的distance调的低一点。

如果使用检查都尝试仍无法解决的问题，那么请使用 **show ip route database**，以及 **show ip route fib**，**show I3** 等命令，然后将这些信息拷贝下来，发送给本公司技术服务中心。

4.7 GRE

4.7.1 GRE 隧道介绍

GRE(通用路由协议封装)是由 Cisco 和 Net-smiths 等公司于 1994 年提交给 IETF 的，标号为 RFC1701 和 RFC1702。目前有多数厂商的网络设备均支持 GRE 隧道协议。GRE 规定了如何用一种网络协议去封装另一种网络协议的方法。GRE 的隧道由两端的源 IP 地址和目的 IP 地址来定义，允许用户使用 IP 包封装 IP、IPX、AppleTalk 包，并支持全部的路由协议(如 RIP2、OSPF 等)。通过 GRE，用户可以利用公共 IP 网络连接 IPX 网络、AppleTalk 网络，还可以使用保留地址进行网络互连，或者对公网隐藏企业网的 IP 地址。GRE 只提供了数据包的封装，它并没有加密功能来防止网络侦听和攻击。所以在实际环境中它常和 IPsec 在一起使用，由 IPsec 提供用户数据的加密，从而给用户提供更好的安全性。

GRE 协议的主要用途有两个：企业内部协议封装和私有地址封装。在国内，由于企业网几乎全部采用的是 TCP/IP 协议，因此在中国建立隧道时没有对企业内部协议封装的市场需求。企业使用 GRE 的唯一理由应该是对内部地址的封装。我们交换机应用 GRE 主要是因为互联网协议的过渡(包括 IPv6 OVER IPv4 和 IPv4 OVER IPv6)。

本文工作根据 RFC1701、1702、2784 实现。

4.7.2 GRE 隧道基本配置

GRE 隧道配置任务序列如下：

8.
 1. 配置隧道模式
 - 1) 配置隧道模式为 GREv4 隧道
 9. 2) 配置隧道模式为 GREv6 隧道
10.
 2. 配置 GRE 隧道的源地址和目的地址

- 1) 配置 GRE 隧道的源地址为 IPv6 或者 IPv4 地址
 11. 2) 配置 GRE 隧道的目的地址为 IPv6 或者 IPv4 地址
 12. 3. 配置 GRE 隧道接口地址
 13. 1) 配置 GRE 隧道接口的 IPv4 地址
 14. 2) 配置 GRE 隧道接口的 IPv6 地址
 15. 4. 配置静态路由的出接口是 GRE 隧道
 16. 1) 配置 IPv4 静态路由的出接口是 GRE 隧道
 17. 2) 配置 IPv6 静态路由的出接口是 GRE 隧道
 - 18.
1. 配置隧道模式

命令	解释
隧道接口配置模式	
tunnel mode gre ip no tunnel mode	配置隧道模式为 GREv4 隧道，数据报文 GRE 封装后是 IPv4 报文头，穿越 IPv4 网络。
tunnel mode gre ipv6 no tunnel mode	配置隧道模式为 GREv6 隧道，数据报文 GRE 封装后是 IPv6 报文头，穿越 IPv6 网络。

2. 配置 GRE 隧道的源地址和目的地址

命令	解释
隧道接口配置模式	
tunnel source {<ipv6-address> <ipv4-address>} no tunnel source	配置 GRE 隧道的源地址为 IPv6 或者 IPv4 地址。
tunnel destination {<ipv6-address> <ipv4-address>} no tunnel destination	配置 GRE 隧道的目的地址为 IPv6 或者 IPv4 地址。

3. 配置 GRE 隧道接口地址

命令	解释
隧道接口配置模式	

ip address <ipv4-address> <mask> no ip address <ipv4-address> <mask>	配置 GRE 隧道接口的 IPv4 地址。
ipv6 address <ipv6-address/prefix> no ipv6 address <ipv6-address/prefix>	配置 GRE 隧道接口的 IPv6 地址。

4. 配置静态路由的出接口是 GRE 隧道

命令	解释
全局配置模式	
ip route <ipv4-address/mask> tunnel <ID> no ip route <ipv4-address/mask> tunnel <ID>	配置 IPv4 静态路由的 出接口是 GRE 隧道。
ipv6 route <ipv6-address/prefix> tunnel <ID> no ipv6 route <ipv6-address/prefix> tunnel <ID>	配置 IPv6 静态路由的 出接口是 GRE 隧道。

4.7.3 GRE 隧道举例

GRE 隧道典型配置举例：

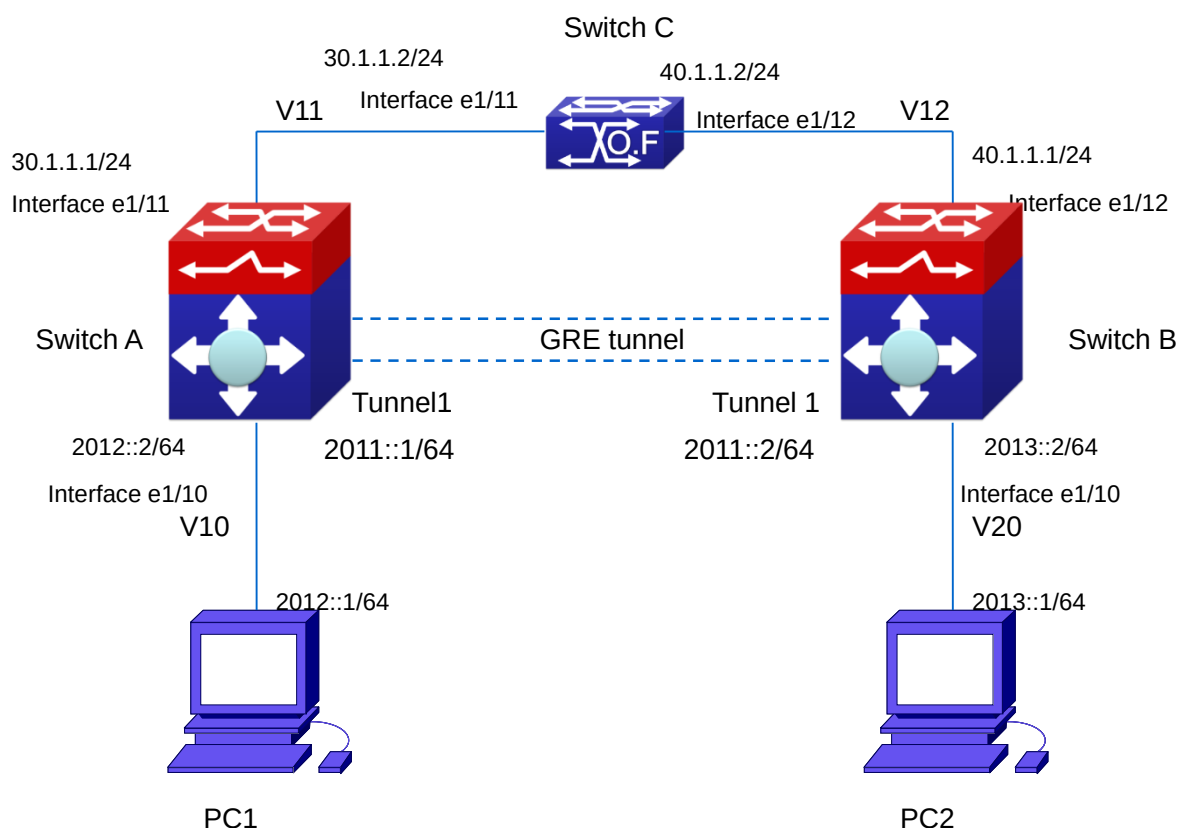


图 4-16 IPv6 over IPv4 GRE 隧道特性典型配置组网图

配置思路

- ☞ 配置 IPv4 网络，保证 IPv4 互通。
- ☞ 配置设备的隧道接口、连接 PC 的业务接口。
- ☞ 配置隧道参数，使能隧道接口。
- ☞ 使能 OSPF 路由协议使得 PC1 和 PC2 间数据通过隧道转发。

配置步骤

说明：本文的组网环境可能与您的实际环境存在差异。为了保证配置效果，请确认设备上现有配置和以下配置不冲突。

（1）设备 A 的配置：

1. 配置步骤

- ☞ 创建业务 VLAN 11 及其接口地址。

```
SwitchA(config)#vlan 11
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.1/24
```

- ☞ 配置从 SwitchA 到 SwitchB 上接口 Vlan11 的 IPv4 静态路由。

```
SwitchA(config)#ip route 40.1.1.1/24 30.1.1.2
```

- ☞ 配置隧道接口及类型和源端、目的端。当隧道起来以后，隧道的源和目的不允许修改，除非隧道源为三层接口时，只有在更改三层接口地址的时候允许更新隧道源

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ip	30.1.1.1	40.1.1.1

可以看到 GRE 隧道已经配置成功。

- ☞ 配置隧道接口 IPv6 地址。对于隧道接口配置 IPv4/IPv6 地址只允许配置一个接口地址，该限制同样适用于其它模式的隧道，如配置隧道、6to4、isatap 等。

注意：配置 IPv4 地址的时候由于实现的原因必须要求隧道 active，存在配置保留后丢失 IPv4 地址配置的可能，这个和 IPv6 地址的配置逻辑不一致，IPv6 不要求隧道必须 active。

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::1/64
```

- ☞ 配置业务 VLAN10 及其接口地址。

```
SwitchA(config)#vlan 10
SwitchA(config-vlan10)#switchport interface ethernet 1/10
```

```
SwitchA(config-vlan10)#exit
SwitchA(config)#interface vlan 10
SwitchA(config-if-vlan10)# ipv6 address 2012::2/64
SwitchA(config-if-vlan10)#exit
```

☞ 配置 OSPF 路由协议。

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#network 30.1.1.1/24 area 0
SwitchA(config-router)#exit
```

(2) 设备 B 的配置:

1. 配置步骤

☞ 创建业务 VLAN 12 及其接口地址。

```
SwitchA(config)#vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 40.1.1.1/24
SwitchA(config-if-vlan12)#exit
SwitchA(config)#
```

☞ 配置从 SwitchB 到 SwitchA 上接口 Vlan12 的 IPv4 静态路由。

```
SwitchA(config)#ip route 30.1.1.1/24 40.1.1.2
```

☞ 配置隧道接口及类型和源端、目的端。

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ip	40.1.1.1	30.1.1.1

可以看到 GRE 隧道已经配置成功。

☞ 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 OSPF 路由协议。

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::2/64
```

☞ 配置业务 VLAN20 及其接口地址。

```
SwitchA(config)#vlan 20
SwitchA(config-vlan20)#switchport interface ethernet 1/10
SwitchA(config-vlan20)#exit
SwitchA(config)#interface vlan 20
```

```
SwitchA(config-if-vlan20)# ipv6 address 2013::2/64
SwitchA(config-if-vlan20)#exit
SwitchA(config)#
```

☞ 配置 OSPF 路由协议。

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.2
SwitchA(config-router)#network 40.1.1.0/24 area 0
SwitchA(config-router)#exit
SwitchA(config)#
```

（3）设备 C 的配置

1. 配置步骤

☞ 创建业务 VLAN 11 及其接口地址。

```
SwitchA(config)#vlan 11
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.2/24
```

☞ 创建业务 VLAN 12 及其接口地址。

```
SwitchA(config)#vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 40.1.1.2/24
SwitchA(config-if-vlan12)#exit
```

（4）PC 的配置

☞ 配置 PC1 的网卡地址及默认网关。

PC1：网卡地址：2012::1/64，默认网关：2012::2

PC2：网卡地址：2013::1/64，默认网关：2013::2

4.7.4 GRE 隧道引用业务环回组举例

业务环回组功能简介：

若同一台设备存在多种业务单板混插的情况，需要通过业务环回功能来实现不同业务单板间的业务重定向。例如支持 GRE 隧道和不支持 GRE 隧道单板混插，此时就需要通过业务环回功能将不支持 GRE 隧道单板收到的 GRE 隧道数据环回到支持 GRE 隧道的单板上进行处理。业务环回组需要占用至少一个业务单板上的闲置端口，这些端口在加入环

回组之前不能有任何配置。为了增加板间业务重定向的带宽，可以将多个端口加入到业务环回组中，这一组端口构成环回组，进行流量负载分担。

GRE 隧道引用业务环回组典型配置举例：

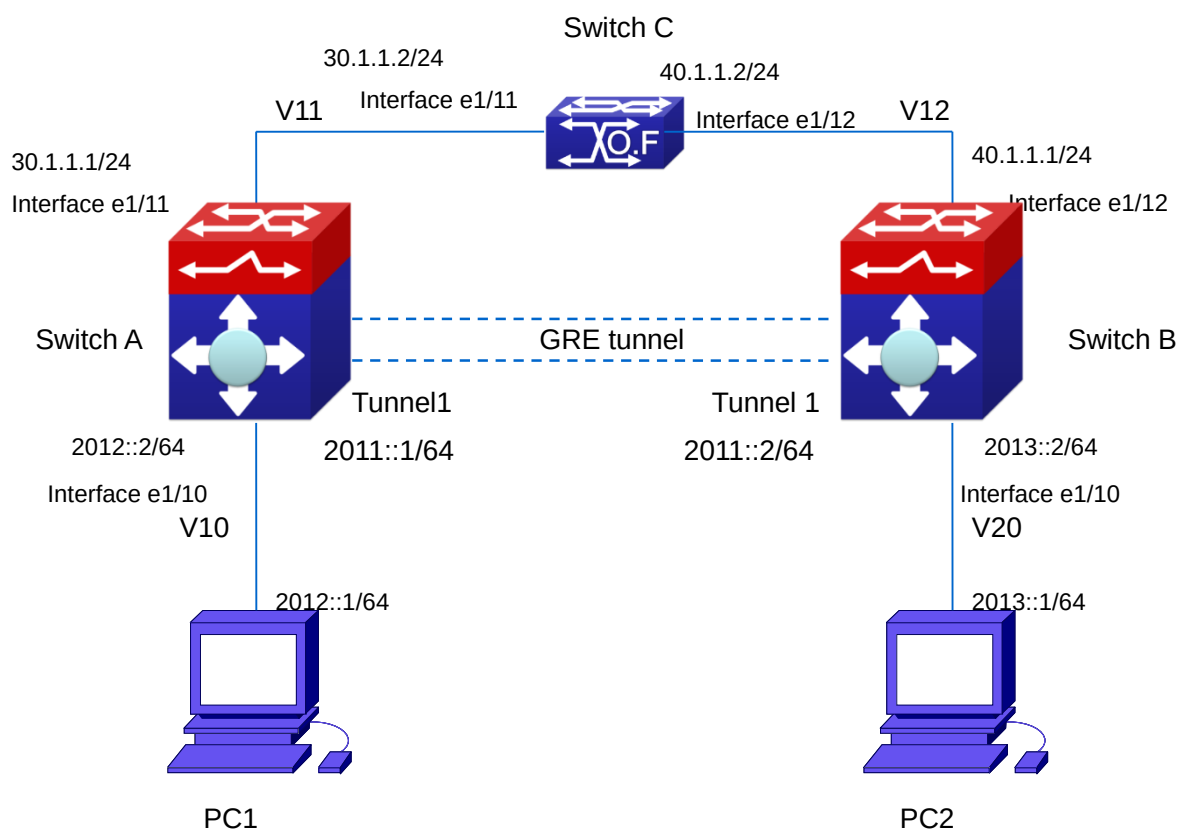


图 4-17 GRE 隧道引用业务环回组典型配置组网图

环回组组网图介绍

SwitchA 和 SwitchB 之间为 ipv4 网络，PC1 和 PC2 处在 IPv6 网络。如果 PC1 需要和远端的 PC2 通信，必须通过 GRE 隧道穿越 SwitchA 和 SwitchB 之间的 ipv4 网络。SwitchA 存在多种业务单板混插的情况：单板 1 不支持 GRE 隧道功能，单板 3 支持 GRE 隧道，因此单板 1 收到 PC1 发往 PC2 的数据需要通过业务环回组功能环回到单板 3 上进行处理。

配置思路

- ☞ 配置 IPv4 网络，保证 IPv4 互通。
- ☞ 配置设备的隧道接口、连接 PC 的业务接口。
- ☞ 配置隧道参数，使能隧道接口。
- ☞ 配置环回组，把单板 3 上的闲置端口 1/12 加入此环回组并让隧道引用该环回组。
- ☞ 使能 OSPF 路由协议使得 PC1 和 PC2 间数据通过隧道转发。

配置步骤

说明：本文的组网环境可能与您的实际环境存在差异。为了保证配置效果，请确认设备

上现有配置和以下配置不冲突。

(1) 设备 A 的配置

1. 配置步骤

- 创建业务 VLAN 11 及其接口地址

```
SwitchA(config)#vlan 11
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.1/24
```

- 配置从 SwitchA 到 SwitchB 上接口 Vlan11 的 IPv4 静态路由

```
SwitchA(config)#ip route 40.1.1.1/24 30.1.1.2
```

- 配置隧道接口及类型和源端、目的端

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ipv6	30.1.1.1	40.1.1.1

可以看到 GRE 隧道已经配置成功。

- 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 ospf 路由协议

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::1/64
```

- 配置业务 VLAN10 及其接口地址

```
SwitchA(config)#vlan 10
SwitchA(config-vlan10)#switchport interface ethernet 1/10
SwitchA(config-vlan10)#exit
SwitchA(config)#interface vlan 10
SwitchA(config-if-vlan10)# ipv6 address 2012::2/64
SwitchA(config-if-vlan10)#exit
```

- 配置环回组并让隧道引用该环回组

```
SwitchA (config)#loopback-group 1
SwitchA (config-if-ethernet1/12)#loopback-group 1
SwitchA (config-if-tunnel1)# loopback-group 1
```

- 配置 OSPF 路由协议

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#network 30.1.1.0/24 area 0
SwitchA(config-router)#exit
```

(2) 设备 B 的配置

1. 配置步骤

- 创建业务 VLAN 12 及其接口地址

```
SwitchA(config)#vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 30.1.1.2/24
SwitchA(config-if-vlan12)#exit
SwitchA(config)#
```

- 配置从 SwitchB 到 SwitchA 上接口 Vlan12 的 IPv4 静态路由

```
SwitchA(config)#ip route 30.1.1.1/24 40.1.1.2
```

- 配置隧道接口及类型和源端、目的端

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ipv6	40.1.1.1	30.1.1.1

可以看到 GRE 隧道已经配置成功。

- 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 ospf 路由协议

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::2/64
```

- 配置业务 VLAN20 及其接口地址

```
SwitchA(config)#vlan 20
SwitchA(config-vlan20)#switchport interface ethernet 1/10
SwitchA(config-vlan20)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-vlan20)# ipv6 address 2013::2/64
SwitchA(config-if-vlan20)#exit
SwitchA(config)#
```

- 配置 OSPF 路由协议

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.2
SwitchA(config-router)#network 40.1.1.0/24 area 0
SwitchA(config-router)#exit
SwitchA(config)#
```

（3）设备 C 的配置

1. 配置步骤

- ☞ 创建业务 VLAN 11 及其接口地址

```
SwitchA(config)#vlan 11
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.2/24
```

- ☞ 创建业务 VLAN 12 及其接口地址

```
SwitchA(config)#vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 40.1.1.2/24
SwitchA(config-if-vlan12)#exit
```

（4）PC 的配置

- ☞ 配置 PC1 的网卡地址及默认网关。

PC1: 网卡地址: 2012::1/64, 默认网关: 2012::2

PC2: 网卡地址: 2013::1/64, 默认网关: 2013::2

4.7.5 GRE 隧道排错帮助

当在使用 GRE 隧道过程中遇到问题, 请检查是否是如下原因:

- ☞ 检查配置, 是否正确配置了隧道的源和目的地址, 配置隧道模式 (tunnel mode gre {ip | ipv6}) 是否正确。
- ☞ 在交换机配置了隧道后, 检查下一跳接口为 GRE 隧道接口的静态路由配置。
- ☞ 交换机之间的连线是否正常, 可通过 debug gre {packet | event | all} 观察交换机能否正确接收和处理相关的报文。

4.8 ECMP

4.8.1 ECMP 简介

ECMP (Equal-cost Multi-path Routing) 等价多路径路由, 在存在多条不同链路到达同

一目的地址的网络环境中，如果使用传统的路由技术，发往该目的地址的数据包只能利用其中的一条链路，其它链路处于备份状态或无效状态，并且在动态路由环境下相互的切换需要一定时间，而等价多路径路由协议可以在这样的网络环境下同时使用多条链路，不仅实现了流量分担，增加了传输带宽，并且可以无时延无丢包地备份失效链路的数据传输。



图 4-18 ECMP 的应用环境

如图所示，R1 到 R4 有两条路径可以选择，分别是 R1-R2-R4 和 R1-R3-R4，如果假设路由类型和代价开销相同，R1 到 R4 可以形成两条路由，目的地都是 R4，但是下一跳不相同，如果这两条路由都被选为最优，它们就形成了等价路由。

4.8.2 ECMP 配置任务序列

1. 配置等价路由的最大数目

1. 配置等价路由的最大数目

命令	解释
全局配置模式	
maximum-paths <1-32> no maximum-paths	配置等价路由的最大数目。

4.8.3 ECMP 典型案例

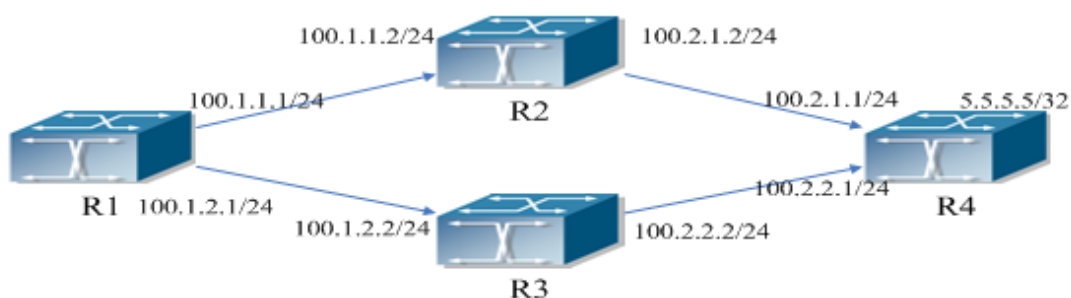


图 4-19 ECMP 的应用环境

如图所示，R1 与 R2、R3 连接的接口地址分别为 100.1.1.1/24、100.1.2.1/24。R2、R3 与 R1 连接的接口地址分别为 100.1.1.2/24、100.1.2.2/24。R4 与 R2、R3 连接的接口地址分别为 100.2.1.1/24、100.2.2.1/24。R2、R3 与 R4 连接的接口地址分别为 100.2.1.2/24、100.2.2.2/24，R4 的 loopback 地址为 5.5.5.5/32。

4.8.3.1 静态路由实现 ECMP

```
R1(config)#ip route 5.5.5.5/32 100.1.1.2
```

```
R1(config)#ip route 5.5.5.5/32 100.1.2.2
```

在 R1 上显示路由表如下：

```
R1(config)#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

* - candidate default

```
C      1.1.1.1/32 is directly connected, Loopback1  tag:0
```

```
S      5.5.5.5/32 [1/0] via 100.1.1.2, Vlan100  tag:0
```

```
          [1/0] via 100.1.2.2, Vlan200  tag:0
```

```
C      100.1.1.0/24 is directly connected, Vlan100  tag:0
```

```
C      100.1.2.0/24 is directly connected, Vlan200 tag:0
```

```
C      127.0.0.0/8 is directly connected, Loopback  tag:0
```

Total routes are : 6 item(s)

4.8.3.2 OSPF 实现 ECMP

R1 的配置

```
R1(config)#interface Vlan100
R1(Config-if-Vlan100)# ip address 100.1.1.1 255.255.255.0
R1(config)#interface Vlan200
R1(Config-if-Vlan200)# ip address 100.1.2.1 255.255.255.0
R1(config)#interface loopback 1
R1(Config-if-loopback1)# ip address 1.1.1.1 255.255.255.255
R1(config)#router ospf 1
R1(config-router)# ospf router-id 1.1.1.1
R1(config-router)# network 100.1.1.0/24 area 0
R1(config-router)# network 100.1.2.0/24 area 0
```

R2 的配置

```
R2(config)#interface Vlan100
R2(Config-if-Vlan100)# ip address 100.1.1.2 255.255.255.0
R2(config)#interface Vlan200
R2(Config-if-Vlan200)# ip address 100.2.1.2 255.255.255.0
R2(config)#interface loopback 1
R2(Config-if-loopback1)# ip address 2.2.2.2 255.255.255.255
R2(config)#router ospf 1
R2(config-router)# ospf router-id 2.2.2.2
R2(config-router)# network 100.1.1.0/24 area 0
R2(config-router)# network 100.2.1.0/24 area 0
```

R3 的配置

```
R3(config)#interface Vlan100
R3(Config-if-Vlan100)# ip address 100.1.2.2 255.255.255.0
R3(config)#interface Vlan200
R3(Config-if-Vlan200)# ip address 100.2.2.2 255.255.255.0
R3(config)#interface loopback 1
R3(Config-if-loopback1)# ip address 3.3.3.3 255.255.255.255
R3(config)#router ospf 1
R3(config-router)# ospf router-id 3.3.3.3
R3(config-router)# network 100.1.2.0/24 area 0
R3(config-router)# network 100.2.2.0/24 area 0
```

R4 的配置

```
R4(config)#interface Vlan100
R4(Config-if-Vlan100)# ip address 100.2.1.1 255.255.255.0
```

```
R4(config)#interface Vlan200
R4(Config-if-Vlan200)# ip address 100.2.2.1 255.255.255.0
R4(config)#interface loopback 1
R4(Config-if-loopback1)# ip address 5.5.5.5 255.255.255.255
R4(config)#router ospf 1
R4(config-router)# ospf router-id 4.4.4.4
R4(config-router)# network 100.2.1.0/24 area 0
R4(config-router)# network 100.2.2.0/24 area 0
```

在 R1 上显示路由表如下：

```
R1(config)#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

* - candidate default

```
C      1.1.1.1/32 is directly connected, Loopback1    tag:0
O      5.5.5.5/32 [110/3] via 100.1.1.2, Vlan100, 00:00:05    tag:0
          [110/3] via 100.1.2.2, Vlan200, 00:00:05    tag:0
C      100.1.1.0/24 is directly connected, Vlan100    tag:0
C      100.1.2.0/24 is directly connected, Vlan200    tag:0
O      100.2.1.0/24 [110/2] via 100.1.1.2, Vlan100, 00:02:25    tag:0
O      100.2.2.0/24 [110/2] via 100.1.2.2, Vlan200, 00:02:25    tag:0
C      127.0.0.0/8 is directly connected, Loopback    tag:0
```

Total routes are : 8 item(s)

4.9 BFD

4.9.1 BFD 介绍

BFD（Bidirectional Forwarding Detection，双向转发检测）是一套全网统一的检测机制，用于快速检测、监控网络中链路连通状况。为了提升现有网络性能，协议邻居之间必须能快速检测到通信故障，从而更快的建立起备用通道恢复通信。

BFD提供了一个通用的、标准化的、介质无关、协议无关的快速故障检测机制，可以为

各上层协议如路由协议、MPLS 等统一地快速检测两台网络设备间双向转发路径的故障。BFD在两台网络设备上建立会话，用来监测两台网络设备间的双向转发路径，为上层协议服务。BFD本身并没有发现机制，而是靠被服务的上层协议通知其该与谁建立会话，会话建立后如果在检测时间内没有收到对端的BFD控制报文则认为发生故障，通知被服务的上层协议，上层协议进行相应的处理。

4.9.2 BFD 配置任务序列

1. BFD 基本功能配置
2. BFD 功能与 RIP（ng）协议联动配置
3. BFD 功能与静态路由（IPv6）协议联动配置
4. BFD 功能与 VRRP（v3）协议联动配置

1. BFD 基本功能配置

命令	解释
全局配置模式	
bfd mode{active passive} no bfd mode	配置 BFD 会话建立前的工作模式，默认为 active 模式。本命令的 no 操作为恢复 active 模式。
bfd authentication key <1-255> text <WORD> no bfd authentication key <1-255>	配置 BFD 认证使用的 key 和文本方式加密的认证字符串。no 命令为删除配置的 key。
bfd authentication key <1-255> md5 <WORD> no bfd authentication key	配置 BFD 使用的 key 和 md5 方式加密的认证字符串。no 命令为删除配置的 key。
接口配置模式	
bfd interval <value1> min_rx <value2> multiplier <value3> no bfd interval	配置 BFD 控制报文的最小发送时间间隔和会话检测系数。本命令的 no 操作为恢复检测系数为默认值。
bfd min-echo-receive-interval <value> no bfd min-echo-receive-interval	配置 BFD 控制报文的最小接收时间间隔，no 命令操作为恢复 echo 报文的最小接收间隔为默认值。
bfd echo no bfd echo	开启 bfd echo 功能，no 命令为关闭 bfd echo 功能。
bfd echo-source-ip <ipv4-address>	配置的 Echo 报文源地址进行

no bfd echo-source-ip	链路故障的探测，no 命令操作为删除配置的 echo 报文源地址。
bfd echo-source-ipv6 <ipv6-address> no bfd echo-source-ipv6	配置的 Echo 报文源地址进行链路故障的探测，no 命令操作为删除配置的 echo 报文源地址。
bfd authentication key <1-255> no bfd authentication key	接口开启 BFD 认证功能和配置使用的密钥。本命令的 no 操作为关闭 BFD 认证功能。

2. BFD 功能与 RIP（ng）协议联动配置

命令	解释
接口配置模式	
rip bfd enable no rip bfd enable	在指定接口上配置 RIP 协议与 BFD 联动，no 命令操作为禁用已经配置的 RIP 协议与 BFD 进行联动功能。
ipv6 rip bfd enable no ipv6 rip bfd enable	在指定接口上配置 RIPng 协议与 BFD 联动；no 命令操作为禁用已经配置的 RIPng 协议与 BFD 进行联动功能。

3. BFD 功能与静态路由（IPv6）协议联动配置

命令	解释
全局配置模式	
ip route {vrf <name> <ipv4-address> <ipv4-address>} mask <nextHop> bfd no ip route {vrf <name> <ipv4-address> <ipv4-address>} mask <nextHop> bfd	配置静态路由与 BFD 进行联动。no 命令操作为取消配置静态路由与 BFD 进行联动。
ipv6 route {vrf <name> <ipv6-address> <ipv6-address>} prefix <nextHop> bfd no ipv6 route {vrf <name> <ipv6-address> <ipv6-address>} prefix <nextHop> bfd	配置 IPv6 静态路由与 BFD 进行联动。no 命令操作为取消配置 IPv6 静态路由与 BFD 进行联动。

4. BFD 功能与 VRRP（v3）协议联动配置

命令	解释

VRRP (v3) 组配置模式	
bfd enable no bfd enable	启用 VRRP (v3) 协议与 BFD 联动功能, 在该组上启用 BFD 检测功能, no 命令操作为禁用 VRRP (v3) 协议与 BFD 联动功能。

4.9.3 BFD 举例

4.9.3.1 BFD 与静态路由联动举例

案例:

在Switch A 上配置静态路由可以到达14.1.1.0/24网段路由, 在Switch B上配置静态路由可以到达15.1.1.0/24网段路由, 并都使能BFD检测功能; 当Switch A 和Switch B 链路出现故障时BFD 能够快速感知。



配置步骤如下:

配置 switch A:

```
Switch#config
```

```
Switch(config)#interface vlan 12
```

```
Switch(config-if-vlan12)#ip address 12.1.1.1 255.255.255.0
```

```
Switch(config)#interface vlan 15
```

```
Switch(config-if-vlan15)#ip address 15.1.1.1 255.255.255.0
```

```
Switch(config)#ip route 14.1.1.0 255.255.255.0 12.1.1.2 bfd
```

配置 switch B:

```
Switch#config
```

```
Switch(config)#interface vlan 12
```

```
Switch(config-if-vlan12)#ip address 12.1.1.2 255.255.255.0
```

```
Switch(config)#interface vlan 14
```

```
Switch(config-if-vlan15)#ip address 14.1.1.1 255.255.255.0
```

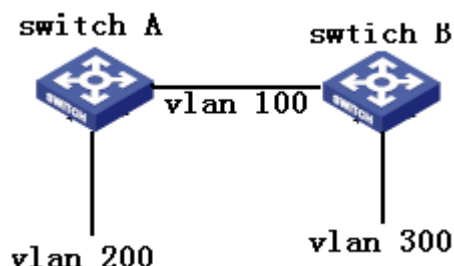
```
Switch(config)#ip route 15.1.1.0 255.255.255.0 12.1.1.1 bfd
```

Switch B 和二层交换机之间链路发生故障时, Switch A能够快速感知Switch B的变化。此时查看静态路由, 路由处于Inactive 状态。

4.9.3.2 BFD 与 RIP 路由联动举例

案例:

Switch A、Switch B连接并运行RIP协议，在两台交换机上开启BFD功能。当Switch A和Switch B链路出现故障时BFD能够快速感知。



配置步骤如下:

配置 switchA:

```
Switch#config
Switch(config)#bfd mode active
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#ip address 10.1.1.1 255.255.255.0
Switch(config)#interface vlan 200
Switch(config-if-vlan200)#ip address 20.1.1.1 255.255.255.0
Switch(config)#router rip
Switch (config-router)#network vlan 100
Switch (config-router)#network vlan 200
Switch(config)#interface vlan 100
Switch(config-if-vlan100) #rip bfd enable
```

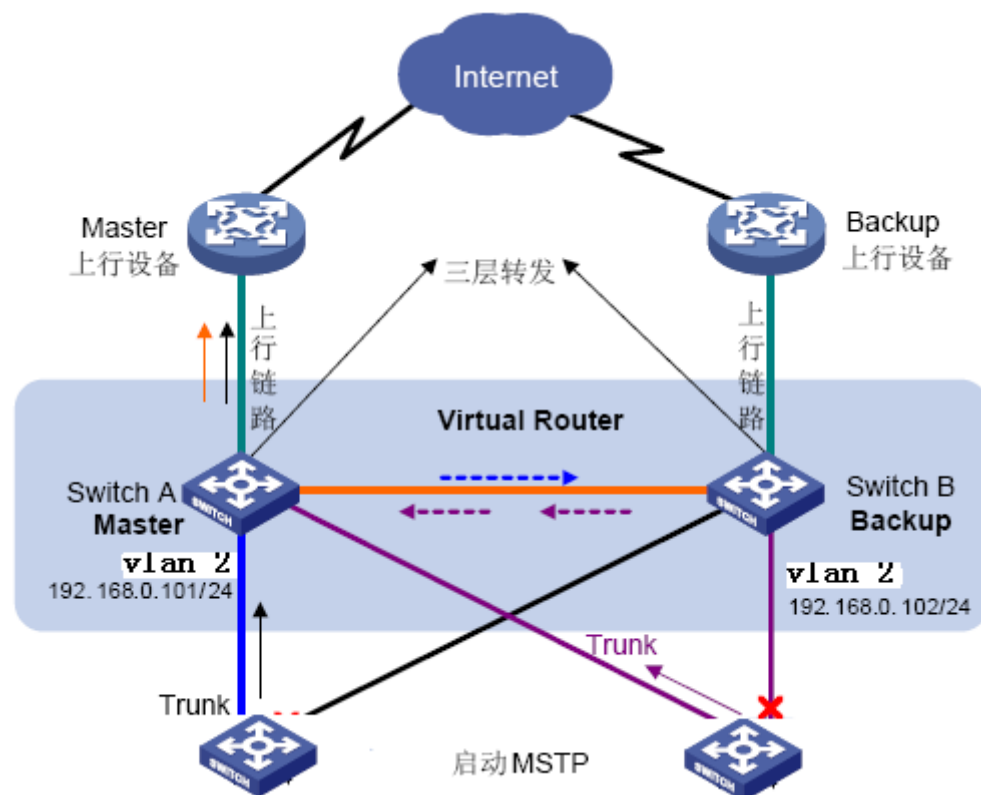
配置 switchB:

```
Switch#config
Switch(config)#bfd mode passive
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#ip address 10.1.1.2 255.255.255.0
Switch(config)#interface vlan 300
Switch(config-if-vlan300)#ip address 30.1.1.1 255.255.255.0
Switch(config)#router rip
Switch (config-router)#network vlan 100
Switch (config-router)#network vlan 300
Switch(config)#interface vlan 100
Switch(config-if-vlan100) #rip bfd enable
```

Switch A 和 Switch B 交换机之间链路发生故障时，BFD 可以快速感知链路故障。BFD 协议通知 RIP 协议删除学习到的路由。

4.9.3.3 BFD 与 VRRP 联动举例

案例：当Master出现故障时，只能依赖于Backup设置的超时时间来判断是否应该抢占，切换时间一般在3秒~4秒之间，无法达到秒级以下的切换速度。VRRP协议必须依赖一种可靠的链路检测机制来探测Master 的当前状态。Backup上的探测协议快速检测Master的运行状态，当Master状态出现故障时，Backup能够立即抢占成为Master，切换时间在100ms 以内。



配置修改：

配置步骤如下：

配置三层交换机Switch A

```
Switch#config
```

```
Switch(config)#bfd mode active
```

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

```
Switch(config-if-vlan2)#ip address 192.16.0.101 255.255.255.0
```

```
Switch(config)#router vrrp 1
```

```
Switch(config-router)#virtual-ip 192.168.0.10
```

```
Switch(config-router)#interface vlan 1
```

```
Switch(config-router)#enable
```

```
Switch(config-router)#bfd enable
```

配置三层交换机Switch B

```
Switch#config
```

```
Switch(config)#bfd mode passive
```

```
Switch(config)#interface vlan 2
Switch(config-ip-vlan2)#ip address 192.16.0.102 255.255.255.0
Switch(config)#router vrrp 1
Switch(config-router)#virtual-ip 192.168.0.10
Switch(config-router)#interface vlan 1
Switch(config-router)#enable
Switch(config-router)#bfd enable
```

4.9.4 BFD 排错帮助

当配置 BFD 功能出现问题时，请检查是否是如下原因：

- ☞ 路由协议的邻居是否已经成功建立，如果路由协议的邻居没有建立成功此时BFD无法进行检测
- ☞ 与静态路由联动时配置的source-ip是否正确，如果两端配置的IP不能互通BFD无法进行检测
- ☞ 与VRRP协议联动时，VRRP组是否已经成功建立，如果VRRP组没有建立成功此时BFD无法进行检测

4.10 BGP GR

4.10.1 GR 简介

随着网络的发展，对于网络的可靠性也有了越来越高的要求，高可靠性技术（HA, High Availability）随之提出，即如何在路由器控制层面出现故障时仍然保证数据包能够正常转发，不影响网络的流量业务。

通常情况下，路由器故障后，其路由协议层面的邻居会检测到它们之间的邻居关系Down掉，然后过段时间再次Up，这个过程被称之为邻居关系震荡。这种邻居关系的震荡将最终导致路由震荡的出现，使得重启路由器在一段时间内出现路由黑洞或者导致邻居将数据业务从重启路由器处旁路，从而导致网络的可靠性大大降低。

为了实现交换机的高可靠性，需要上层的路由协议支持GR(Graceful Restart,优雅重启)的功能。使用路由协议的GR功能，可以在路由器的控制平面出现故障时，保证数据平面正常的包处理与转发。

GR功能的运用还能够减少路由震荡，减少控制平面资源消耗，提高网络稳定性。本文档所述的主要内容为本BGP路由协议的优雅重启（GR），即使BGP协议的重启过程尽可能的不影响转发过程，使数据包能够以正确的路径转发出去。

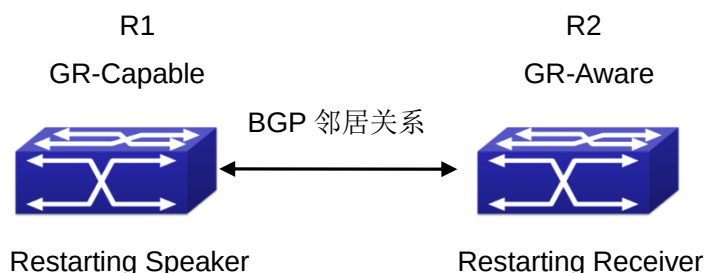


图 4-20 GR 的应用环境

BGP 路由协议的 GR 需要 GR-Capable 路由器与 GR-Aware 路由器协作完成，重启的路由器可以称为 **Restarting Speaker**（也可以叫做 **GR-Restarter**），而它的邻居都可以称为 **Receiving Speaker**（也可以叫做 **GR-Helper**）。Restarting Speaker 为 GR-Capable 路由器，Receiving Speaker 为 GR-Aware 路由器，这样才能协作完成 BGP 优雅重启的动作。假设路由器 R1、R2 建立 BGP 邻居关系，如图 12-1 所示，GR 的过程可以描述为：

对于 Restarting Speaker（GR-Restarter）来说：

- 1、在初始建立 BGP 邻居时，R1、R2 通过 OPEN 消息协商 GR 能力。
- 2、R1 重启发生，路由在接口板被保留，继续指导转发。
- 3、R1 同邻居 R2 重新建立 TCP 连接，在 BGP OPEN 消息中将 Restart state 置为 1，表明该路由器刚刚发生了重启。同时，通知邻居 restart time 的值（应该小于 OPEN 消息中的 Holdtime 时间）。另外，还需要告诉邻居，重启路由器支持哪一类地址族的路由的 GR。
- 4、R1 同邻居 R2 重新成功建立 BGP 邻居关系后，接收邻居发来的路由更新信息并处理，启动延期优选定时器。
- 5、R1 推迟本地的 BGP 路由计算过程，直到收到所有 GR-Aware 的 BGP 邻居的路由更新结束标志 End-of-RIB 或者等到本地的延期优选定时器超时。
- 6、执行路由计算，向各邻居发送路由更新；发送路由更新完毕后，向邻居发送 End-of-RIB 标志。

对于 Receiving Speaker（GR-Helper）来说：

- 1、R2 与 R1 初始建立 BGP 邻居关系的过程中，与重启路由器协商 GR 能力，记录路由器 R1 为支持 GR-Capable 的路由器。
- 2、R1 发生重启后，R2 可能感知到和 R1 之间的 TCP 断掉，也可能在两者重新建立 TCP 连接前检测不到。如果没有检测到，执行 4；否则，检测到执行 3。
- 3、将重启路由器 R1 发过来的路由保持住，并且打上 stale 老化标志。启动 Restart Timer。
- 4、中断旧的 TCP 连接，继续处理新的 TCP 连接，将重启路由器 R1 发过来的路由保持住，并且打上 stale 老化标志。启动 Restart Timer。
- 5、与重启路由器建立新的邻居关系，删除 Restart Timer，启动 Stale Path Timer。
- 6、如果在建立新的邻居关系前，Restart Timer 超时，或者收到新连接的 OPEN 消息中的 Restart flag 不为 1，或者 OPEN 消息中不含有对应的 AFI/SAFI 地址族支持信息，

则清除被保持的重启路由器的路由。

7、向重启路由器发送路由信息更新，更新完毕，发送End-Of-RIB标志。

8、如果Stale Path Timer超时，清除被保持的重启路由器的路由。

4.10.2 GR 配置任务序列

1. 配置是否支持 GR 能力
2. 配置特定邻居是否支持 GR 能力
3. 配置重启超时时间
4. 配置邻居的重启超时时间
5. 配置 BGP GR 保留废弃（STALE）路由超时时间
6. 配置 BGP GR 功能的优选延期超时时间

1. 配置是否支持 GR 能力

命令	解释
BGP 路由配置模式	
bgp graceful-restart no bgp graceful-restart	使能 BGP 支持 GR。

2. 配置特定邻居是否支持 GR 能力

命令	解释
BGP 协议单播地址族模式、VRF 地址族模式	
neighbor (A.B.C.D X:X::X:X WORD) capability graceful-restart no neighbor (A.B.C.D X:X::X:X WORD) capability graceful-restart	为邻居设置标记，在发送 OPEN 消息的时候携带 GR 的能力参数。

3. 配置重启超时时间

命令	解释
BGP 路由配置模式	

bgp graceful-restart restart-time <1-3600> no bgp graceful-restart restart-time <1-3600>	如果同时在 BGP 路由模式下和 BGP 邻居模式下配置该命令，以邻居模式下配置的命令为准。配置 BGP GR 的重启超时时间（Receiving Speaker 为发生重启的邻居启动一个重启超时定时器，使用 restart-time 为超时时间）。重启超时时间指定 Receiving Speaker 从发现邻居重启到收到邻居的 OPEN 消息最长的等待时间，如果超过这个时间 Receiving Speaker 没有收到邻居的 OPEN 消息，Receiving Speaker 可以删除为邻居保留的 SATLE 路由。

4. 配置邻居的重启超时时间

命令	解释
BGP 协议单播地址族模式、VRF 地址族模式	
neighbor (A.B.C.D X:X::X:X WORD) restart-time <1-3600> no neighbor (A.B.C.D X:X::X:X WORD) restart-time <1-3600>	配置邻居的重启超时时间，本命令 no 操作使该定时器恢复默认时间。

5. 配置 BGP GR 保留废弃（STALE）路由超时时间

命令	解释
BGP 路由配置模式	
bgp graceful-restart stale-path-time <1-3600> no bgp graceful-restart stale-path-time <1-3600>	BGP GR 功能的 stalepath-time 采用默认值 360s。stalepath-time 的默认值要比重启超时时间和优选延期超时时间要长的多，因为 Receiving Speaker 在从收到重启邻居的 OPEN 消息到收到该邻居的 EOR 这个时间段内不仅要完成向重启邻居发送初始路由更新的动作，还等待接收重启邻居的初始路由更新，直到接收完毕。

6. 配置 BGP GR 功能的优选延期超时时间

命令	解释
BGP 路由配置模式	

<pre> bgp selection-deferral-time <1-3600> no bgp selection-deferral-time <1-3600> </pre>	指定了 Restarting Speaker 从收到邻居的 OPEN 消息到收到所有邻居的 EOR 后开始路由优选计算的最长的等待时间，如果超过这个时间 Restarting Speaker 还没有收到所有邻居的 EOR，Restarting Speaker 可以开始路由的优选计算。

4.10.3 GR 典型案例

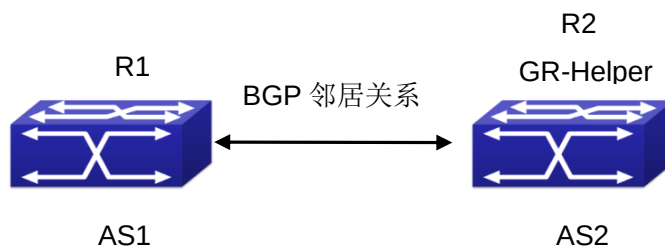


图 4-21 GR 的应用环境

如图 12-2 所示，R1 与 R2 建立 bgp 邻居关系。R1 与 R2 断开邻居关系，R2 上的 BGP 协议会进入 helper 模式，保留 R1 传递给 R2 的路由表项，同时启动重启超时定时器，在定时器时间内收到 R1 的 open 消息或者定时器超时，在 R2 上被标记为 stale 的路由被删除。当 R1 重新与 R2 建立邻居关系后，R1 启动优选定时器，等待 R2 发送 EOR 消息或者定时器超时，R1 优选路由。R2 在收到 R1 的 open 消息后，启动废弃路由定时器，在接受 R1 发送的 EOR 或定时器超时，R2 删除定时器并删除 stale 路由。

R1 配置 int vlan 12, ip address 12.1.1.1

R2 配置 int vlan 12, ip address 12.1.1.2

R1 配置如下

```
R1#config
```

```
R1(config)#vlan 12
```

```
R1(config-vlan12)#int vlan 12
```

```
R1(config-if-vlan12)#ip address 12.1.1.1 255.255.255.0
```

```
R1(config-if-vlan12)#exit
```

```
R1(config)#router bgp 1
```

```
R1(config-router)#neighbor 12.1.1.2 remote-as 2
```

```
R1(config-router)#neighbor 12.1.1.2 capability graceful-restart
```

```
R1(config-router)#bgp selection-deferral-time 120
```

```
R1(config-router)#bgp graceful-restart restart-time 60
```

```
R1(config-router)#bgp graceful-restart stale-path-time 180
```

```
R1(config-router)#exit
```

R2 配置如下

```
R2#config
```

```
R2(config)#vlan 12
```

```
R2(config-vlan12)#int vlan 12
```

```
R2(config-if-vlan12)#ip address 12.1.1.2 255.255.255.0
```

```
R2(config-if-vlan12)#exit
```

```
R2(config)#router bgp 2
```

```
R2(config-router)#neighbor 12.1.1.1 remote-as 1
```

```
R2(config-router)#neighbor 12.1.1.1 capability graceful-restart
```

```
R2(config-router)#bgp selection-deferral-time 120
```

```
R2(config-router)#bgp graceful-restart restart-time 60
```

```
R2(config-router)#bgp graceful-restart stale-path-time 180
```

```
R2(config-router)#exit
```

4.11 OSPF GR

4.11.1 OSPF GR 介绍

OSPF Graceful-Restart（简称 OSPF GR），即 OSPF 平滑重启，主要实现的功能是在路由协议重启或三层交换机发生主备切换时维持原有的数据转发正常，保证关键业务流量不中断，属于高可靠性（HA，High Availability）技术的一种。

目前，高端的三层交换机普遍采用了控制和转发分离的设计。负责路由协议计算的控制模块一般位于主控板上，而数据的转发则位于线卡，这样在主控板卡重启时，才有可能不影响线卡上的数据转发。因此，支持 GR 功能的设备，一般都是机架式设备，具有双主控板结构。

标准的 OSPF 协议（RFC2328）本身不支持 GR 技术，因此在由于各种原因出现主备切换或协议重启时，会造成网络流量中断和路由振荡。例如，在下图所示的拓扑环境中，S1 发生了主备倒换，S2 与 S1 之间的邻接关系将会丢失，这时 S2 会向 S3 和 S4 发送 Router-LSA，该 LSA 中将不包含 S1 与 S2 之间的链路。S3 和 S4 收到该 LSA 后会重新进行路由计算，计算出的转发路径不会包含 S1 与 S2 之间的链路。当 S1 完成主备倒换后，会重新与 S2 建立邻接关系，同步数据库，并导致 S2、S3、S4 需要重新进行路由计算。可见，当 S1 发生主备倒换会造成路由抖动，这对于一些性能要求较高的网络是不可接受的。

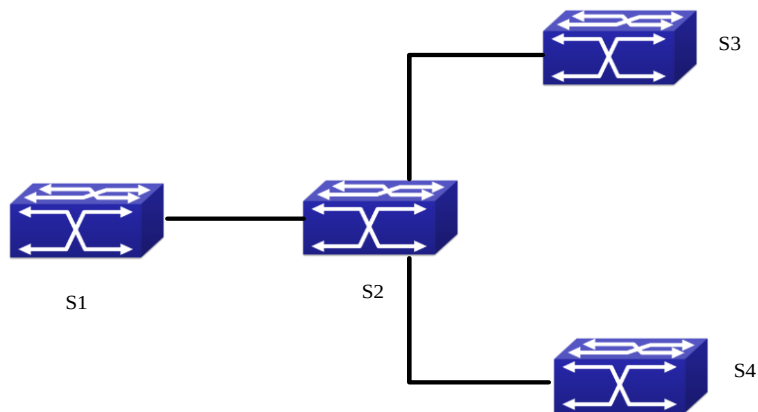


图 4-22 典型应用场景

RFC3623 中所描述 OSPF GR 就是针对以上情况提出的。它的基本思想是：如果在三层交换机主备倒换期间网络拓扑一直保持稳定，并且在这期间发生主备倒换的三层交换机可以保持它的转发表不变，那么该交换机的邻居可以维持与之的邻接关系不变，使得该交换机依然在转发路径上。因此如果 S1 与 S2 支持并启用了 GR 功能，那么在 S1 发生主备切换期间，S1 线卡会保持流量转发，S2 也将保持宣告与 S1 的邻居关系不变，同时 S3 及 S4 不会发现网络拓扑的变化，也不需要重新计算路由，这样就保证了 S1 主备切换期间的流量转发，避免路由振荡。

按照三层交换机在 GR 过程中的不同作用可以将其分为 GR restarter 和 GR helper。GR restarter 是发生主备切换或进行协议重启的三层交换机，而 GR helper 是协助 GR restarter 进行 GR 的三层交换机。上例中 S1 是 GR restarter，S2 是 GR helper。

- 基于 OSPF GR 以上特征，OSPF GR 主要有以下优点：
- 提高网络可靠性
 - 减少路由振荡对网络的影响
 - 减少对业务流量的影响，避免切换期间的丢包

4.11.2 OSPF GR 基本配置

- OSPF GR 配置任务序列如下：
1. 使能 OSPF 进程的 GR 功能
 2. 配置 OSPF GR restarter 的 grace-period（可选）
 3. 配置 OSPF GR helper 的策略（可选）

1. 使能 OSPF 进程的 GR 功能

命令	解释
OSPF 协议配置模式	

capability restart graceful no capability restart	开启特定 OSPF 进程的 GR 功能(默认开启)。

2. 配置 OSPF GR restarter 的 grace-period (可选)

命令	解释
全局配置模式	
ospf graceful-restart grace-period <integer> no ospf restart grace-period	配置 GR restarter (进行主备切换或协议重启的交换机) 的 grace period。本命令的 no 操作为恢复 restarter 的 grace period 为缺省值。

3. 配置 OSPF GR helper 的策略 (可选)

命令	解释
全局配置模式	
ospf graceful-restart helper max-grace-period <integer> no ospf graceful-restart helper	GR helper 策略之一, 配置 helper 支持的最大 grace period。本命令的 no 操作为删除所有配置的 helper 策略。
ospf graceful-restart helper never no ospf graceful-restart helper	GR helper 策略之一, 配置交换机不能作为 OSPF GR helper 角色。本命令的 no 操作为删除所有配置的 helper 策略。

4.11.3 OSPF GR 举例

案例:

四台交换机 S1-S4 (都是双主控结构, 支持 OSPF GR 功能), 均使能 OSPF, 实现以下功能:

1. 上游交换机 S1 在主备切换时保持流量转发, 下游 S2-S4 不会出现路由振荡, 网络流量不断;
2. S1 需要在 120S 内完成主备切换及协议重启, 否则 S2 将退出 GR, 重新计算路由;
3. S1 不充当 OSPF GR Helper 角色 (即当 S2 发生主备切换或 OSPF 协议重启时, S1 不会协助其进行 GR, 而是重新计算路由)

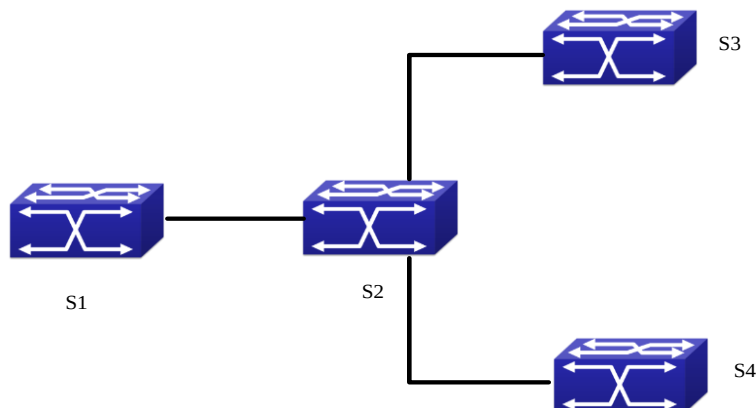


图 4-23 典型应用

配置步骤：交换机缺省已经使能 OSPF GR 功能，因此只需要配置 OSPF GR 的参数及 helper 策略即可。（下面只列出了与 OSPF GR 相关的配置，省略拓扑中 OSPF 的具体配置）

S1

```
S1(config)#ospf graceful-restart grace-period 120
```

```
S1(config)# ospf graceful-restart helper never
```

S2

```
S2(config)# ospf graceful-restart helper max-grace-period 120
```

4.11.4 OSPF GR 排错帮助

当在使用 OSPF GR 过程中遇到问题，请检查是否是如下原因：

- ☞ GR restarter 交换机是否支持 OSPF GR 功能且是双主控结构，请确保没有关闭特定 OSPF 进程的 GR 功能
- ☞ 在 OSPF GR 过程中，是否出现网络拓扑变化，如果出现变化交换机有可能退出 GR，进行正常的 OSPF 重启
- ☞ 请确保 GR restarter 的所有邻居均支持 GR 功能
- ☞ 请尽量不要在交换机进行 GR 期间修改 OSPF 的相关配置

第5章 IPv4 组播协议

5.1 组播

本章对IPv4组播协议的配置进行介绍。

5.1.1 组播简介

当信息（包括数据、语音和视频）传送的目的地是网络中的少数用户时，可以采用多种传送方式。可以采用单播（Unicast）的方式，即为每个用户单独建立一条数据传送通路；或者采用广播（Broadcast）的方式，把信息传送给网络中的所有用户，不管他们是否需要，都会接收到广播来的信息。例如，在一个网络上有200个用户需要接收相同的信息时，传统的解决方案是用单播方式把这一信息分别发送200次，以便确保需要数据的用户能够得到所需的数据；或者采用广播的方式，在整个网络范围内传送数据，需要这些数据的用户可直接在网络上获取。这两种方式都浪费了大量宝贵的带宽资源，而且广播方式也不利于信息的安全和保密。

IP组播技术的出现及时解决了这个问题。组播源仅发送一次信息，组播路由协议为组播数据包建立树型路由，被传递的信息在尽可能远的分叉路口才开始复制和分发，因此，信息能够被准确高效地传送到每个需要它的用户。

需要注意的是，组播源不一定需要加入组播组，它向某些组播组发送数据，自己不一定是该组的接收者。可以同时有多个源向一个组播组发送报文。网络中可能有不支持组播的路由器，组播路由器可以使用隧道方式将组播包封装在单播IP包中传送给相邻的组播路由器，相邻的组播路由器再将单播IP头剥掉，然后继续进行组播传输。从而避免对网络的结构进行较大的改动。组播的优势主要在于：

- 1) 提高效率：降低网络流量，减轻服务器和CPU负荷；
- 2) 优化性能：减少冗余流量；
- 3) 分布式应用：使多点应用成为可能。

5.1.2 组播地址

组播报文的目的地址使用D类IP地址，范围是从224.0.0.0到239.255.255.255。D类地址不能出现在IP报文的源IP地址字段。单播数据传输过程中，一个数据包传输的路径是从源地址路由到目的地址，利用“逐跳”（hop-by-hop）的原理在IP网络中传输。然而在IP组播环境中，数据包的目的地址不是一个，而是一组，形成组地址。所有的信息接收者都加入到一个组内，并且一旦加入之后，流向组地址的数据立即开始向接收者传输，组中的所有成员都

能接收到数据包。组播组中的成员是动态的，主机可以在任何时刻加入和离开组播组。

组播组可以是永久的也可以是临时的。组播组地址中，有一部分由官方分配的，称为永久组播组。永久组播组保持不变的是它的IP地址，组中的成员构成可以发生变化。永久组播组中成员的数量都可以是任意的，甚至可以为零。那些没有保留下来供永久组播组使用的IP组播地址，可以被临时组播组利用。

224.0.0.0~224.0.0.255为预留的组播地址（永久组地址），地址224.0.0.0保留不做分配，其它地址供路由协议使用； 224.0.1.0~238.255.255.255为用户可用的组播地址（临时组地址），全网范围内有效；239.0.0.0~239.255.255.255为本地管理组播地址，仅在特定的本地范围内有效。常用的预留组播地址列表如下：

- 224.0.0.0 基准地址（保留）
- 224.0.0.1 所有主机的地址
- 224.0.0.2 所有组播路由器的地址
- 224.0.0.3 不分配
- 224.0.0.4 DVMRP 路由器
- 224.0.0.5 OSPF 路由器
- 224.0.0.6 OSPF DR
- 224.0.0.7 ST 路由器
- 224.0.0.8 ST 主机
- 224.0.0.9 RIP-2 路由器
- 224.0.0.10 IGRP 路由器
- 224.0.0.11 活动代理
- 224.0.0.12 DHCP 服务器/中继代理
- 224.0.0.13 所有PIM 路由器
- 224.0.0.14 RSVP 封装
- 224.0.0.15 所有CBT 路由器
- 224.0.0.16 指定SBM
- 224.0.0.17 所有SBMS
- 224.0.0.18 VRRP
- 224.0.0.22 IGMP

以太网传输单播IP报文的时候，目的MAC地址使用的是接收者的MAC地址。但是在传输组播报文时，传输目的不再是一个具体的接收者，而是一个成员不确定的组，所以使用的是组播MAC地址。组播MAC地址是和组播IP地址对应的。IANA（Internet Assigned Number Authority）规定，组播MAC地址的高25bit为0x01005e，MAC 地址的低23bit为组播IP地址的低23bit。

由于IP组播地址的后28位中只有23位被映射到MAC地址，这样就会有32个IP组播地址映射到同一MAC地址上。

5.1.3 IP 组播报文转发

在组播模型中，源主机向IP数据包目的地址字段内的组播组地址所表示的主机组传送信息。和单播模型不同的是，组播模型必须将组播数据包转发到多个外部接口上以便能传送到所有接收站点，因此组播转发过程比单播转发过程更加复杂。

为了保证组播信息包都是通过最短路径到达路由器，组播必须依靠单播路由表或者单独提供给组播使用的单播路由表（如DVMRP路由），对组播信息包的接收接口进行一定的检查，这种检查机制就是大部分组播路由协议进行组播转发的基础——RPF（Reverse Path Forwarding，逆向路径转发）检查。组播路由器利用到达的组播数据包的源地址来查询单播路由表或者独立的组播路由表，以确定此数据包到达的入接口处于接收站点至源地址的最短路径上。如果使用的是有源树，这个源地址就是发送组播数据包的源主机的地址；如果使用的是共享树，该源地址就是共享树的根的地址。当组播数据包到达路由器时，如果RPF 检查通过，数据包则按照组播转发项进行转发，否则，数据包被丢弃。

5.1.4 IP 组播应用

IP组播技术有效地解决了单点发送多点接收的问题，实现了IP网络中点到多点的高效数据传输，能够大量节约网络带宽、降低网络负载。利用网络的组播特性可以方便地提供一些新的增值业务。在线直播、网络电视、远程教育、远程医疗、网络电台、实时视/音频会议等互联网的信息服务领域可以提供如下应用：

- 1) 多媒体、流媒体的应用；
- 2) 数据仓库、金融应用（股票）等；
- 3) 任何“点到多点”的数据发布应用。

在 IP 网络中多媒体业务日渐增多的情况下，组播有着巨大的市场潜力，组播业务也将逐渐得到推广和普及。

5.2 PIM-DM

5.2.1 PIM-DM 介绍

PIM-DM（Protocol Independent Multicast, Dense Mode，协议独立组播—密集模式）属于密集模式的组播路由协议，适用于小型网络，在这种网络环境下，组播组的成员相对比较密集。

PIM-DM 的工作过程可以概括为：邻居发现、扩散—剪枝过程、嫁接阶段。

1. 邻居发现

PIM-DM 路由器刚开始启动时，需要使用Hello报文来发现邻居。运行PIM-DM的各网络节点之间使用Hello报文保持联系。PIM-DM Hello报文是周期性发送的。

2. 扩散—剪枝过程（Flooding&Prune）

PIM-DM 假设网络上的所有主机都准备接收组播数据。当某组播源S开始向组播组G发

送数据时,在路由器接收到组播报文后,首先根据单播路由表进行RPF检查,如果检查通过,路由器创建一个(S,G)表项,然后将组播报文向网络上所有下游PIM-DM节点转发(Flooding)。如果没有通过RPF检查,即组播报文是从错误的接口输入,则将该报文丢弃。经过这个过程,在PIM-DM组播域内,每个节点都会创建一个(S,G)表项。如果下游节点没有组播组成员,则向上游节点发剪枝(Prune)消息,通知上游节点不用再转发该组播组数据。上游节点收到剪枝消息后,就将相应的接口从其组播转发表项(S,G)对应的输出接口列表中删除,这就建立了一个以源S为根的SPT(Shortest Path Tree, SPT)树。剪枝过程最先由叶子路由器发起。

3. RPF检查

PIM-DM采用RPF检查,利用现存单播路由表构建一棵从数据源始发的组播转发树。当一个组播包到达时,路由器首先判断到达路径的正确性。如果到达接口是单播路由指示的通往组播源的接口,就认为这个组播包是从正确路径而来;否则,将组播包作为冗余报文丢弃。作为路径判断依据的单播路由信息可以来源于任何一种单播路由协议,如RIP、OSPF等发现的路由信息,而不依赖于特定的单播路由协议。

4. Assert机制

如果处于一个LAN网段上的两台组播路由器A和B,各自有到组播源S的接收途径,它们在接收到组播源S发出的组播数据报文以后,都会向LAN上转发该组播报文,这时,下游节点组播路由器C就会收到两份相同的组播报文。路由器检测到这种情况后,需要通过Assert机制来选定一个唯一的转发者。通过发送Assert报文,选出一条最优的转发路径,如果两条或两条以上路径的优先级和开销相同,则选择IP地址大的节点作为该(S,G)项的上游邻居,由它负责该(S,G)组播报文的转发。

5. 嫁接(Graft)

当被剪枝的下游节点需要恢复到转发状态时,该节点使用嫁接报文通知上游节点恢复组播数据转发。

6. PIM-DM的状态刷新

为了减少PIM-DM周期性的扩散-剪枝,PIM-DM协议最新版本提供了一个选项(option),它将从组播源的第一跳路由器开始,周期性的向下面广播(S,G)状态刷新消息以完成状态刷新。当下游PIM路由器收到(S,G)状态刷新消息,它就将剪枝计时器复位,终止剪枝计时器超时,从而即保证了网络的时效性,避免了周期性的扩散-剪枝过程。

5.2.2 PIM-DM 配置任务序列

3.启动 PIM-DM (必须)

4.配置静态组播表项 (可选)

5.配置 PIM-DM 辅助参数 (可选)

- a) 配置 PIM-DM hello 报文间隔时间
- b) 配置 state-refresh 报文间隔时间
- c) 配置边缘接口
- d) 配置管理边界

6. 关闭 PIM-DM 协议

1. 启动 PIM-DM 协议

在三层交换机上运行 PIM-DM 路由协议的基本配置很简单，需全局配置模式下打开 PIM 组播开关，然后在相应接口下打开 PIM-DM 开关即可。

命令	解释
全局配置模式	
ip pim multicast-routing no ip pim multicast-routing	使各个接口上的 PIM-DM 协议进入使能状态（但真正在接口上开始 PIM-DM 协议，还需下面的命令）。

然后在接口上打开 PIM-SM 开关

命令	解释
接口配置模式	
ip pim dense-mode	启动本接口 PIM-DM 协议。（必须）

2. 配置静态组播表项

命令	解释
全局配置模式	
ip mroute <A.B.C.D> <A.B.C.D> <ifname> <.ifname> no ip mroute <A.B.C.D> <A.B.C.D> [<.ifname> <.ifname>]	配置静态组播表项，其 no 命令删除静态组播表项或其出接口。

3. 配置 PIM-DM 辅助参数

a) 配置 PIM-DM hello 报文间隔时间

命令	解释
接口配置模式	
ip pim hello-interval <interval> no ip pim hello-interval	配置接口 PIM-DM hello 报文间隔时间；本命令的 no 操作恢复为缺省值。

b) 配置 state-refresh 报文间隔时间

命令	解释
全局配置模式	
ip pim state-refresh origination-interval no ip pim state-refresh origination-interval	全局模式下配置 PIM-DM state-refresh 报文间隔时间；本命令的 no 操作恢复为缺省值。

c) 配置边缘接口

命令	解释
接口配置模式	
ip pim bsr-border no ip pim bsr-border	配置接口为 PIM-DM 边缘接口，边缘接口上，BSR 相关消息不向该接口发送也不从该接口接收，连接的网络被认为都是该接口的直连网络。本命令的 no 操作取消该配置。

d) 配置管理边界

命令	解释
接口配置模式	
ip pim scope-border <1-99> <acl_name> no ip pim scope-border	配置 PIM-DM 管理边界和使用的 ACL。组播数据不向 SCOPE-BORDER 扩散，默认的情形下，认为 239.0.0.0/8 的范围为管理组范围，若配置了 ACL，则 ACL permit 的范围为管理组范围。本命令的 no 操作取消该配置。

4. 关闭 PIM-DM 协议

命令	解释
接口配置模式	
no ip pim dense-mode	在接口上关闭 PIM-DM 协议。
全局配置模式	
no ip pim multicast-routing	全局关闭 PIM-DM 协议。

5.2.3 PIM-DM 典型案例

如下图，将 switchA，switchB 的以太网接口加入到相应的 VLAN 中，并在各 VLAN 接口上启动 PIM-DM 协议。如果 VLAN 内同时配置了 IGMP SNOOPING，则组播流量能精确到端口，而不在整个 VLAN 内泛洪。

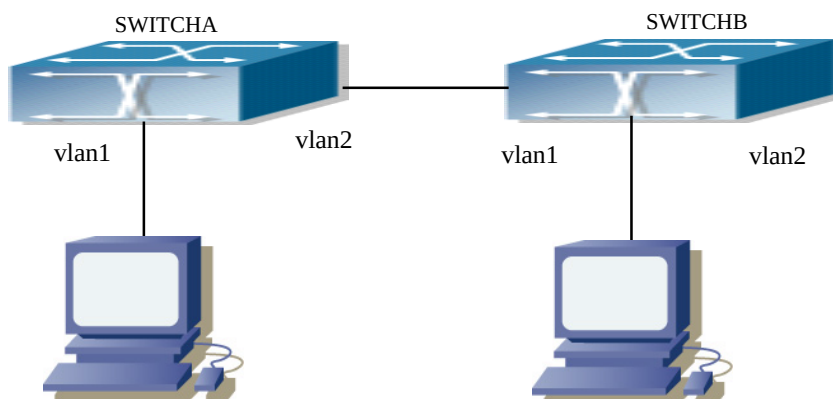


图 5-1 PIM-DM 典型环境

switchA 和switchB配置步骤如下：

(1) 配置SwitchA:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ip address 10.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)# ip pim dense-mode
Switch(Config-if-Vlan1)#exit
Switch (config)#interface vlan2
Switch(Config-if-Vlan2)# ip address 12.1.1.1 255.255.255.0
Switch(Config-if-Vlan2)# ip pim dense-mode
```

(2) 配置SwitchB:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ip address 12.1.1.2 255.255.255.0
Switch(Config-if-Vlan1)# ip pim dense-mode
Switch(Config-if-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-if-Vlan2)# ip address 20.1.1.1 255.255.255.0
Switch(Config-if-Vlan2)# ip pim dense-mode
```

同时要注意配置好单播路由协议，确保网络中各设备之间能够在网络层互通，并且能够借助单播路由协议实现动态路由更新。

5.2.4 PIM-DM 排错帮助

在配置、使用 PIM-DM 协议时，可能会由于物理连接、配置错误等原因导致 PIM-DM 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证接口和链路协议是 UP（使用 show interface 命令）
- ☞ 保证全局配置模式下打开 PIM 协议(使用 ip pim multicast-routing)
- ☞ 在接口上启动 PIM-DM 协议（使用 ip pim dense-mode 命令）
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性

如果使用检查仍无法解决 PIM-DM 的问题，那么请使用 debug pim 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.3 PIM-SM

5.3.1 PIM-SM 介绍

PIM-SM (Protocol Independent Multicast, Sparse Mode) 即与协议无关组播-稀疏模式, 属于稀疏模式的组播路由协议, 主要用于组成员分布相对分散、范围较广、大规模的网络。与密集模式的扩散—剪枝不同, PIM-SM 协议假定所有的主机都不需要接收组播数据包, 只有主机明确指定需要时, PIM-SM 路由器才向它转发组播数据包。

PIM-SM 通过设置汇聚点 RP (Rendezvous Point) 和自举路由器 BSR (Bootstrap Router), 向所有 PIM-SM 路由器通告组播信息, 并利用路由器的加入/剪枝信息, 建立起基于 RP 的共享树 RPT (RP-rooted shared tree)。从而减少数据报文和控制报文占用的网络带宽, 降低路由器的处理开销。组播数据沿着共享树流到该组播组成员所在的网段, 当数据流量达到一定程度, 组播数据流可以切换到基于源的最短路径树 SPT, 以减少网络延迟。PIM-SM 不依赖于特定的单播路由协议, 而是使用现存的单播路由表进行 RPF 检查。

1. PIM-SM 工作原理

PIM-SM 的工作过程主要有: 邻居发现、RP 共享树 (RPT) 的生成、组播源注册、SPT 切换等。其中, 邻居发现机制与 PIM-DM 相同, 这里不再介绍。

(1) RP 共享树 (RPT) 的生成

当主机加入一个组播组 G 时, 与该主机直接相连的叶子路由器通过 IGMP 报文了解到有组播组 G 的接收者, 就为组播组 G 计算出对应的汇聚点 RP, 然后向朝着 RP 方向的上一级节点发送加入组播组的消息 (join 消息)。从叶子路由器到 RP 之间途经的每个路由器都会在转发表中生成 (*, G) 表项, 表示由任意源发出的, 发送至组播组 G 的, 都适用于该表项。当 RP 收到发往组播组 G 的报文后, 报文就会沿着已经建立好的路径到达叶子路由器, 进而到达主机。这样就生成了以 RP 为根的 RPT。

(2) 组播源注册

当组播源 S 向组播组 G 发送了一个组播报文时, 与其直接相连的 PIM-SM 组播路由器收到该报文以后, 就负责将该组播报文封装成注册报文, 单播给对应的 RP。如果一个网段上有多个 PIM-SM 组播路由器, 这时候将由指定路由器 DR (Designated Router) 负责发送该组播报文。

(3) SPT 切换

当组播路由器发现从 RP 发来的目的地址为 G 的组播报文的速率超过了阈值时, 组播路由器就向朝着源 S 的上一级节点发送加入消息, 导致 RPT 向 SPT 的切换。

2. PIM-SM 配置前准备工作

(1) 配置候选 RP

在 PIM-SM 网络中, 可以存在多个 RP (候选 RP), 每个候选 RP (Candidate-RP, C-RP) 负责转发目的地址在一定范围内的组播报文。配置多个候选 RP 可以实现 RP 负载分担。候选 RP 之间没有主次之分, 所有的组播路由器收到 BSR 通告的候选 RP 消息后, 根据相同的算法

计算出与某一组播组对应的RP。

注意，一个RP可以为多个组播组服务，也可以为所有组播组服务。每个组播组在任意时刻，只能唯一地对应一个RP，不能同时对应多个RP。

(2) 配置BSR

BSR 是PIM-SM 网络里的管理核心，它负责收集候选RP发来的信息，并把它们广播出去。

一个网络内部只能有一个BSR，但可以配置多个候选BSR(Candidate-BSR, C-BSR)。这样，一旦某个BSR发生故障后，能够切换到另外一个。C-BSR通过自动选举产生BSR。

5.3.2 PIM-SM 配置任务序列

- 1、启动 PIM-SM（必须）
- 2、配置静态组播表项（可选）
- 3、配置 PIM-SM 辅助参数（可选）
 - (1) 配置 PIM-SM 接口参数
 - 1) 配置 PIM-SM hello 报文间隔时间
 - 2) 配置 PIM-SM hello 报文 holdtime 时间
 - 3) 配置 PIM-SM 邻居访问列表
 - 4) 配置接口为 PIM-SM 边缘接口
 - 5) 配置接口为 PIM-SM 管理边界
 - (2) 配置 PIM-SM 全局参数
 - 1) 配置交换机作为候选 BSR
 - 2) 配置交换机作为候选 RP
 - 3) 配置静态 RP
 - 4) 配置内核组播路由缓存时间
- 4、关闭 PIM-SM 协议

1. 启动 PIM-SM 协议

在三层交换机上运行 PIM-SM 路由协议的基本配置很简单，需全局配置模式下打开 PIM 组播开关，然后在相应接口下打开 PIM-SM 开关即可。

命令	解释
全局配置模式	
ip pim multicast-routing	使各个接口上的 PIM-SM 协议进入使能状态(但真正在接口上开始 PIM-SM 协议，还需下面的命令)。(必须)

然后在接口上打开 PIM-SM 开关

命令	解释
接口配置模式	
ip pim sparse-mode	启动本接口 PIM-SM 协议。(必须)

2. 配置静态组播表项

命令	解释
全局配置模式	
ip mroute <A.B.C.D> <A.B.C.D> <ifname> <.ifname> no ip mroute <A.B.C.D> <A.B.C.D> [<ifname> <.ifname>]	配置静态组播表项，其 no 命令删除静态组播表项或其出接口。

3. 配置 PIM-SM 辅助参数

(1) 配置 PIM-SM 接口参数

1) 配置 PIM-SM hello 报文间隔时间

命令	解释
接口配置模式	
ip pim hello-interval < interval> no ip pim hello-interval	配置接口 PIM-SM hello 报文间隔时间；本命令的 no 操作恢复为缺省值。

2) 配置 PIM-SM hello 报文 holdtime 时间

命令	解释
接口配置模式	
ip pim hello-holdtime <value> no ip pim hello-holdtime	配置接口 PIM-SM hello 报文中的 holdtime 域的值；本命令的 no 操作恢复为缺省值。

3) 配置 PIM-SM 邻居访问列表

命令	解释
接口配置模式	
ip pim neighbor-filter {<access-list-number> } no ip pim neighbor-filter {<access-list-number> }	配置邻居访问列表。如果被列表过滤，如果已经同此邻居建立连接，则此连接马上被切断，如果没有建立连接，则这个连接不能建立。

4) 配置边缘接口

命令	解释
接口配置模式	
ip pim bsr-border no ip pim bsr-border	配置接口为 PIM-DM 边缘接口，边缘接口上，BSR 相关消息不向该接口发送也不从该接口接收，连接的网络被认为都是该接口的直连网络。本命令的 no 操作取消该配置。

5) 配置管理边界

命令	解释
接口配置模式	
ip pim scope-border <1-99> <acl_name> no ip pim scope-border	配置 PIM-DM 管理边界和使用的 ACL。组播数据不向 SCOPE-BORDER 扩散, 默认的情形下, 认为 239.0.0.0/8 的范围为管理组范围, 若配置了 ACL, 则 ACL permit 的范围为管理组范围。acl_name 应当是 IPV4 标准 ACL 名。本命令的 no 操作取消该配置。

(2) 配置 PIM-SM 全局参数

1) 配置交换机作为候选 BSR

命令	解释
全局配置模式	
ip pim bsr-candidate {vlan <vlan-id> <ifname>} [<mask-length>] [<priority>] no ip pim bsr-candidate	该命令为全局候选 BSR 配置命令, 用于配置 PIM-SM 候选 BSR 的信息, 以用于同其它候选 BSR 竞争 BSR 路由器; 本命令的 no 操作为取消候选 BSR 的配置。

2) 配置交换机作为候选 RP

命令	解释
全局配置模式	
ip pim rp-candidate {vlan<vlan-id> loopback<index> <ifname>} [<A.B.C.D>] [<priority>] no ip pim rp-candidate	该命令为全局候选 RP 配置命令, 用于配置 PIM-SM 候选 RP 的信息, 以用于同其它候选 RP 竞争 RP 路由器; 本命令的 no 操作为取消候选 RP 的配置。

3) 配置静态 RP

命令	解释
全局配置模式	
ip pim rp-address <A.B.C.D> [<A.B.C.D/M>] no ip pim rp-address <A.B.C.D> {<all/> <A.B.C.D/M>}	该命令为全局或某个组播地址范围的组播组配置静态 RP, 本命令的 no 操作为取消静态 RP 的配置。

4) 配置内核组播路由缓存时间

命令	解释
全局配置模式	

ip multicast unresolved-cache aging-time <value> no ip multicast unresolved-cache aging-time	配置内核组播路由缓存时间；本命令的 no 操作恢复为缺省值。

4. 关闭 PIM-SM 协议

命令	解释
接口配置模式	
no ip pim sparse-mode no ip pim multicast-routing (全局配置模式)	关闭 PIM-SM 协议。

5.3.3 PIM-SM 典型案例

如下图所示，将switchA，switchB，switchC，switchD的以太网接口加入到相应VLAN中，并在各VLAN接口上启动PIM-SM协议。如果VLAN内同时配置了IGMP SNOOPING，则组播流量能精确到端口，而不在整个VLAN内泛洪。

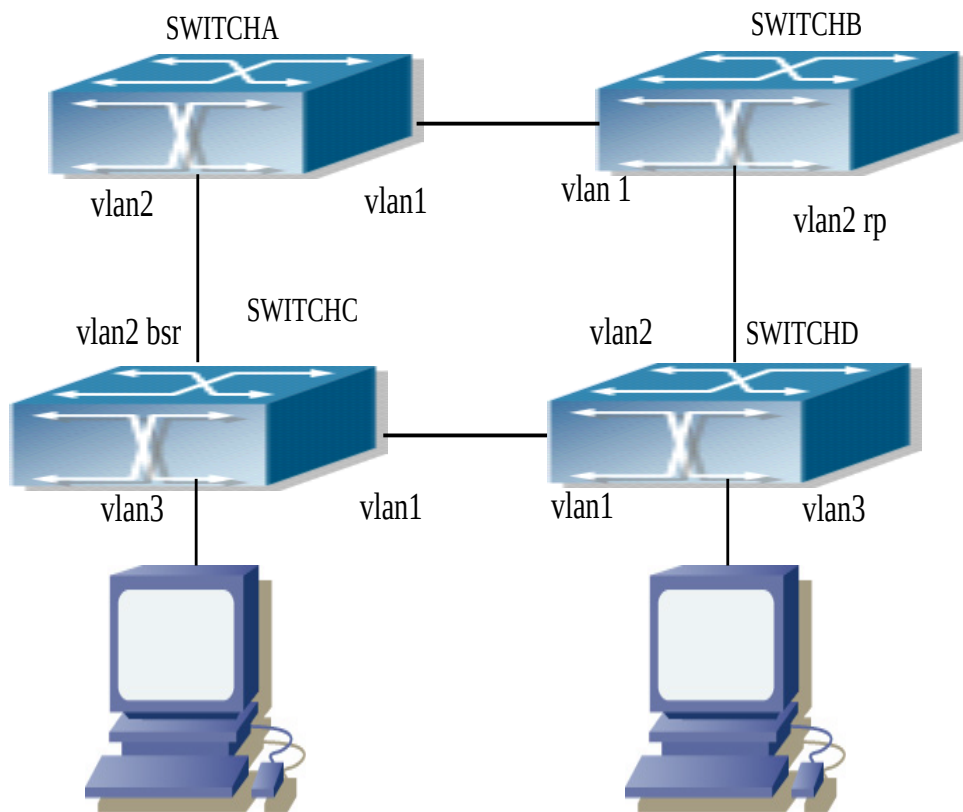


图 5-2 PIM-SM 典型环境

switchA 和 switchB, switchC, switchD 配置步骤如下:

(1) 配置 SwitchA:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ip address 12.1.1.1 255.255.255.0
Switch(Config-If-Vlan1)# ip pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ip address 13.1.1.1 255.255.255.0
Switch(Config-If-Vlan2)# ip pim sparse-mode
```

(2) 配置 SwitchB:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ip address 12.1.1.2 255.255.255.0
Switch(Config-If-Vlan1)# ip pim sparse-mode
```

```
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ip address 24.1.1.2 255.255.255.0
Switch(Config-If-Vlan2)# ip pim sparse-mode
Switch(Config-If-Vlan2)# exit
Switch (config)# ip pim rp-candidate vlan2
```

(3) 配置 SwitchC:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ip address 34.1.1.3 255.255.255.0
Switch(Config-If-Vlan1)# ip pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ip address 13.1.1.3 255.255.255.0
Switch(Config-If-Vlan2)# ip pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch (config)#interface vlan 3
Switch(Config-If-Vlan3)# ip address 30.1.1.1 255.255.255.0
Switch(Config-If-Vlan3)# ip pim sparse-mode
Switch(Config-If-Vlan3)# exit
Switch (config)# ip pim bsr-candidate vlan2 30 10
```

(4) 配置 SwitchD:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ip address 34.1.1.4 255.255.255.0
Switch(Config-If-Vlan1)# ip pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ip address 24.1.1.4 255.255.255.0
Switch(Config-If-Vlan2)# ip pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch (config)#interface vlan 3
Switch(Config-If-Vlan3)# ip address 40.1.1.1 255.255.255.0
Switch(Config-If-Vlan3)# ip pim sparse-mode
```

同时要注意配置好单播路由协议，确保网络中各设备之间能够在网络层互通，并且能够借助单播路由协议实现动态路由更新。

5.3.4 PIM-SM 排错帮助

在配置、使用 PIM-SM 协议时，可能会由于物理连接、配置错误等原因导致 PIM-SM 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误；
- ☞ 保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 保证全局配置模式下 PIM 协议打开（使用 `ip pim multicast-routing`）；
- ☞ 保证接口上配置 PIM-SM（使用 `ip pim sparse-mode`）；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性；
- ☞ PIM-SM 协议需要有 rp 和 bsr 的支持，所以首先使用 `show ip pim bsr-router`，看是否有 bsr 信息，如果不存在，则需要查看是否有通向 bsr 的单播路由；
- ☞ 使用 `show ip pim rp-hash` 命令查看 rp 信息是否正确，如果没有 rp 信息，也要检查单播路由。

如果使用检查仍无法解决 PIM-SM 的问题，那么请使用 `debug pim/ debug pim bsr` 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.4 MSDP

5.4.1 MSDP 介绍

MSDP（Multicast Source Discovery Protocol，组播源发现协议）用来发现其它 PIM-SM 域内的组播源信息。配置了 MSDP 对等体的 RP 将其域内的活动组播源信息通过 SA 消息通告给它的所有 MSDP 对等体，这样，一个 PIM-SM 域内的组播源信息就会被传递到另一个 PIM-SM 域。在 MSDP 里使用的是域间信源树而不是共享树，而且要求域内组播路由协议必须是 PIM-SM。

➤ PIM-SM 协议中 RP 工作原理

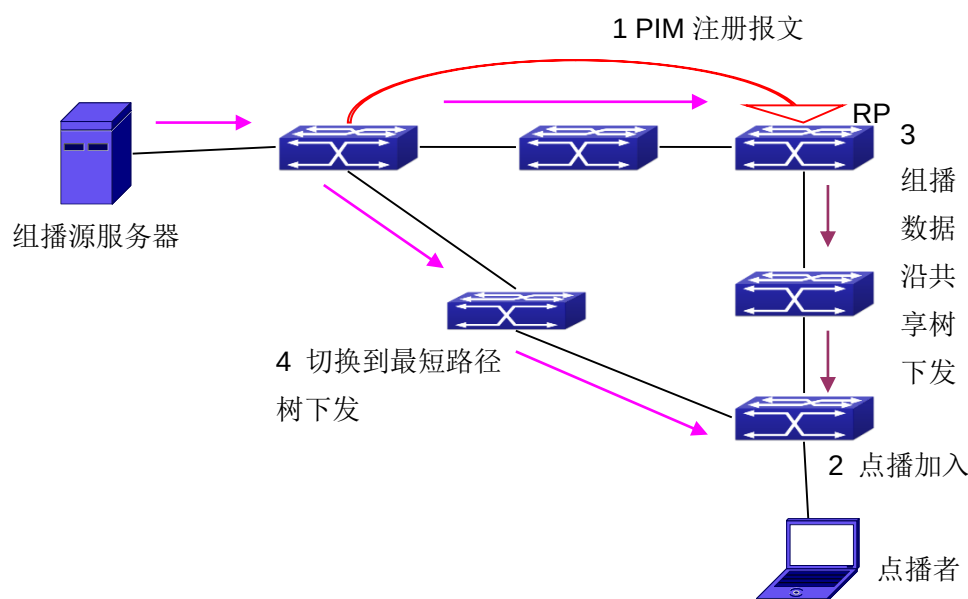


图 5-3 RP 工作原理

5.4.2 MSDP 配置任务简介

1. 配置MSDP基本功能
 - 1) 使能MSDP(必选)
 - 2) 创建MSDP对等体(必选)
 - 3) 配置Connect-Source接口
 - 4) 配置静态RPF对等体
 - 5) 配置Originator-RP
 - 6) 配置TTL阈值
2. 配置MSDP对等体参数
 - 1) 配置Connect-Source接口
 - 2) 配置MSDP对等体的描述信息
 - 3) 配置AS号
 - 4) 配置MSDP全连接组
 - 5) 配置缓存最大数目
3. 配置报文收发参数
 - 1) 配置SA报文创建过滤策略
 - 2) 配置SA报文接收和转发过滤规则
 - 3) 配置SA请求报文
 - 4) 配置SA-Request报文过滤策略
4. 配置SA-cache参数
 - 1) 配置SA报文缓存
 - 2) 配置缓存表项的存活时间
 - 3) 配置缓存最大数目

5.4.3 配置 MSDP 基本功能

本节的所有命令都是在PIM-SM域内的RP上配置的，这些RP将成为MSDP对等体的一端。

5.4.3.1 配置准备

在配置MSDP基本功能之前，需完成以下任务：

- 配置任一单播路由协议，实现域内和域间网络层互通
- 配置PIM-SM基本功能，实现域内组播

在配置MSDP基本功能之前，需准备以下数据：

- MSDP对等体的IP地址

- 过滤策略列表

注意：MSDP不能和Any-cast RP一起使用，但是可以配置基于MSDP协议的Any-cast RP。

5.4.3.2 使能 MSDP

在配置 MSDP 各功能之前，必须先使能 MSDP。

1.启动 MSDP 功能

2.配置 MSDP

1.启动 MSDP 功能

命令	解释
全局配置模式	
router msdp no router msdp	启动 MSDP 功能。no 命令关闭全局 MSDP 功能。

2.配置 MSDP 辅助参数

命令	解释
MSDP 配置模式	
connect-source <interface-type> <interface-number> no connect-source	配置 MSDP Peer 的 Connect-Source 接口。 no 命令取消配置的 Connect-Source 接口。
default-rpf-peer <peer-address> [rp-policy <acl-list-number> <word>] no default-rpf-peer	配置静态 RPF Peer。 no 命令取消配置的 RPF Peer。
originating-rp <interface-type> <interface-number> no originating-rp	配置 Originator-RP。 no 命令取消配置的 Originator-RP。
ttl-threshold <tth> no ttl-threshold	配置 TTL 阈值。 no 命令取消配置的 TTL 阈值。

5.4.4 配置 MSDP 对等体

5.4.4.1 创建 MSDP Peer

命令	解释
MSDP 配置模式	
peer <peer-address> no peer <peer-address>	创建 MSDP Peer。 no 命令删除 MSDP Peer。

5.4.4.2 配置 MSDP 辅助参数

命令	解释
MSDP Peer 配置模式	
connect-source <interface-type> <interface-number> no connect-source	配置 MSDP Peer 的 Connect-Source 接口。 no 命令取消配置的 Connect-Source 接口。
description <text> no description	配置 MSDP 对等体的描述信息。 no 命令删除 MSDP 对等体的描述信息。
remote-as <as-num> no remote-as <as-num>	配置 MSDP Peer 的 AS 号。 no 命令取消配置 MSDP Peer 的 AS 号。
mesh-group <name> no mesh-group <name>	配置 MSDP Peer 加入指定全连接组。 no 命令取消配置 MSDP Peer 的全连接组成员身份。

5.4.5 配置报文收发

命令	解释
MSDP 配置模式	
redistribute [list <acl-list-number acl-name>] no redistribute	配置 SA 报文创建过滤策略。 no 命令取消配置 SA 报文创建过滤策略。
MSDP 配置模式或 MSDP Peer 配置模式	
sa-filter (in out) [list <acl-number acl-name> rp-list <rp-acl-number rp-acl-name>] no sa-filter (in out) [[list <acl-number acl-name> rp-list <rp-acl-number rp-acl-name>]	配置 SA 报文接收和转发过滤规则。 no 命令取消配置的 SA 报文接收和转发过滤规则。
MSDP Peer 配置模式	
sa-request no sa-request	配置发送 SA 请求报文。 no 命令取消发送 SA 请求报文。
MSDP 配置模式	
sa-request-filter [list <access-list-number access-list-name>] no sa-request-filter [list <access-list-number 	配置 SA-Request 报文过滤策略。 取消配置 SA-Request 报文过滤策略。

<code>access-list-name>]</code>	

5.4.6 配置 SA-cache 参数

命令	解释
MSDP 配置模式	
cache-sa-state	使能 SA 报文缓存。
no cache-sa-state	关闭 SA 报文缓存。
cache-sa-holdtime <150-3600>	配置缓存表项的存活时间。
no cache-sa-holdtime	恢复缓存表项的默认的存活时间。
MSDP 配置模式或 MSDP Peer 配置模式	
cache-sa-maximum <sa-limit>	配置缓存最大数目。
no cache-sa-maximum	恢复默认缓存最大数目。

5.4.7 MSDP 举例

案例 1: MSDP 功能。

组播配置:

1. 假设有组播服务器上提供组播地址 224.1.1.1 的节目，组播源开始发送数据包；
2. 连接组播源的指定路由器 DR（Designated Router）将组播源发出的数据封装在 Register 报文里，发给本域内的 RP（RP1）；
3. RP 将报文解封装，沿域内的共享树向下转发给域内的所有成员，域内成员可以选择是否切换到源树上；
4. 同时，域内 RP 作为源端 RP，将生成一个 SA（Source Active，活动源）消息，发送给 MSDP 对等体（RP2）。
5. 如果 MSDP 对等体所在的域里有组成员（RP3），RP3 将 SA 消息中封装的组播数据沿共享树下发到组成员的同时，向组播源发送加入消息；即 RP 创建(S, G)表项，向源端 DR 逐跳发送(S, G)加入消息(Join Message)，从而跨越各 PIM-SM 域直接加入以该组播源为根的 SPT。

如果某个 MSDP 对等体所在的域（RP2）里没有组成员，则 RP2 不会创建(S, G)表项，也不加入以该组播源为根的 SPT。

6. 当逆向转发路径建立起来之后，组播源发出的数据将直接发送到 RP3 上，RP 向共享树转发数据。此时，连接组成员的最后一跳路由器可以自行选择是否切换到 SPT 上。

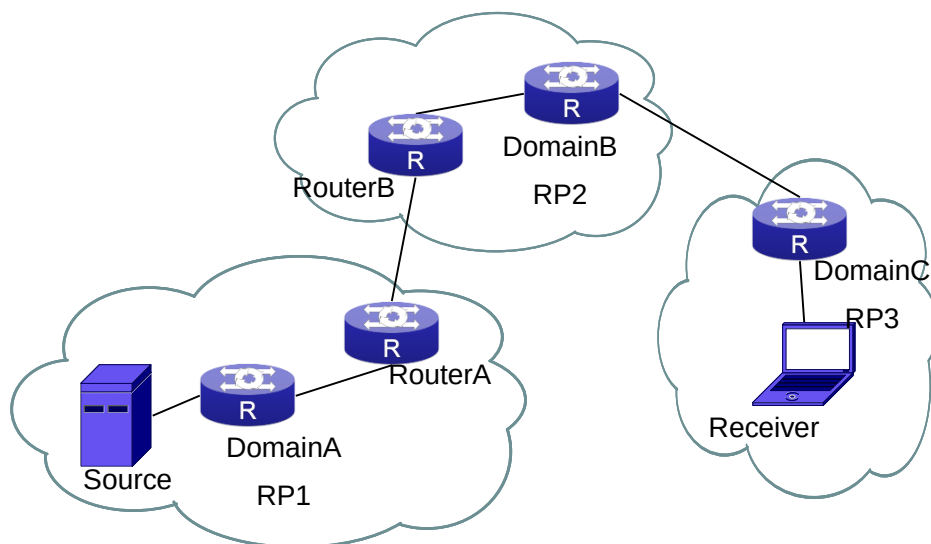


图 5-4 MSDP 对等体位置

配置步骤如下：

配置准备：

在每个路由器上启动单播路由协议和PIM协议，保证域内域间路由相通，域内组播正常。

假设 Domain A 中的组播服务器 S 上提供组播地址 224.1.1.1 的节目，Domain C 有主机 R 点播该节目，没有使能 MSDP 之前点播不到该节目，完成以下配置之后，R 能收看 S 的节目。

Domain A 的 RP1:

```
Switch#config
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 10.1.1.2
```

Domain A 的 Router A :

```
Switch#config
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 20.1.1.2 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(config)#router msdp
```

```
Switch(router-msdp)#peer 10.1.1.1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 20.1.1.1
```

Domain B 的 Router B :

```
Switch#config
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(config)#interface vlan 3
Switch(Config-if-Vlan3)#ip address 30.1.1.1 255.255.255.0
Switch(Config-if-Vlan3)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 20.1.1.2
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 30.1.1.2
```

Domain B 的 RP2:

```
Switch#config
Switch(config)#interface vlan 3
Switch(Config-if-Vlan3)#ip address 30.1.1.2 255.255.255.0
Switch(config)#interface vlan 4
Switch(Config-if-Vlan4)#ip address 40.1.1.2 255.255.255.0
Switch(Config-if-Vlan4)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 30.1.1.1
Switch(config)#router msdp
Switch(router-msdp)#peer 40.1.1.1
```

Domain C 的 RP3:

```
Switch(config)#interface vlan 4
Switch(Config-if-Vlan1)#ip address 40.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 40.1.1.2
```

案例 2：MSDP Mesh-Group 应用。

Mesh-Group 规则可以在全连接组中有效的减少 SA 消息的泛洪。通常把同一个域中全

连接的所有 Peer，配置为一个 Mesh-Group，且所有组成员均使用相同的组名称。

如图，对同域中四个全连接的 Peer 配置了 Mesh-Group 规则后，从 SA 消息的转发情况可看出，SA 消息的泛洪情况得到了很好的解决。

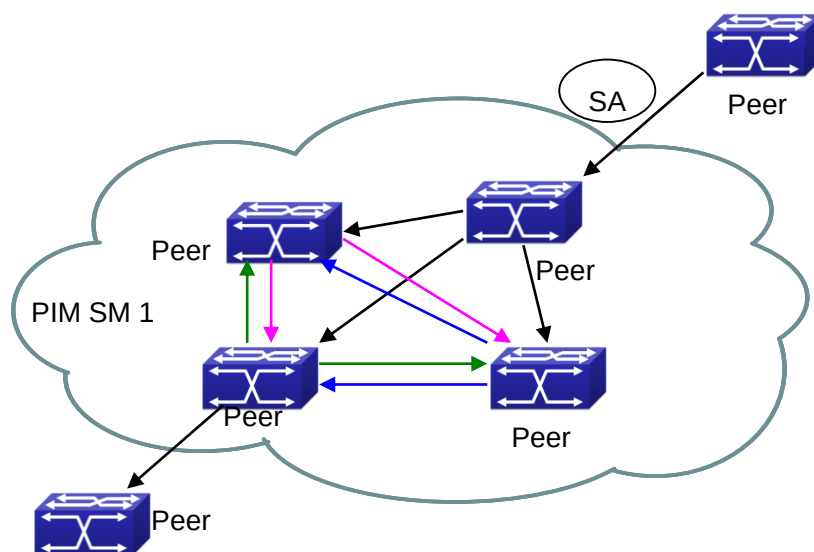


图 5-5 SA 消息泛洪

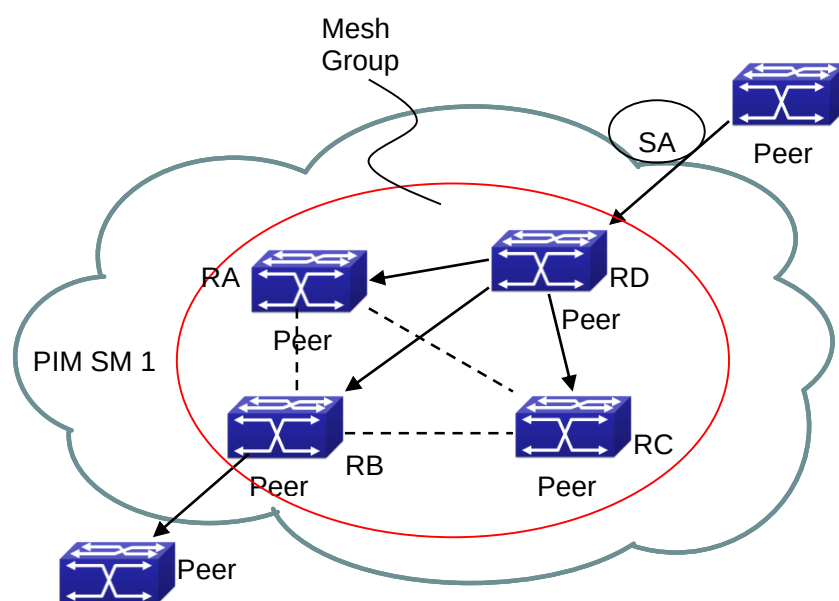


图 5-6 Mesh-Group 下控制 SA 消息泛洪

配置步骤如下：

Router A:

```
Switch#config
```

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(config)#interface vlan 3
Switch(Config-if-Vlan3)#ip address 30.1.1.1 255.255.255.0
Switch(Config-if-Vlan3)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 10.1.1.2
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 20.1.1.4
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 30.1.1.3
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
```

Router B :

```
Switch#config
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 4
Switch(Config-if-Vlan4)#ip address 40.1.1.2 255.255.255.0
Switch(Config-if-Vlan4)#exit
Switch(config)#interface vlan 6
Switch(Config-if-Vlan6)#ip address 60.1.1.2 255.255.255.0
Switch(Config-if-Vlan6)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 10.1.1.1
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 40.1.1.4
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 60.1.1.3
Switch(router-msdp)#mesh-group test-1
```

Router C :

```
Switch#config
Switch(config)#interface vlan 4
Switch(Config-if-Vlan4)#ip address 40.1.1.4 255.255.255.0
Switch(Config-if-Vlan4)#exit
Switch(config)#interface vlan 5
Switch(Config-if-Vlan5)#ip address 50.1.1.4 255.255.255.0
Switch(Config-if-Vlan5)#exit
Switch(config)#interface vlan 6
Switch(Config-if-Vlan6)#ip address 60.1.1.4 255.255.255.0
Switch(Config-if-Vlan6)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 20.1.1.1
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 40.1.1.4
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 60.1.1.2
Switch(router-msdp)#mesh-group test-1
```

Router D:

```
Switch#config
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 20.1.1.4 255.255.255.0
Switch(Config-if-Vlan2)#exit
Switch(config)#interface vlan 4
Switch(Config-if-Vlan1)#ip address 40.1.1.4 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 5
Switch(Config-if-Vlan5)#ip address 50.1.1.4 255.255.255.0
Switch(Config-if-Vlan5)#exit
Switch(config)#router msdp
Switch(router-msdp)#peer 20.1.1.1
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 40.1.1.2
```

```
Switch(router-msdp)#mesh-group test-1
Switch(msdp-peer)#exit
Switch(router-msdp)#peer 50.1.1.3
Switch(router-msdp)#mesh-group test-1
```

5.4.8 MSDP 排错帮助

在配置、使用 MSDP 功能时，可能会由于物理连接、配置错误等原因导致 MSDP 未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证各个 Peer 之间路由可达，即域内和域间网络层互通
- ☞ 保证各个域内的组播协议使用 PIM-SM，PIM-SM 配置正确并且运行正常
- ☞ 保证 MSDP 已经使能，配置的 Peer 使用的接口是 UP 的
- ☞ 使用 **show msdp gloabl** 命令查看 MSDP 配置信息是否正确

如果使用检查仍无法解决 MSDP 的问题，那么请使用 **debug msdp** 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.5 ANYCAST RP

5.5.1 ANYCAST RP 介绍

基于 PIM 协议的 Anycast RP 是一种为了让 RP 失效能够尽快恢复而提供冗余的技术，Anycast-RP 地址通常是配置在 loopback 接口上的地址。

Anycast RP 的核心思想是在整个网络上配置的 RP 地址，存在于多台组播路由器上（通常的情况是在各个提供 ANYCAST RP 的设备上都使用 LOOPBACK 接口，并配置 RP 地址在这个接口上，使用最长的掩码），而单播路由算法将决定了 PIM 路由器总是能找到离自己最近的 RP，因此对于比较大的网络提供了更短更快的找到 RP 的路径。而一旦正在使用的某一个 RP 失效，单播路由算法将保证 PIM 路由器很快找到新的 RP 路径，从而使组播服务迅速恢复。多个 RP 会出现新的问题，即如果组播源和接收者注册到不同的 RP，会导致有的接收者不能接收到组播源数据（注册报文显然只会青睐离自己的最近的 RP）。因此，为了在各 RP 间保持联络，Anycast RP 规定离组播源最近的 RP 在收到注册报文时应将源注册信息转发通知给其他 RP，保证所有 RP 上的加入者都可以找到该组播源。

基于 PIM 协议的 Anycast RP 具体实现方法是：在每个配置了 Anycast RP 的交换机上维护一份 ANYCAST RP 的列表，并使用另一个地址作为他们之间互相识别的标志，在一台 Anycast RP 设备收到注册消息后，以标志自己的地址作为源地址向其它 Anycast RP 设备发送注册消息，以达到让其它设备也知道本源的目的。

5.5.2 ANYCAST RP 配置任务

1. 启动 ANYCAST RP v4 功能

2. 配置 ANYCAST RP v4

1. 启动 ANYCAST RP v4 功能

命令	解释
全局配置模式	
ip pim anycast-rp	启动 ANYCAST RP 功能。（必须）
no ip pim anycast-rp	no 操作关闭全局 ANYCAST RP 功能。

2. 配置 ANYCAST RP v4

(1) 配置候选 RP

命令	解释
全局配置模式	
ip pim rp-candidate {vlan<vlan-id> loopback<index> <ifname>} [<A.B.C.D>] [<priority>] no ip pim rp-candidate	目前 PIM-SM 已允许配置 Loopback 接口作为候选 RP。（必须） 需要注意，ANYCAST RP 协议可以配置 Loopback 接口或普通三层 VLAN 接口作为候选 RP。为使网络中 PIM 路由器找到 RP 位置，应将候选 RP 接口添加到路由中。 no 操作取消本路由器候选 RP 配置。

(2) 配置 self-rp-address (本 RP 通信地址)

命令	解释
全局配置模式	
ip pim anycast-rp self-rp-address A.B.C.D no ip pim anycast-rp self-rp-address	在本路由器（作为 RP）上配置本路由器的 self-rp-address。该地址唯一标志本路由器，用于与其他 RPs 之间的通讯。 self-rp-address 具体作用有两个方面： 1 当本路由器（作为 RP）收到从 DR 单播过来的注册报文，需要将该注册报文向网络中其他 RP 转发，使得网络中所有 RP 获得源（S,G）状态。转发时，本路由器修改注册报文源地址为 self-rp-address。 2 当本路由器（作为 RP）收到从其他 RP 单播转发过来的注册报文，如注册报文目的地

	址为本路由器 self-rp-address, 则建立(S,G) 状态并反方向发送注册中止报文, 注册中止 报文目的地址即注册报文的源地址。 注意: self-rp-address 应为本路由器上某个 三层接口地址, 但在配置时我们允许此接口 当前不存在。self-rp-address 是唯一的。 no 操作取消本路由器(作为 RP)用于与其 他 RP 通信的 self-rp-address。

(3) 配置 other-rp-address (其他 RP 通信地址)

命令	解释
全局配置模式	
<pre>ip pim anycast-rp <anycast-rp-addr> <other-rp-addr> no ip pim anycast-rp <anycast-rp-addr> <other-rp-addr></pre>	在本路由器(作为 RP)上配置 anycast-rp-addr 地址。该单播地址实际上就 是在网络中多个 RP 上配置的 RP 地址, 对 应候选 RP 接口(或者 Loopback 接口)的地 址。 anycast-rp-addr 具体作用有: 1 允许配置多个 anycast-rp-addr 地址, 只有 与当前配置的候选 RP 接口地址相同的 anycast-rp-addr 地址生效。此时对应该 anycast-rp-addr 地址的 other-rp-address 也 才会生效。 2 配置时我们允许 anycast-rp-addr 对应的接 口当前不存在。 在本路由器(作为 RP)上配置与本路由器通 讯的其他 RP 的 other-rp-address。该单播地 址标志其他 RP, 用于与本地路由器之间的通 讯。 other-rp-address 具体作用有两个方面: 1 当本路由器(作为 RP)收到从 DR 单播过 来的注册报文, 需要将该注册报文向网络中 其他 RP 转发, 使得网络中所有 RP 获得源 (S, G) 状态。转发时, 本路由器修改注册 报文目的地址为 other-rp-address。 2 对应一个 anycast-rp-addr 可以配置多个 other-rp-address, 当收到 DR 单播过来的注

	册报文，依次向这些其他 RP 转发。 no 操作取消与本路由器通信的某个 other-rp-address。

5.5.3 ANYCAST RP 典型案例

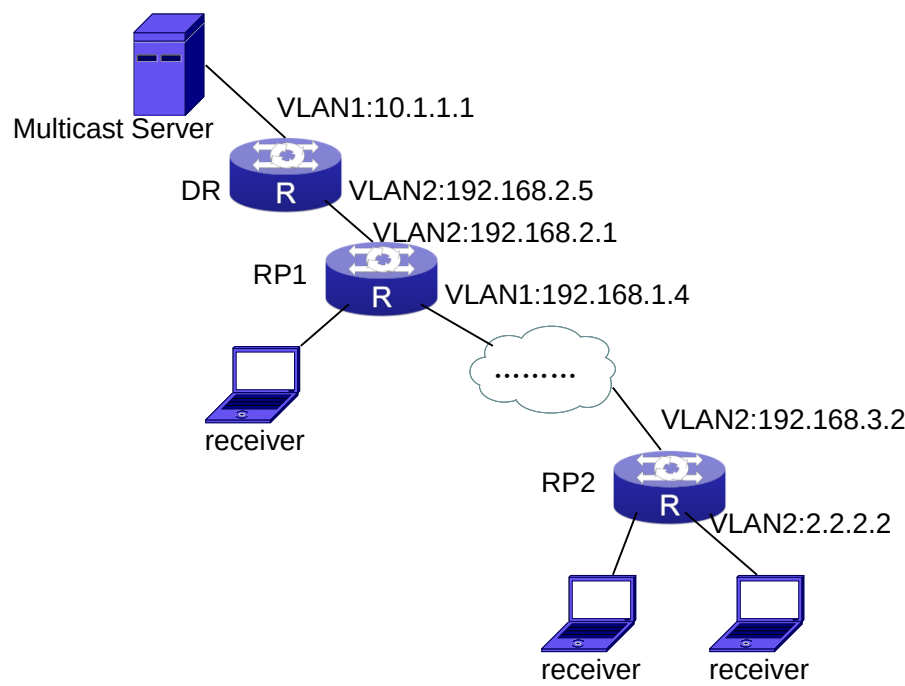


图 5-7 路由器 ANYCAST RP v4 功能

如图所示，整体网络环境为 PIM-SM，网络中提供了两台支持 ANYCAST RP 的路由器 RP1，RP2。组播源服务器发出的组播数据到达 DR 后，DR 依据单播路由算法向距离自己最近的 RP 单播发送组播源注册报文，此处为 RP1。在 RP1 收到来自 DR 的注册报文后，除了根据已加入自己的点播者向共享树下发，RP1 会在收到注册报文后将组播源注册信息转发通知给 RP2, 保证所有 RP2 上的加入者都可以找到该组播源。由于在配置了 ANYCAST RP 的路由器 RP1 上维护了一份 ANYCAST RP 的列表，该列表保存了网络中所有其他 RP 的单播通讯地址，因此 RP1 收到注册消息后，以标志自己的 self-rp-address 地址作为源地址向 RP2 转发注册消息。本图中云层代表处于 RP1 和 RP2 之间运行 PIM-SM 的网络。

配置步骤如下：

RP1 配置：

```
Switch#config
Switch(config)#interface loopback 1
```

```
Switch(Config-if-Loopback1)#ip address 1.1.1.1 255.255.255.255
Switch(Config-if-Loopback1)#exit
Switch(config)#ip pim rp-candidate loopback1
Switch(config)#ip pim bsr-candidate vlan 1
Switch(config)#ip pim multicast-routing
Switch(config)#ip pim anycast-rp
Switch(config)#ip pim anycast-rp self-rp-address 192.168.2.1
Switch(config)#ip pim anycast-rp 1.1.1.1 192.168.3.2
```

RP2 配置:

```
Switch#config
Switch(config)#interface loopback 1
Switch(Config-if-Loopback1)#ip address 1.1.1.1 255.255.255.255
Switch(Config-if-Loopback1)#exit
Switch(config)#ip pim rp-candidate loopback1
Switch(config)#ip pim multicast-routing
Switch(config)#ip pim anycast-rp
Switch(config)#ip pim anycast-rp self-rp-address 192.168.3.2
Switch(config)#ip pim anycast-rp 1.1.1.1 192.168.2.1
```

5.5.4 ANYCAST RP 排错帮助

在配置、使用 ANYCAST RP 功能时，可能会由于物理连接、配置错误等原因导致

ANYCAST RP 未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证 PIM-SM 协议正确运行
- ☞ 保证全局配置模式下 ANYCAST RP 打开
- ☞ 保证全局配置模式下 self-rp-address 配置正确
- ☞ 保证全局配置模式下 other-rp-address 配置正确
- ☞ 保证各接口路由正确添加，包括作为 RP 的 loopback 接口也要正确加入路由
- ☞ 使用 show ip pim anycast rp status 命令查看 ANYCAST RP 配置信息是否正确

如果使用检查仍无法解决 ANYCAST RP 的问题，那么请使用 debug pim anycast-rp 调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.6 PIM-SSM

5.6.1 PIM-SSM 介绍

源指定组播(PIM-SSM)是一种区别于传统组播的新的业务模式，它使用组播组地址和组播源地址来同时标志一个组播会话。SSM保留了传统PIM-SM模式中的主机显示加入组播组的高效性，但是跳过了PIM-SM模式中的共享树和RP规程。SSM直接建立由(S,G)标志的spt树，其中G表示组播组的地址，S表示发向组播组G的特定组播源的地址。SSM的一个（S，G）对也被称为一个频道。SSM特别适用于点到多点的组播服务，如网络体育频道，网络新闻频道等业务。在缺省情况下，SSM组播组地址范围限制在IP地址范围：232.0.0.0-232.255.255.255内，但是可以根据需要对地址进行扩展。

PIM-DM的环境中也支持PIM-SSM。

5.6.2 PIM-SSM 配置任务序列

命令	解释
全局配置模式	
ip multicast ssm {default range <access-list-number >} no ip multicast ssm	该命令配置 pim ssm 组播组地址范围；本命令的 no 操作删除配置的 pim ssm 组播组。

5.6.3 PIM-SSM 典型案例

如下图所示，将switchA，switchB，switchC，switchD的以太网接口加入到相应VLAN中，并在各VLAN接口上启动PIM-SM协议，或者PIM-DM协议，全局启动PIM-SSM。本例以PIM-SM协议为例。

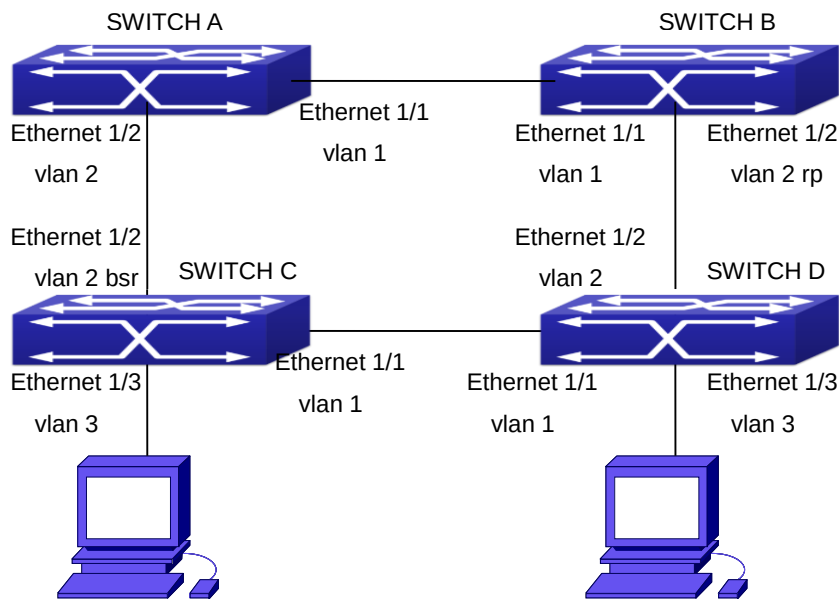


图 5-8 PIM-SSM 典型环境

switchA 和 switchB, switchC, switchD 配置步骤如下:

(1) 配置 switchA:

```
switch(config)#ip pim multicast-routing
switch(config)#interface vlan 1
switch(Config-If-Vlan1)#ip pim sparse-mode
switch(Config-If-Vlan1)#exit
switch(config)#interface vlan 2
switch(Config-If-Vlan2)#ip pim sparse-mode
switch(Config-If-Vlan2)#exit
switch(config)#access-list 1 permit 224.1.1.1 0.0.0.255
switch(config)#ip multicast ssm range 1
```

(2) 配置 switchB:

```
switch(config)#ip pim multicast-routing
switch(config)#interface vlan 1
switch(Config-If-Vlan1)#ip pim sparse-mode
switch(Config-If-Vlan1)#exit
switch(config)#interface vlan 2
switch(Config-If-Vlan2)#ip pim sparse-mode
switch(Config-If-Vlan2)#exit
switch(config)# ip pim rp-candidate vlan2
switch(config)#access-list 1 permit 224.1.1.1 0.0.0.255
switch(config)#ip multicast ssm range 1
```

(3) 配置 switchC:

```
switch(config)#ip pim multicast-routing
switch(config)#interface vlan 1
switch(Config-If-Vlan1)#ip pim sparse-mode
switch(Config-If-Vlan1)#exit
switch(config)#interface vlan 2
switch(Config-If-Vlan2)#ip pim sparse-mode
switch(Config-If-Vlan2)#exit
switch(config)#interface vlan 3
switch(Config-If-Vlan3)#ip pim sparse-mode
switch(Config-If-Vlan3)# exit
switch(config)#ip pim bsr-candidate vlan2 30 10
switch(config)#access-list 1 permit 224.1.1.1 0.0.0.255
switch(config)#ip multicast ssm range 1
```

(4) 配置 switchD:

```
switch(config)#ip pim multicast-routing
```

```
switch(config)#interface vlan 1
switch(Config-If-Vlan1)#ip pim sparse-mode
switch(Config-If-Vlan1)#exit
switch(config)#interface vlan 2
switch(Config-If-Vlan2)#ip pim sparse-mode
switch(Config-If-Vlan2)#exit
switch(config)#interface vlan 3
switch(Config-If-Vlan3)#ip pim sparse-mode
switch(Config-If-Vlan3)#exit
switch(config)#access-list 1 permit 224.1.1.1 0.0.0.255
switch(config)#ip multicast ssm range 1
```

5.6.4 PIM-SSM 排错帮助

在配置、使用 PIM-SSM 协议时，可能会由于物理连接、配置错误等原因导致 PIM-SSM 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误；
- ☞ 保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 保证全局配置模式下 PIM 协议打开（使用 ip pim multicast-routing）；
- ☞ 保证接口上配置 PIM-SM（使用 ip pim sparse-mode）；
- ☞ 保证全局模式下配置 SSM；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性。

如果使用检查仍无法解决问题，那么请使用 debug pim event/debug pim packet 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.7 DVMRP

5.7.1 DVMRP 介绍

DVMRP（Distance Vector Multicast Routing Protocol）协议即“距离向量组播路由协议”。它是一种密集模式的组播路由协议，采用类似 RIP 方式的路由交换给每个源建立了一个转发广播树，然后通过动态的剪枝/嫁接给每个源建立起一个截断广播树，也就是到源的最短路径树。通过反向路径检查(RPF)来决定组播包是否应该被转发到下游。

DVMRP 的一些重要特性是：

1. 用于决定反向路径检查信息的路由交换以距离向量为基础（方式与 RIP 相似）
2. 路由交换更新周期性的发生（缺省为 60 秒）
3. TTL 上限=32 跳（而 RIP 是 16）
4. 路由更新包括掩码，支持 CIDR

相对单播路由来说，组播路由是一种颠倒的路由（也就是，你所感兴趣的是信息包从哪里来而不是到哪里去），因此在DVMRP路由表中的信息是被用于确定是否在正确的接口收到一个输入的组播信息包。否则，为了防止组播循环将放弃该信息包。

把为了确定信息包到达正确的接口所进行的测试称为RPF检查。当有组播数据包到达某个接口，通过查找DVMRP路由表来决定到源网络的反向路径。如果数据包到达的接口是用于向源传送单播信息的接口，则逆向路径检查正确，数据包从所有下游接口转发出去。如果不是，则可能是出现了故障，丢弃该组播包。

因为不是所有的交换机都支持组播，DVMRP支持隧道组播通信，隧道是在被不支持组播路由的交换机隔开的DVMRP交换机之间发送组播数据报的一种方法。它充当两个DVMRP交换机之间的虚拟网络。组播数据包被封装在单播数据包之内，直接发送到下一个支持组播的交换机。DVMRP协议平等对待隧道接口与一般的物理接口。

如果在一个多入口网络上连接了两个或两个以上的交换机，就可能会把一个数据包的多份拷贝发送到该子网上。因此必须指定一个指定的转发者，DVMRP在利用路由交换的机制来达到这一目的，当多入口网络上的两个交换机交换路由信息时，就会互相知道对方到源网络的路由度量，因此到源网络的度量最小的交换机成为该子网上的指定转发者，如果度量一致，则IP地址较低的获胜。

当交换机的某个接口配置了运行DVMRP协议以后，就在该接口上向其他的DVMRP交换机组播探寻（Probe）消息，用于发现邻居且互相探知对方的能力（Capabilities）。如果在邻居超时前一直没有收到该邻居的Probe消息，则认为该邻居丢失。

在DVMRP中，源网络路由选择信息用和RIP相同的基本方式交换。也就是说，路由报告消息定期（缺省为60秒）在DVMRP邻居之间发送。DVMRP路由选择表中的路由信息被用于建立源分布树，也就是决定通过哪一个邻居能够到达发送组播信息的源，到该邻居的接口被称为上游接口。路由报告包含源网络（用掩码）地址和用于路由尺度的跳数项。

为了转发能够正确的完成，每个DVMRP交换机都需要知道哪些下游交换机需要通过它从某个特定的源网络接收组播信息。当接收到某个特定的源发来的包后，DVMRP交换机首先从所有的下游接口，也就是在该接口上有其他的对该特定源表示了依赖性的DVMRP交换机，把该组播包广播出去。当接口上收到了某个下游交换机发送的剪枝（Prune）消息后，就把该交换机剪枝。DVMRP交换机使用毒性反转来通知某个特定源的上游交换机：“我是你的下游”。DVMRP交换机通过把它广播的某个特定源的路由度量加上无限（32）来回应该源上游交换机来完成毒性反转。这意味着度量正确值为1到2*无限（32）-1或1到63，1到31意味着可以到达源网络，32意味着源网络不可达，33到63意味着产生该报告（Report）消息的交换机依赖上游路由器来接收特定源的组播信息。

5.7.2 配置任务序列

- 1、全局启动和关闭 DVMRP（必须）
- 2、配置在接口启动和关闭 DVMRP 协议（必须）
- 3、配置 DVMRP 辅助参数（可选）

配置 DVMRP 接口参数

- 1) 配置 DVMRP 接口上发送 report 报文的延迟和每次发送的报文数
- 2) 配置 DVMRP 接口 metric 值
- 3) 配置 DVMRP 是否与不能 Prune/Graft 的 DVMRP 路由器建立邻居
- 4、配置 DVMRP 隧道

1. 全局启动 DVMRP 协议

在三层交换机上运行 DVMRP 路由协议的基本配置很简单，首先需在全局模式下打开 DVMRP 开关。

命令	解释
全局配置模式	
[no] ip dvmrp multicast-routing	全局启动 DVMRP 协议，本命令的 no 操作全局关闭 DVMRP 协议。（必须）

2. 在接口上启动 DVMRP 协议

在三层交换机上运行 DVMRP 路由协议的基本配置很简单，在全局启动 DVMRP 协议后，需在相应的接口下打开 DVMRP 开关。

命令	解释
接口配置模式	
ip dvmrp enable no ip dvmrp	在接口上启动 DVMRP 协议，本命令的 no 操作在接口上关闭 DVMRP 协议。

3. 配置 DVMRP 辅助参数

(1) 配置 DVMRP 接口参数

- 1) 配置 DVMRP 接口上发送 report 报文的延迟和每次发送的报文数
- 2) 配置 DVMRP 接口 metric 值
- 3) 配置 DVMRP 是否与不能 Prune/Graft 的 DVMRP 路由器建立邻居

命令	解释
接口配置模式	
ip dvmrp output-report-delay <delay_val> [<burst_size>] no ip dvmrp output-report-delay	配置接口上发送 DVMRP report 报文的延时以及每次发送所发送的报文数量，本命令的 no 操作恢复默认值。
ip dvmrp metric <metric_val> no ip dvmrp metric	配置接口 DVMRP report 报文度量值；本命令的 no 操作恢复为缺省值。
ip dvmrp reject-non-pruners no ip dvmrp reject-non-pruners	配置接口上拒绝与 non pruning/grafting 的 DVMRP 路由器建立邻居关系，本命令的 no 操作恢复为可以建立邻居关系。

4. 配置 DVMRP 隧道

命令	解释
接口配置模式	
ip dvmrp tunnel <index> <src-ip> <dst-ip> no ip dvmrp tunnel {<index> <src-ip> <dst-ip>}	本命令配置一条 DVMRP 隧道；本命令的 no 操作删除一条 DVMRP 隧道。

5.7.3 DVMRP 典型案例

如下图，将switchA 和switchB 的以太网接口加入到相应的VLAN中，并在各VLAN接口上启动DVMRP。

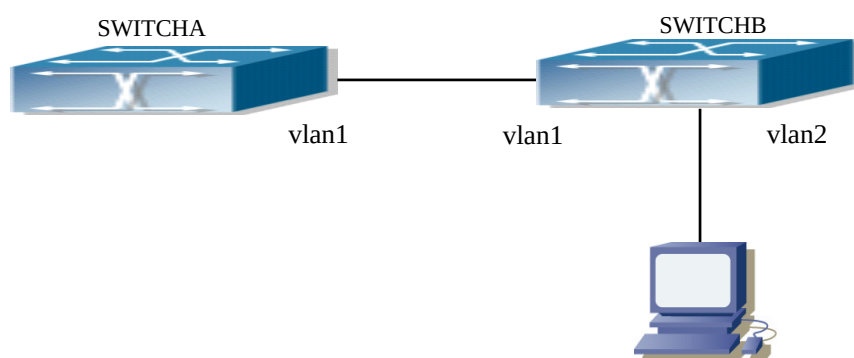


图 5-9 DVMRP 网络拓扑示意图

switchA 和switchB配置步骤如下：

(1) 配置SwitchA:

```

Switch (config)#ip dvmrp multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ip address 10.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)# ip dvmrp enable
  
```

(2) 配置SwitchB:

```

Switch (config)#ip dvmrp multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
Switch(Config-if-Vlan1)# ip dvmrp enable
Switch(Config-if-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-if-Vlan2)# ip address 20.1.1.1 255.255.255.0
  
```


Switch(Config-if-Vlan2)# ip dvmrp enable

由于 DVMRP 自己不依赖单播路由协议，因此无需配置单播路由协议，这是与 PIM-DM 和 PIM-SM 所不同的。

5.7.4 DVMRP 排错帮助

在配置、使用 DVMRP 协议时，可能会由于物理连接、配置错误等原因导致 DVMRP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 请检查接口是否已配置了正确的 IP 地址；
- ☞ 然后，在接口上启动 DVMRP 协议（使用 ip dvmrp enable 和 ip dvmrp multicast-routing 命令）；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性（DVMRP 使用自己的单播路由表，查看请使用 show ip dvmrp route 命令）。

如果仍无法解决 DVMRP 的问题，那么请使用 debug dvmrp 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.8 DCSCM

5.8.1 DCSCM 介绍

DCSCM（Destination Control and Source Control Multicast）技术主要包含三个方面，即组播信源可控、组播用户可控及面向服务优先级的策略组播。

受控组播技术的组播信源可控技术主要采取以下的方式进行：

1. 在边缘交换机上，如果配置的源受控组播，只有指定源发出的指定组的组播数据才能通过。
2. 对于处于 PIM-SM 核心地位的 RP 交换机，对于指定源及指定组以外的 REGISTER 信息，直接发送 REGISTER_STOP，而不允许建立表项。（该任务在 PIM-SM 模块中实现）。

受控组播技术的组播用户可控技术的实现基于对用户发出的 IGMP REPORT 报文的控制，因此进行控制的模块是 IGMP Snooping 和 IGMP 模块，其控制逻辑包括以下三种，即根据发送报文的 VLAN+MAC 地址进行控制，根据发送报文的 IP 地址进行控制和根据报文进入的端口进行控制。其中 IGMP Snooping 可以同时使用上述三种方式进行控制，而 IGMP 模块由于处于三层，仅针对发送报文的 IP 地址进行控制。

受控组播技术的面向服务优先级的策略组播采用下述的方式：对于限定范围的组播数据，在接入端就设置用户指定的优先级，使得数据在干道端口（TRUNK）上以更高的优先级发送，从而在整个网络上保证其以用户指定的优先级传送。

5.8.2 DCSCM 配置任务序列

1. 源受控的配置
2. 目的受控的配置
3. 组播策略的配置

1. 源受控的配置

源受控的配置分三个部分，首先是全局启动源受控，全局启动源受控的命令如下：

命令	解释
全局配置模式	
ip multicast source-control(必须) no ip multicast source-control	全局启动源控制，本命令的 no 操作全局关闭源控制。值得注意的是，全局启动源控制后，所有组播报文都默认丢弃。所有源控制配置都要在全局启动后才可以进行，而只有关闭所有已经配置的规则后，才可以全局关闭源控制。

其次是配置源控制的规则，它使用与 ACL 相同的方式配置，使用 5000—5099 的 ACL 号，每个规则号最多可配置 10 个规则，值得注意的是，这些规则是有顺序的，先配置的在最前面，一旦所配置的规则得到匹配，其后面的规则将不会起作用，所以全局允许的规则一定要配置在最后。其命令如下

命令	解释
全局配置模式	

<pre> access-list <5000-5099> {deny permit} ip {{<source> <source-wildcard>}}{<host-source <source-host-ip>}}{<any-source> {{<destination> <destination-wildcard>}}{<host-des tination <destination-host-ip>}}{<any-destin ation>} no access-list <5000-5099> {deny permit} ip {{<source> <source-wildcard>}}{<host-source <source-host-ip>}}{<any-source> {{<destination> <destination-wildcard>}}{<host-de stination <destination-host-ip>}}{<any-destin ation>} </pre>	用于配置源受控使用的规则，该规则只有应用到指定的端口上时才可以生效。使用其 NO 形式可以删除指定的规则。

最后是把配置的规则配置到指定的端口上。

注意：由于配置的规则要占据硬件的表项，配置过多的规则可能导致由于底层表项满而配置失败，因此建议用户尽可能使用简单的规则。配置命令如下

命令	解释
端口配置模式	
<pre> ip multicast source-control access-group <5000-5099> no ip multicast source-control access-group <5000-5099> </pre>	用于把源受控使用的规则配置到端口上，其 NO 形式取消该配置。

2. 目的受控的配置

目的受控的配置与源受控配置相似，也是分三个步骤。

首先要全局打开目的受控，由于目的受控要防止未获授权的用户接收到组播数据，因此配置全局目的受控后，交换机对收到的组播数据不再广播，因此，应当避免在启动了目的受控的交换机上同一 VLAN 内连接两台或以上的其它三层交换机。其配置命令如下

命令	解释
全局配置模式	

multicast destination-control(必须) no multicast destination-control	全局同时启动 IPv4 和 IPv6 目的受控组播,本命令的 no 操作全局关闭目的控制。所有其它的配置都只有在全局启动后才能生效。

其次是配置组播目的受控 profile 规则列表,使用 1—50 的 profile-id 号。

命令	解释
全局配置模式	
profile-id <1-50> {deny permit} {{<source/M> }}{host-source <source-host-ip> (range <2-65535>)}}any-source} {{<destination/M>}}{host-destination <destination-host-ip> (range <2-255>)}}any-destination} no profile-id <1-50>	配置目的受控的 profile 规则。其 no 操作用于删除该 profile 规则。

最后是配置目的控制规则,它与源控制相似,只是使用 6000—7999 的 ACL 号。

命令	解释
全局配置模式	
access-list <6000-7999> {{{add delete} profile-id WORD} {{deny permit} (ip) {{<source/M> }}{host-source <source-host-ip> (range <2-65535>)}}any-source} {{<destination/M>}}{host-destination <destination-host-ip> (range <2-255>)}}any-destination}} no access-list <6000-7999> {deny permit} ip {{<source> <source-wildcard>}}{host-source <source-host-ip> {range <2-65535>)}}any-source} {{<destination> <destination-wildcard>}}{host-destination <destination-host-ip> {range <2-255>)}}any-destination}	用于配置目的受控使用的规则,该规则只有应用到指定的源 IP 或 VLAN-MAC、端口上时才可以生效。使用其 NO 形式可以删除指定的规则。

最后要把该规则配置到指定的源 IP、源 VLAN MAC 或指定端口上。这里需要注意的是,由于上面所说的情况,只有在启动 IGMP-SNOOPING 的情况下才可以全面使用这些规则,

而如果没有启动 IGMP-SNOOPING，则只有源 IP 的规则才能在 IGMP 协议使用。其配置命令如下：

命令	解释
端口配置模式	
ip multicast destination-control access-group <6000-7999> no ip multicast destination-control access-group <6000-7999>	用于把目的受控使用的规则配置到端口上，其 NO 形式取消该配置。
全局配置模式	
ip multicast destination-control <1-4094> <macaddr> access-group <6000-7999> no ip multicast destination-control <1-4094> <macaddr> access-group <6000-7999>	用于把目的受控使用的规则配置到指定的 VLAN-MAC 上，其 NO 形式取消该配置。
ip multicast destination-control <IPADDRESS/M> access-group <6000-7999> no ip multicast destination-control <IPADDRESS/M> access-group <6000-7999>	用于把目的受控使用的规则配置到指定的源 IP 地址/掩码上，其 NO 形式取消该配置。

3. 组播策略的配置

组播策略使用为指定的组播数据指定优先级的方式达到保证特定用户需求的效果，值得注意的是组播数据只有在干道端口上传输才能保证数据一直得到重点照顾。其配置很简单，只有一个命令，即为指定的组播设定优先级，命令如下：

命令	解释
全局配置模式	
[no] ip multicast policy <IPADDRESS/M> <IPADDRESS/M> cos <priority>	配置组播策略，为特定范围的源、组指定优先级，范围为<0-7>。

5.8.3 DCSCM 典型案例

1. 源受控

为了防止一台边缘交换机随意发布组播数据，我们在边缘交换机上配置，只有在端口 Ethernet1/5 的交换机才允许发送组播数据，且数据的组必须是 225.1.2.3。而上连端口 Ethernet1/10 可以不受限制的转发组播数据，则我们可以进行如下配置。

```
Switch (config)#access-list 5000 permit ip any host 225.1.2.3
```

```
Switch (config)#access-list 5001 permit ip any any
```

```
Switch (config)#ip multicast source-control
```

```
Switch (config)#interface ethernet1/5
```

```
Switch (Config-If-Ethernet1/5)#ip multicast source-control access-group 5000
```

```
Switch (config)#interface ethernet1/10
```

```
Switch (Config-If-Ethernet1/10)#ip multicast source-control access-group 5001
```

2. 目的受控

我们要限制地址在 10.0.0.0/8 网段的用户不能加入 238.0.0.0/8 的组，我们可以做如下配置：

首先在其所在的 VLAN(这里认为是 VLAN2)内要启动 IGMP Snooping。

```
Switch (config)#ip igmp snooping
```

```
Switch (config)#ip igmp snooping vlan 2
```

然后配置相关的目的控制访问列表，并配置指定的 IP 地址使用该访问列表。

```
Switch (config)#access-list 6000 deny ip any 238.0.0.0 0.255.255.255
```

```
Switch (config)#access-list 6000 permit ip any any
```

```
Switch (config)#multicast destination-control
```

```
Switch (config)#ip multicast destination-control 10.0.0.0/8 access-group 6000
```

这样，该网段的用户就只能加入 238.0.0.0/8 以外的组了。

或者可以按照下面添加 profile 列表的方式进行配置相关的目的控制访问列表。

```
Switch (config)#profile-id 1 deny ip any 238.0.0.0 0.255.255.255
```

```
Switch (config)#access-list 6000 add profile-id 1
```

```
Switch (config)#multicast destination-control
```

```
Switch (config)#ip multicast destination-control 10.0.0.0/8 access-group 6000
```

3. 组播策略

服务器 210.1.1.1 正在组 239.1.2.3 上发布重要的组播数据，我们可在其接入交换机上配置如下：

```
Switch(config)#ip multicast policy 210.1.1.1/32 239.1.2.3/32 cos 4
```

这样该组播流在通过本交换机的 TRUNK 端口通向其它交换机时，将拥有值为 4 的优先级（通常这已经比较高了，更高的可能是协议数据了，若设置更高的优先级，在该组播数据太多时，有可能造成交换机协议行为的不正常）。

5.8.4 DCSCM 排错帮助

DCSCM 模块本身作用与 ACL 相似，发生问题通常与不当的配置有关，请您详细阅读上面的说明，如果仍不能确定问题发生的原因，请把您的配置和希望达到的效果发送给本公司的售后服务人员。

5.9 IGMP

5.9.1 IGMP 介绍

IGMP (Internet Group Management Protocol, 因特网组管理协议) 是 TCP/IP 协议族中负责 IP 组播成员管理的协议。它用来在 IP 主机和与其直接相邻的组播交换机之间建立、维护组播组成员关系。IGMP 不包括组播交换机之间的组成员关系信息的传播与维护, 这部分工作由各组播路由协议完成。所有参与组播的主机必须实现 IGMP 协议。

参与 IP 组播的主机可以在任意位置、任意时间、成员总数不受限制地加入或退出组播组。组播交换机不需要也不可能保存所有主机的成员关系, 它只是通过 IGMP 协议了解每个接口连接的网段上是否存在某个组播组的接收者, 即组成员。而主机方只需要保存自己加入了哪些组播组。

IGMP 在主机与路由器之间是不对称的: 主机需要响应组播交换机的 IGMP 查询报文, 即, 以成员资格报告报文响应; 交换机周期性发送成员资格查询报文, 然后根据收到的响应报文确定某个特定组在自己所在子网上是否有主机加入, 并且当收到主机的退出组的报告时, 发出特定组的查询 (IGMP 版本 2), 以确定某个特定组是否已无成员存在。

到目前为止, IGMP 有三个版本: IGMP 版本 1 (由 RFC1112 定义)、IGMP 版本 2 (由 RFC2236 定义) 和 IGMP 版本 3 (RFC3376 定义)。

IGMP 版本 2 对版本 1 所做的改进主要有:

1. 共享网段上组播交换机的选举机制

共享网段即一个网段上有多个组播交换机的情况。在这种情况下, 由于此网段下运行 IGMP 的交换机都能从主机那里收到成员资格报告消息, 因此, 只需要一个交换机发送成员资格查询消息, 这就需要一个交换机选举机制来确定一个交换机作为查询器。在 IGMP 版本 1 中, 查询器的选择由组播路由协议决定; IGMP 版本 2 对此做了改进, 规定同一网段上有多个组播交换机时, 具有最低 IP 地址的组播交换机被选举出来充当查询器。

2. IGMP 版本 2 增加了离开组机制

在 IGMP 版本 1 中, 主机悄然离开组播组, 不会给任何组播交换机发出任何通知。造成组播交换机只能依靠组播组响应超时来确定组播成员的离开。而在版本 2 中, 当一个主机决定离开一个组播组时, 如果它是对最近一条成员资格查询消息作出响应的主机, 那么它就会发送一条离开组的消息。

3. IGMP 版本 2 增加了对特定组的查询

在 IGMP 版本 1 中, 组播交换机的一次查询, 是针对该网段下的所有组播组。这种查询称为普通组查询。在 IGMP 版本 2 中, 在普通组查询之外增加了特定组的查询, 这种查询报文的目 IP 地址为该组播组的 IP 地址, 报文中的组地址域部分也为该组播组的 IP 地址。这样就避免了属于其它组播组成员的主机发送响应报文。

4. IGMP 版本 2 增加了最大响应时间字段

IGMP 版本 2 增加最大响应时间字段, 以动态地调整主机对组查询报文的响应时间。

版本3主要特点是允许主机选择接收或者拒绝特定的源，是SSM（Source-Specific Multicast）组播的基础。例如：当一个主机对于某个组G发出INCLUDE{10.1.1.1, 10.1.1.2}的报告时，意味着主机需要路由器转发组G中来自10.1.1.1和10.1.1.2的流量；当一个主机对组G发出EXCLUDE{192.168.1.1}的报告时，意味着主机需要组G的所有源发来的流量，除了192.168.1.1。这就和之前的IGMP有了很大的区别。

- IGMP 版本3对IGMP 版本2所作的改进主要有：
1. 维护的状态为组和源列表，不仅仅是 IGMPv2 中的组。
 2. 在 IGMPv3 状态中定义了与 IGMPv1 和 IGMPv2 的互操作。
 3. IP 服务接口改变，从而允许特定的源列表。
 4. 查询者在查询分组中包含其 Robustness Variable 和 Query Interval 以允许与非查询者的这些变量同步。
 5. 查询消息中的 Max Response Time 有一个指数级的范围，最大值从 v2 的 25.5 秒增加到约 53 分钟，可用于有海量系统的链路。
 6. 为增强健壮性，主机重传 State-Change 消息。
 7. 定义了附加数据以适应将来的扩展。
 8. 报告分组被发送到 224.0.0.22，用于协助二层交换机的 IGMP Snooping。
 9. 报告分组可以包含多个组记录，允许使用少量分组报告完整的当前状态。
 10. 主机不再进行抑止操作，简化了实现并且允许直接的成员资格跟踪。
 11. 在查询消息中新的路由器方抑止处理（S 标志）修改了 IGMPv2 存在的健壮性问题。

5.9.2 配置任务序列

- 1、启动 IGMP（必须）
- 2、配置 IGMP 辅助参数（可选）
 - (1) 配置 IGMP 组 group 参数
 - 1) 配置 IGMP 组过滤条件
 - 2) 配置 IGMP 加入组
 - 3) 配置 IGMP 加入静态组
 - (2) 配置 IGMP 查询参数
 - 1) 配置 IGMP 发送查询报文的时间间隔
 - 2) 配置 IGMP 查询的最大反应时间
 - 3) 配置 IGMP 查询的超时时间
 - (3) 配置 IGMP 版本
- 3、关闭 IGMP 协议

1. 启动 IGMP 协议
- 在三层交换机上没有启动 IGMP 协议的专门命令，在相应接口下启动任何一种组播协议，则 IGMP 自动随之启动。

命令	解释

全局配置模式	
ip dvmrp multicast-routing ip pim multicast-routing	启动全局组播协议，是启动 IGMP 协议的必要前提，相应命令的 no 操作关闭组播协议以及 IGMP 协议。（必须）

命令	解释
接口配置模式	
ip dvmrp enable ip pim dense-mode ip pim sparse-mode	启动 IGMP 协议，相应命令的 no 操作关闭 IGMP 协议。（必须）

2. 配置 IGMP 辅助参数

(1) 配置 IGMP 组 group 参数

- 1) 配置 IGMP 组过滤条件
- 2) 配置 IGMP 加入组
- 3) 配置 IGMP 加入静态组

命令	解释
接口配置模式	
ip igmp access-group {<acl_num acl_name>} no ip igmp access-group	配置接口对 IGMP 组的过滤条件；本命令的 no 操作取消过滤条件。
ip igmp join-group <A.B.C.D> no ip igmp join-group <A.B.C.D>	配置接口加入某个 IGMP 组；本命令的 no 操作取消加入。
ip igmp static-group <A.B.C.D> no ip igmp static-group <A.B.C.D>	配置接口加入某个 IGMP 静态组；本命令的 no 操作取消加入。

(2) 配置 IGMP 查询参数

- 1) 配置 IGMP 发送查询报文的时间间隔
- 2) 配置 IGMP 查询的最大响应时间
- 3) 配置 IGMP 查询的超时时间

命令	解释
接口配置模式	
ip igmp query-interval <time_val> no ip igmp query-interval	配置周期发送 IGMP 查询消息的时间间隔；本命令的 no 操作恢复缺省值。
ip igmp query-max-response-time <time_val> no ip igmp query-max-response-time	配置接口对 IGMP 查询的最大反应时间；本命令的 no 操作恢复缺省值。

ip igmp query-timeout <time_val> no ip igmp query-timeout	配置接口对 IGMP 查询的超时时间；本命令的 no 操作恢复缺省值。

(3) 配置 IGMP 版本

命令	解释
全局配置模式	
ip igmp version <version> no ip igmp version	配置接口上 IGMP 版本；本命令的 no 操作恢复为缺省值。

3. 关闭 IGMP 协议

命令	解释
接口配置模式	
no ip dvmrp enable no ip pim dense-mode no ip pim sparse-mode no ip pim multicast-routing (全局配置模式) no ip pim multicast-routing (全局配置模式)	关闭 IGMP 协议。

5.9.3 IGMP 典型案例

如下图，将switchA 和switchB 的以太网端口加入相应的VLAN，并在各VLAN接口上启动PIM-DM。

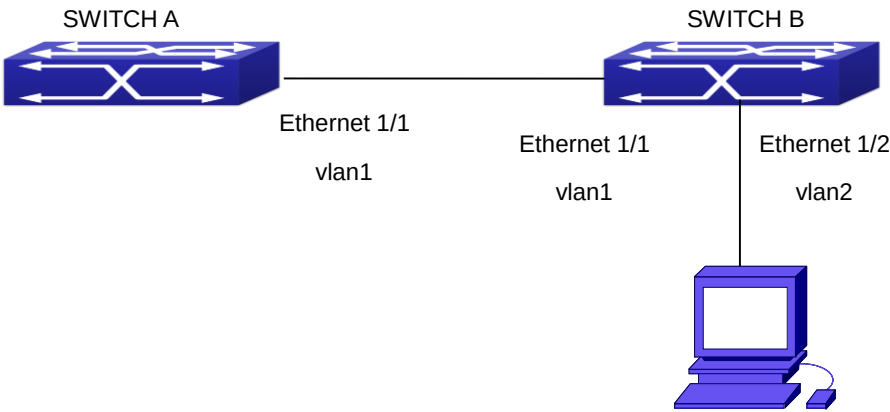


图 5-10 IGMP 网络拓扑示意图

switchA 和 switchB 配置步骤如下：

(1) 配置 SwitchA:

```
Switch(config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)#ip address 12.1.1.1 255.255.255.0
Switch(Config-If-Vlan1)#ip pim dense-mode
```

(2) 配置 SwitchB:

```
Switch(config)#ip pim multicast-routing
Switch(config)#interface vlan1
Switch(Config-If-Vlan1)#ip address 12.1.1.2 255.255.255.0
Switch(Config-If-Vlan1)#ip pim dense-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan2
Switch(Config-If-Vlan1)#ip address 20.1.1.1 255.255.255.0
Switch(Config-If-Vlan2)#ip pim dense-mode
Switch(Config-If-Vlan2)#ip igmp version 3
```

5.9.4 IGMP 排错帮助

在配置、使用 IGMP 协议时，可能会由于物理连接、配置错误等原因导致 IGMP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后，确保在接口上启动一种组播协议；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性。

5.10 IGMP Snooping

5.10.1 IGMP Snooping 介绍

IGMP (Internet Group Management Protocol) 互联网组管理协议，用于实现 IP 的组播。IGMP 被支持组播的网络设备（如路由器）用来进行主机资格查询，也被想加入某组播组的主机用来通知路由器接收某个组播地址的数据包，而这些都是通过 IGMP 消息交换来完成的。路由器首先利用一个可寻址到所有主机的组地址（即 224.0.0.1）发送一条 IGMP 主机成员资格查询（IGMP Host Membership Query）消息。若一个主机希望加入某组播组，它就利用该组播组的组地址回应一条 IGMP 主机成员资格报告（IGMP Host Membership Report）消息。

IGMP Snooping 即 IGMP 侦听。交换机通过 IGMP Snooping 来限制组播流量的泛滥，只把组播流量转发给与组播设备相连的端口。交换机侦听组播路由器和主机之间的 IGMP 消息，根据侦听结果维护组播转发表，而交换机根据组播转发表来决定组播包的转发。

交换机实现了 IGMP Snooping 功能，并且支持 IGMP v3，这样，用户可以用交换机实现 IP 组播。

5.10.2 IGMP Snooping 配置任务

1. 启动 IGMP Snooping 功能

2. 配置 IGMP Snooping

1. 启动 IGMP Snooping 功能

命令	解释
全局配置模式	
ip igmp snooping no ip igmp snooping	启动 IGMP Snooping 功能。no 操作关闭全局 IGMP snooping 功能。

2. 配置 IGMP Snooping

命令	解释
全局配置模式	
ip igmp snooping vlan <vlan-id> no ip igmp snooping vlan <vlan-id>	启动指定 VLAN 的 IGMP Snooping 功能。 No 操作关闭指定 VLAN 上的 IGMP Snooping。
ip igmp snooping proxy no ip igmp snooping proxy	开启 IGMP Snooping 代理功能，本命令的 no 操作为关闭 IGMP Snooping 代理功能。
ip igmp snooping vlan <vlan-id> limit {group <g_limit> source <s_limit>} no ip igmp snooping vlan <vlan-id> limit	设置 IGMP snooping 可加入组的个数和每个组中源个数的最大值。no 操作恢复默认值。
ip igmp snooping vlan <1-4094> interface (ethernet port-channel) IFNAME limit {group <1-65535> source <1-65535>} strategy (replace drop) no ip igmp snooping vlan <1-4094> interface (ethernet port-channel) IFNAME limit group source strategy	设置 IGMP Snooping 端口下可允许加入组的个数和每个组中源个数的最大值，及当达到上限值时的处理策略：包括替换和丢弃。 no 操作设置为“无限制”。
ip igmp snooping vlan <vlan-id> l2-general-querier no ip igmp snooping vlan <vlan-id>	将该 VLAN 设为二层普通查询者，每一个网段推荐配置一个二层普通查询者。no 操作取消二层普通查询者设置。

I2-general-querier	
ip igmp snooping vlan <vlan-id> I2-general-querier-version <version>	该命令配置二层查询者发送普通查询的版本号。
ip igmp snooping vlan <vlan-id> I2-general-querier-source <source>	该命令配置二层查询者发送普通查询的源地址。
ip igmp snooping vlan <vlan-id> mrrouter-port interface <interface – name> no ip igmp snooping vlan <vlan-id> mrrouter-port interface <interface – name>	设置指定 VLAN 内的静态 mrrouter 端口。no 操作取消 mrrouter 端口设置。
ip igmp snooping vlan <vlan-id> mrrouter-port learnpim no ip igmp snooping vlan <vlan-id> mrrouter-port learnpim	打开指定 VLAN 根据 pim 报文学习 mrrouter-port 的功能；本命令的 no 操作为关闭指定 VLAN 根据 pim 报文学习 mrrouter-port 的功能。
ip igmp snooping vlan <vlan-id> mrpt <value> no ip igmp snooping vlan <vlan-id> mrpt	设置 mrrouter 端口存活时间，no 操作恢复默认值。
ip igmp snooping vlan <vlan-id> query-interval <value> no ip igmp snooping vlan <vlan-id> query-interval	设置查询间隔，no 操作恢复默认值。
ip igmp snooping vlan <vlan-id> immediately-leave no ip igmp snooping vlan <vlan-id> immediately-leave	设置指定 VLAN 的 IGMP Snooping 具有快速离开组播组功能，no 操作取消 immediately-leave 设置。
ip igmp snooping vlan <vlan-id> query-mrsp <value> no ip igmp snooping vlan <vlan-id> query-mrsp	设置查询的最大响应时间，no 操作恢复默认值。
ip igmp snooping vlan <vlan-id> query-robustness <value> no ip igmp snooping vlan <vlan-id> query-robustness	设置查询鲁棒值，no 操作恢复默认值。
ip igmp snooping vlan <vlan-id> suppression-query-time <value> no ip igmp snooping vlan <vlan-id>	设置抑制查询时间值，no 操作恢复默认值。

suppression-query-time	
ip igmp snooping vlan <vlan-id> static-group <A.B.C.D> [source< A.B.C.D>] interface [ethernet port-channel] <IFNAME> no ip igmp snooping vlan <vlan-id> static-group <A.B.C.D> [source< A.B.C.D>] interface [ethernet port-channel] <IFNAME>	设置静态组源, no 操作取消设置的静态组源。
ip igmp snooping vlan <vlan-id> report source-address <A.B.C.D> no ip igmp snooping vlan <vlan-id> report source-address	设置转发 IGMP 报文源地址, no 操作取消设置的报文源地址。
ip igmp snooping vlan <vlan-id> report source-mac proxy mode no ip igmp snooping vlan <vlan-id> report source-mac proxy mode	设置向 mrouter 口转发 IGMP report 报文的源 mac 为本交换机的 mac 地址, no 操作取消设置的报文源 mac。 注意: 本交换机不支持此命令。
ip igmp snooping vlan <vlan-id> specific-query-mrsp <value> no ip igmp snooping vlan <vlan-id> specific-query-mrspt	配置特定组/源查询的最大响应时间值, no 操作恢复默认值。

5.10.3 IGMP Snooping 典型案例

案例 1: IGMP Snooping 功能。

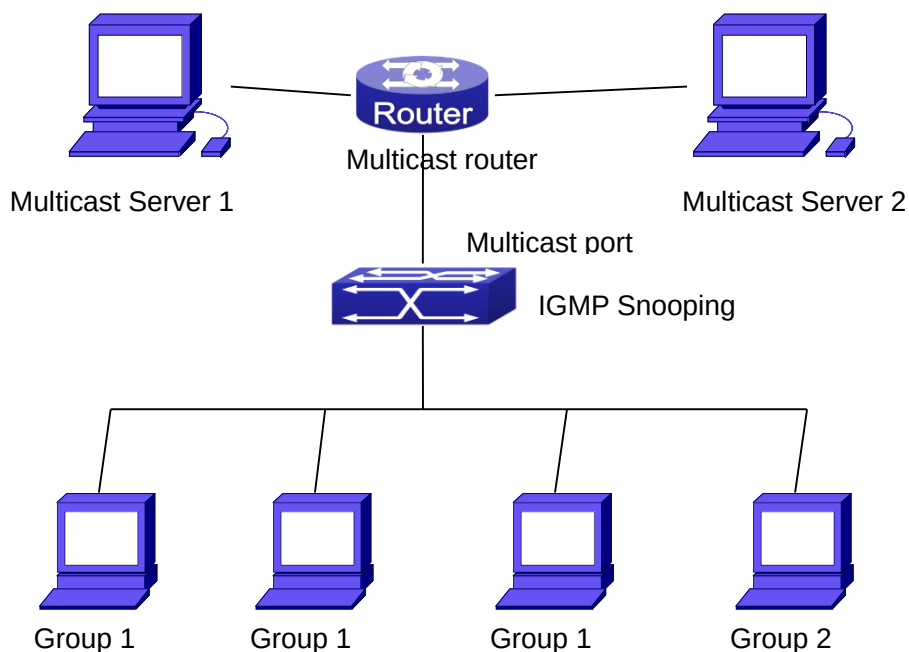


图 5-11 打开交换机 IGMP Snooping 功能图

如图所示，switch 上配置 VLAN 100 包含端口 1、2、6、10、12。四台主机分别连在端口 2、6、10、12 上，组播路由器连在端口 1 上。端口 1 因为与组播路由器相连，会被识别为 MROUTER 端口。假设我们需要在 VLAN 100 上做 IGMP Snooping，缺省情况下，交换机的全局 IGMP Snooping 功能和各 VLAN 上的 IGMP Snooping 功能都不打开。因此，需要打开全局下的 IGMP Snooping 功能，同时在 VLAN 100 上打开 IGMP Snooping，还需要设置 VLAN 100 的端口 1 为 mrouter 端口。如果组播路由器启动了三层组播，则 VLAN 100 的端口 1 被自动识别为 mrouter 端口，不需手动配置。

配置步骤如下：

```
Switch#config
```

```
Switch(config)#ip igmp snooping
```

```
Switch(config)#ip igmp snooping vlan 100
```

```
Switch(config)#ip igmp snooping vlan 100 mrouter-port interface ethernet 1/1
```

组播配置：

假设有两个组播服务器 Multicast Server 1 和 Multicast Server 2，其中组播服务器 1 上提供节目 1，组播服务器 2 上提供节目 2。分别使用组地址 Group1 和 Group 2 四台主机上同时运行组播应用软件，连在端口 2、6、10 上的三台主机播放节目 1，连在端口 12 上的主机播放节目 2。

IGMP Snooping 侦听结果：

VLAN 100 上 IGMP Snooping 建立的组播表显示：其中端口 1、2、6、10 在（Multicasting Server 1， Group1）中，端口 1、12 在（Multicast Server 2， Group 2）中。

四台主机都能正常地收到自己感兴趣的节目，端口 2、6、10 不会收到节目 2 的流量，端口 12 不会收到节目 1 的流量。

“show mac-address-table multicast”命令可以查看 switch B 上的二层组播转发表。注意：224.0.0.1~224.0.0.255 范围内的地址为预留组播地址，所以与这类地址的 MAC 地址（01-00-5e-00-00-xx）相同的组播组地址不应该被使用，因为他们不会下发到二层组播转发表。

案例 2: IGMP L2-general-querier

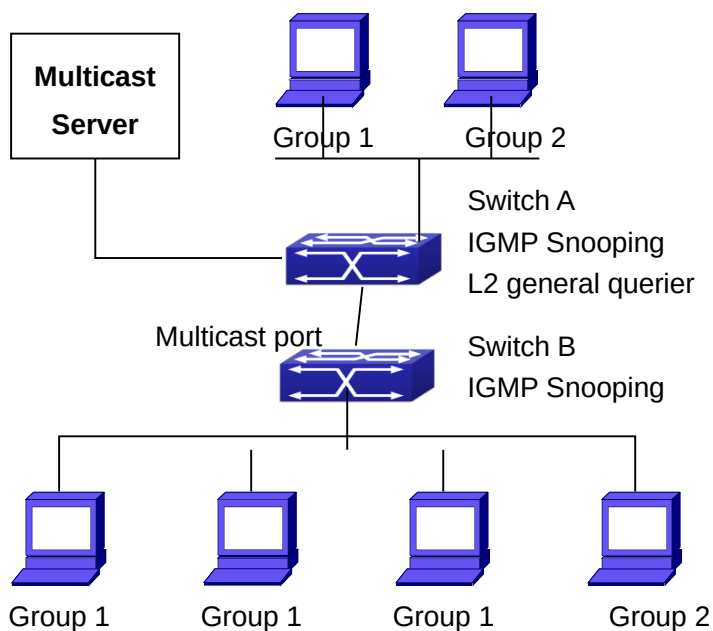


图 5-12 交换机作为 IGMP Querier 功能图

switch B 的配置与案例 1 中交换机相同，交换机 switch A 取代案例 1 中的 Multicast Router。假设其上配置 VLAN 60 包含端口 1、2、10、12。端口 1 连接组播服务器，端口 2 连接 switch B。为了定期发 Query，需要打开全局下的 IGMP Snooping 功能，同时需要执行 IGMP Snooping vlan 60 l2-general-querier，将 VLAN 60 设置成为二层普通查询者。

配置步骤如下：

```
SwitchA#config
SwitchA(config)#ip igmp snooping
SwitchA(config)#ip igmp snooping vlan 60
SwitchA(config)#ip igmp snooping vlan 60 l2-general-querier
```

```
SwitchB#config
SwitchB(config)#ip igmp snooping
```



```
SwitchB(config)#ip igmp snooping vlan 100
```

```
SwitchB(config)#ip igmp snooping vlan 100 mrouter interface ethernet 1/1
```

组播配置：

同案例 1。

IGMP Snooping 侦听结果：

与案例 1 相似。

案例 3： 与三层组播协议混合运行

我们将案例1中的SWITCH替换为ROUTER，配置与原SWITCH配置相同，组播配置与IGMP snooping侦听结果与案例1相同。在ROUTER上全局启动三层组播(PIM-SM)，同时在VLAN 100上启动PIM SM。(与上层组播路由器使用相同PIM模式)

配置步骤如下：

```
switch#config
```

```
switch(config)#ip pim multicast-routing
```

```
switch(config)#interface vlan 100
```

```
switch(config-if-vlan100)#ip pim sparse-mode
```

此时由于存在三层组播协议，IGMP snooping 不再下发表项，只负责以下几件事情：

- 删除二层组播表项
- 向三层提供端口查询函数，参数为 VLAN，S，G
- 当三层 IGMP 关闭时，重新下发二层组播表项

通过观察三层 IPMC 表项，发现三层组播出入表项仍然可以精确到端口，保证了 IGMP snooping 与三层组播协议的混合运行。

5.10.4 IGMP Snooping 排错帮助

在配置、使用 IGMP Snooping 功能时，可能会由于物理连接、配置错误等原因导致 IGMP Snooping 未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证全局配置模式下 IGMP Snooping 打开（使用 ip igmp snooping）
- ☞ 保证全局配置模式下 VLAN 上配置 IGMP Snooping（使用 ip igmp snooping vlan <vlan-id>）
- ☞ 确保同一网段内存在查询者，并确认是否存在 mrouter
- ☞ 使用 show ip igmp snooping vlan <vid>命令查看 IGMP Snooping 信息是否正确
- ☞ 如果使用检查仍无法解决 IGMP Snooping 的问题，那么请使用 debug igmp snooping 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心

5.11 IGMP Proxy

5.11.1 IGMP Proxy 介绍

IGMP/MLD proxy是在rfc4605中提出的一种简化的组播末端协议，其核心是在运行环境简单的组播协议末端，不运行复杂的PIM/DVMRP等组播路由协议，通过IGMP/MLD代理的方式与组播协议对话，从而简化在低端设备上的组播实现。

IGMP/MLD协议是用来在组播路由器和客户端间通信的协议，同时也存在组播路由器和客户端两种协议行为，IGMP/MLD代理设备在配置上明确上游接口和下游接口，对于上游接口，协议运行HOST端协议，在下游接口，则运行ROUTER端协议，IGMP/MLD代理交换机把从下游收集的IGMP/MLD加入信息汇聚成HOST端状态，以IGMP/MLD客户端的身份发送加入、离开消息给上层的组播路由器，从而实现不需组播路由协议的简单组播成员关系。

IGMP proxy 功能与 PIM 及 DVMRP的功能是互斥的。

5.11.2 IGMP Proxy 配置任务

1. 启动 IGMP Proxy 功能
2. 在不同接口上打开 IGMP Proxy 的上游接口和下游接口开关
3. 配置 IGMP Proxy

1. 启动 IGMP Proxy 功能

命令	解释
全局配置模式	
ip igmp proxy no ip igmp proxy	启动 IGMP Proxy 功能。本命令的 no 操作关闭全局 IGMP Proxy 功能。

2. 在不同接口上打开 IGMP Proxy 的上游接口和下游接口开关

命令	解释
接口配置模式	
ip igmp proxy upstream no ip igmp proxy upstream	启动 IGMP Proxy 上游接口功能。本命令的 no 操作关闭 IGMP Proxy 上游接口功能。
ip igmp proxy downstream no ip igmp proxy downstream	启动 IGMP Proxy 下游接口功能。本命令的 no 操作关闭 IGMP Proxy 下游接口功能。

3. 配置 IGMP Proxy 辅助参数

命令	解释
全局配置模式	

ip igmp proxy limit {group <1-500> source <1-500>} no ip igmp proxy limit	设置 IGMP Proxy 上游接口可加入组的个数和每个组中源个数的最大值。本命令的 no 操作恢复默认值。
ip igmp proxy unsolicited-report interval <1-5> no ip igmp proxy unsolicited-report interval	设置 IGMP Proxy 上游接口状态改变报告重传时间间隔。本命令的 no 操作恢复默认值。
ip igmp proxy unsolicited-report robustness <2-10> no ip igmp proxy unsolicited-report robustness	设置 IGMP Proxy 上游接口状态改变报告重传个数。本命令的 no 操作恢复默认值。
ip igmp proxy aggregate no ip igmp proxy aggregate	设置 IGMP Proxy 非查询者下游接口也可以作为下发表项的出接口。本命令的 no 操作恢复默认设置。
ip multicast ssm range <1-99> ip multicast ssm default no ip mulitcast ssm	设置 IGMP Proxy SSM 组播组地址范围；本命令的 no 操作删除配置的 igmp proxy ssm 组播组。
ip igmp proxy multicast-source no ip igmp proxy multicast-source	设置指定下游接口作为组播数据源接口；本命令的 no 操作取消该下游接口作为组播数据源接口。

5.11.3 IGMP Proxy 举例

案例 1：IGMP Proxy 功能。

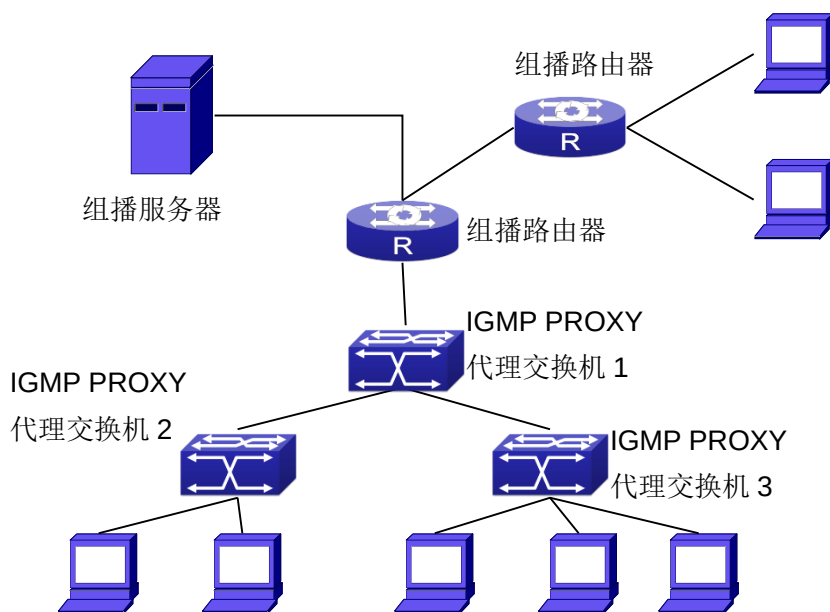


图 5-13 打开交换机 IGMP Proxy 功能图

如图所示，IGMP PROXY 代理交换机以树型拓扑存在于简单的网络环境中，组播流量沿着树枝从上游接口向下游扩散，而 igmp 组播成员报告沿着树枝从下游向上游汇聚，通知上游接口。三台 IGMP PROXY 代理交换机以树型拓扑组织，分别有一个接口与上层路由器相连，一个或多个下游接口或者直接跟主机相连，或者跟 IGMP PROXY 代理交换机的上游接口相连。

配置步骤如下：

```
Switch#config
Switch(config)#ip igmp proxy
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip igmp proxy upstream
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip igmp proxy downstream
```

组播配置：

假设有组播服务器上提供组播地址 224.1.1.1 的节目，网络末端有主机点播该节目，igmp 组播成员报告通知 IGMP PROXY 代理交换机 2 和 3 的下游接口，IGMP PROXY 代理交换机 2 和 3 的上游接口汇聚该组播成员关系，再向 IGMP PROXY 代理交换机 1 点播，通知上层组播路由器。当有组播流量到达时，IGMP PROXY 代理交换机的上游接口将组播成员关系下发，组播流量按照 IGMP PROXY 下发的出接口转发。

案例 2：IGMP Proxy 下游接口组播数据源功能。

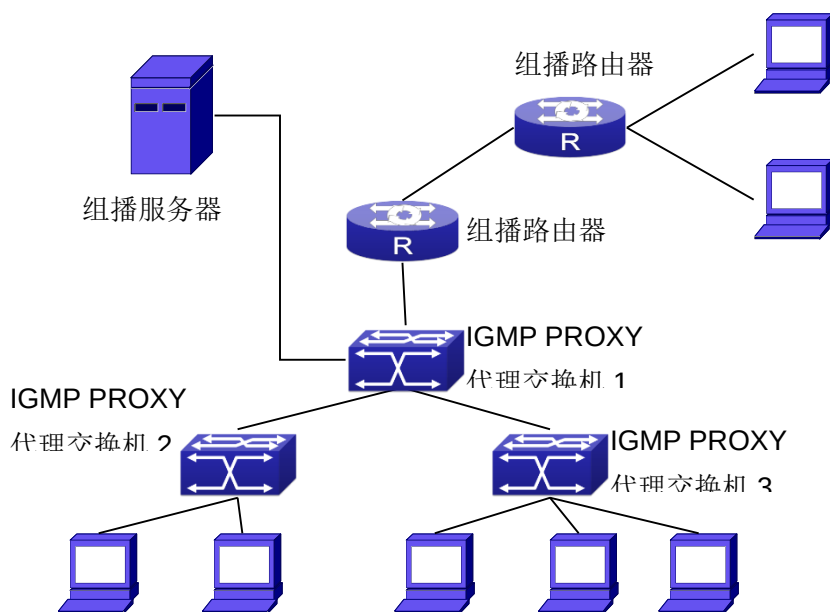


图 5-14 打开交换机 IGMP Prox 下游接口组播源功能图

如图所示，IGMP PROXY 代理交换机以树型拓扑存在于简单的网络环境中，组播源位于代理交换机 1 的下游，组播流量到达时，默认上游接口作为出接口，沿着树枝从下游接口向上游和其他下游接口扩散。三台 IGMP PROXY 代理交换机以树型拓扑组织，分别有一个接口与上层路由器相连，一个或多个下游接口或者直接跟主机相连，或者跟 IGMP PROXY 代理交换机的上游接口相连。

配置步骤如下：

IGMP PROXY 代理交换机 1 配置：

```
Switch#config
Switch(config)#ip igmp proxy
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip igmp proxy upstream
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip igmp proxy downstream
Switch(Config-if-Vlan2)#ip igmp proxy multicast-source
```

组播路由器 1 配置：

```
Switch#config
Switch(config)#ip pim multicast
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip pim sparse-mode
Switch(Config-if-Vlan1)#ip pim bsr-border
```

组播配置：

假设有组播服务器上提供组播地址 224.1.1.1 的节目，网络末端有主机点播该节目，igmp 组播成员报告通知 IGMP PROXY 代理交换机 2 和 3 的下游接口，IGMP PROXY 代理交换机 2 和 3 的上游接口汇聚该组播成员关系，再向 IGMP PROXY 代理交换机 1 点播，通知上层组播路由器。当有组播流量到达时，IGMP PROXY 代理交换机的上游接口将组播成员关系下发，下发接口中默认加入上游接口，组播流量按照 IGMP PROXY 下发的出接口转发。组播路由器收到代理服务器发出的组播流量，认为所有该接口接收的组播数据源均为直连，而确定 DR 和 ORIGINATOR 的身份，组播数据按照 PIM 协议继续转发。

5.11.4 IGMP Proxy 排错帮助

在配置、使用 IGMP Proxy 功能时，可能会由于物理连接、配置错误等原因导致 IGMP Proxy 未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误；
- ☞ 保证全局配置模式下 IGMP Proxy 打开（使用 ip igmp proxy）；
- ☞ 保证接口配置模式下一个上游接口和至少一个下游接口（使用 ip igmp proxy upstream, ip igmp proxy downstream）；
- ☞ 使用 show ip igmp proxy 命令查看 IGMP Proxy 配置信息是否正确。

如果使用检查仍无法解决 IGMP Proxy 的问题，那么请使用 debug igmp proxy 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

5.12 组播 VLAN

5.12.1 组播 VLAN 介绍

基于当前的组播点播方式，当处于不同的VLAN 的用户点播时，每个VLAN 会在本VLAN 内复制一个组播流。这样的组播点播方式，浪费了大量的带宽。因此我们通过配置组播VLAN 的方式，将交换机的端口加入到组播VLAN 内，并在使能了IGMP Snooping /MLD Snooping功能以后，使不同VLAN 内的用户公用一个组播VLAN，组播流只是在一个组播VLAN 内进行传输，从而节省了带宽。由于组播VLAN 与用户VLAN 完全隔离，因此安全和带宽都得以保证。在配置了组播VLAN 以后就保证了组播信息流能够持续不断的发送到用户。

5.12.2 组播 VLAN 配置任务

1. 启动组播 VLAN 功能
2. 配置 IGMP Snooping

1. 启动组播 VLAN 功能

命令	解释
VLAN 配置模式	
multicast-vlan no multicast-vlan	配置一个 VLAN 启动组播 VLAN 功能。no 操作关闭该 VLAN 的组播 VLAN 功能。
multicast-vlan association <vlan-list> no multicast-vlan association <vlan-list>	将一个组播 VLAN 与多个 VLAN 关联。No 操作删除组播 VLAN 的关联 VLAN。
multicast-vlan association interface (ethernet port-channel) IFNAME no multicast-vlan association interface (ethernet port-channel) IFNAME	将特定端口关联到组播 VLAN 上，这样关联到组播 VLAN 的端口上都能收到组播流量；no 命令取消关联端口关联关系。
multicast-vlan mode {dynamic compatible} no multicast-vlan mode {dynamic compatible}	配置组播 vlan 的两种模式；本命令的 no 操作为取消组播 VLAN 的模式配置。

2. 配置 IGMP Snooping

命令	解释
全局配置模式	
ip igmp snooping vlan <vlan-id> no ip igmp snooping vlan <vlan-id>	启动组播 VLAN 的 IGMP Snooping 功能。 No 操作关闭组播 VLAN 上的 IGMP Snooping。
ip igmp snooping no ip igmp snooping	启动 IGMP Snooping 功能。no 操作关闭全局 IGMP Snooping 功能。

5.12.3 组播 VLAN 举例

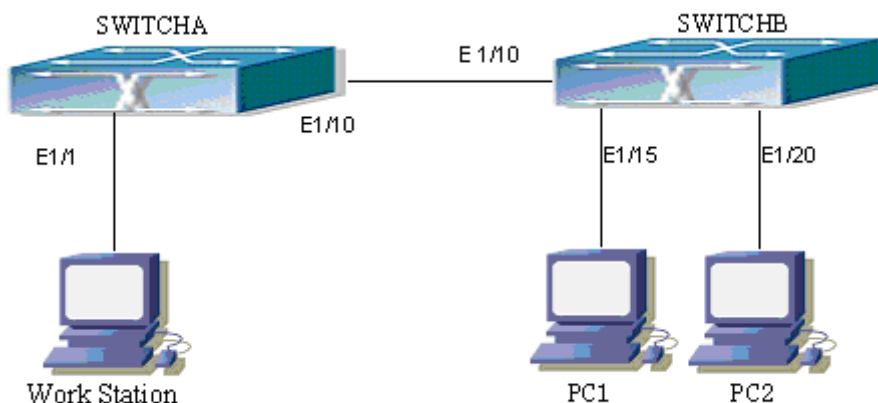


图 5-15 组播 VLAN 功能配置图

如图所示，组播服务器通过端口1/1与三层交换机switchA相连，端口1/1属于交换机的VLAN 10。三层交换机switchA通过端口1/10与二层交换机switchB相连，端口1/10配置为trunk模式。switchB上配置VLAN 100包含端口1/15，VLAN 101包含端口1/20。PC1和PC2分别连在端口1/15和1/20上。switchB通过端口1/10与交换机switchA相连，端口1/10配置为trunk模式。VLAN 20为组播VLAN。

通过配置组播VLAN，使PC1和PC2通过组播VLAN接收组播数据。

以下配置基于switchA的IP地址已经配置，并且设备连接正确的情况下进行配置。

配置步骤如下：

```
SwitchA#config
SwitchA(config)#vlan 10
SwitchA(Config-vlan10)#switchport interface ethernet 1/1
SwitchA(Config-vlan10)#exit
SwitchA(config)#interface vlan 10
SwitchA(Config-if-Vlan10)#ip pim dense-mode
SwitchA(Config-if-Vlan10)#exit
SwitchA(config)#vlan 20
SwitchA(config-vlan20)#exit
SwitchA(config)#interface vlan 20
SwitchA(Config-if-Vlan20)#ip pim dense-mode
SwitchA(Config-if-Vlan20)#exit
SwitchA(config)#ip pim multicast
SwitchA(config)#interface ethernet 1/10
SwitchA(Config-If-Ethernet1/10)#switchport mode trunk
```

```
SwitchB#config
SwitchB(config)#vlan 100
```



```
SwitchB(Config-vlan100)#switchport interface ethernet 1/15
SwitchB(Config-vlan100)#exit
SwitchB(config)#vlan 101
SwitchB(Config-vlan101)#switchport interface ethernet 1/20
SwitchB(Config-vlan101)#exit
SwitchB(config)#interface ethernet 1/10
SwitchB(Config-If-Ethernet1/10)#switchport mode trunk
SwitchB(Config-If-Ethernet1/10)#exit
SwitchB(config)#vlan 20
SwitchB(Config-vlan20)#multicast-vlan
SwitchB(Config-vlan20)#multicast-vlan association 100,101
SwitchB(Config-vlan20)#exit
SwitchB(config)#ip igmp snooping
SwitchB(config)#ip igmp snooping vlan 20
```

组播 VLAN 支持 IPV6 组播时,使用方法同 IPV4 原理相同,不同的只是与 MLD Snooping 配合使用, 这里不再单独举例。

第6章 安全功能操作

6.1 ACL

6.1.1 ACL 介绍

ACL (Access Control Lists)是交换机实现的一种数据包过滤机制，通过允许或拒绝特定的数据包进入网络，交换机可以对网络访问进行控制，有效保证网络的安全运行。用户可以基于报文中的特定信息制定一组规则（rule），每条规则都描述了对匹配一定信息的数据包所采取的动作：允许通过（permit）或拒绝通过（deny）。用户可以把这些规则应用到特定交换机端口的入口，这样特定端口上特定方向的数据流就必须依照指定的 ACL 规则进入交换机。

Egress ACL 可以在出口和入口方向实现对报文的过滤，符合特定规则的报文可以允许通过或者禁止通过。ACL 可以支持IP ACL，MAC ACL，MAC-IP ACL，IPv6 ACL。在端口入口方向上可以同时配置四种ACL；在端口出口方向上由于硬件只有四个资源，IP和MAC ACL各占一个资源，MAC-IP和IPv6 ACL各占两个资源，当硬件资源占满后则无法再配置，即一个端口在出口方向上不能同时配置四种ACL，同时配置三种ACL时也只能是IP、MAC和MAC-IP类型同时配置或IP、MAC和IPv6类型同时配置，同时配置两种ACL时可以任意选取，没有限制。一种类型的ACL在端口上只能配置一条。

6.1.1.1 Access-list

Access-list是一个有序的语句集，每一条语句对应一条特定的规则(rule)。每条rule包括了过滤信息及匹配此rule时应采取的动作。Rule包含的信息可以包括源MAC、目的MAC、源IP、目的IP、IP协议号、TCP端口、UDP端口等条件的有效组合。根据不同的标准，access-list可以有如下分类：

1. 根据过滤信息：ip access-list, ipv6 access-list（三层以上信息），mac access-list（二层信息），mac-ip access-list（二层以上信息）。
2. 根据配置的复杂程度：标准(standard)和扩展(extended)，扩展方式可以指定更加细致过滤信息。
3. 根据命名方式：数字(numbered)和命名(named)。

对一条ACL的说明应当从这三个方面加以描述。

6.1.1.2 Access-group

当用户按照实际需要制定了一组access-list之后，就可以把他们分别应用到不同端口的入口方向上。access-group就是对特定的一条access-list与特定端口的绑定关系的描述。当

您建立了一条access-group之后，流经此端口此方向的所有数据包都会试图匹配指定的access-list规则，以决定交换动作是允许(permit)或拒绝(deny)。另外还可以在端口加上对ACL规则统计计数器，以便统计符合ACL规则数据包的数量。

6.1.1.3 Access-list 动作及全局默认动作

Access-list动作及默认动作分为两种：允许通过(permit)或拒绝通过(deny)。具体有如下：

4. 在一个access-list内，可以有多条规则(rule)。对数据包的过滤从第一条规则(rule)开始，直到匹配到一条规则(rule)，其后的规则(rule)不再进行匹配。
5. 只有在包过滤功能打开且端口上没有绑定任何的ACL或不匹配任何绑定的ACL时才会匹配入口的全局的默认动作。

6.1.2 ACL 配置

ACL配置任务序列如下：

1. 配置 access-list

- (1) 配置数字标准 IP 访问列表
- (2) 配置数字扩展 IP 访问列表
- (3) 配置命名标准 IP 访问列表
 - a) 创建一个命名标准 IP 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出访问表配置模式
- (4) 配置命名扩展 IP 访问列表
 - a) 创建一个命名扩展 IP 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出访问表配置模式
- (5) 配置数字标准 MAC 访问列表
- (6) 配置数字扩展 MAC 访问列表
- (7) 配置命名扩展 MAC 访问列表
 - a) 创建一个命名扩展 MAC 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出 MAC 访问表配置模式
- (8) 配置数字扩展 MAC-IP 访问列表
- (9) 配置命名扩展 MAC-IP 访问列表
 - a) 创建一个命名扩展 MAC-IP 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出 MAC-IP 访问表配置模式
- (10) 配置数字标准 IPv6 访问列表
- (11) 配置数字扩展 IPv6 访问列表

- (12) 配置命名标准 IPv6 访问列表
 - a) 创建一个命名标准 IPv6 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出 IPv6 访问表配置模式
- (13) 配置命名扩展 IPv6 访问列表
 - a) 创建一个命名扩展 IPv6 访问列表
 - b) 指定多条 permit 或 deny 规则表项
 - c) 退出 IPv6 访问表配置模式
- 2. 配置包过滤功能
 - (1) 全局打开包过滤功能
 - (2) 全局配置 ACL deny 优先功能模块（可选）
- 3. 配置时间范围功能
 - (1) 创建时间范围名称
 - (2) 配置周期性时段
 - (3) 配置绝对性时段
- 4. 将 access-list 绑定到特定端口的特定方向
- 5. 显示端口 ACL 的配置及统计信息
- 6. 清空指定接口的包过滤统计信息

1. 配置 access-list

(1) 配置数字标准 IP 访问列表

命令	解释
全局配置模式	
access-list <num> {deny permit} {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} no access-list <num>	创建一条数字标准 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项；本命令的 no 操作为删除一条数字标准 IP 访问列表。

(2) 配置数字扩展 IP 访问列表

命令	解释
全局配置模式	
access-list <num> {deny permit} icmp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [<icmp-type> [<icmp-code>]] [precedence <prec>] [tos <tos>][time-range<time-range-name>]	创建一条 ICMP 数字扩展 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。

<pre>access-list <num> {deny permit} igmp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [<igmp-type>] [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 IGMP 数字扩展 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>access-list <num> {deny permit} tcp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [ack+fin+psh+rst+urg+syn] [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 TCP 数字扩展 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>access-list <num> {deny permit} udp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 UDP 数字扩展 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>access-list <num> {deny permit} {eigrp gre igrp ipinip ip ospf < protocol-num >} {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条匹配其他特定 IP 协议或所有 IP 协议的数字扩展 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>no access-list <num></pre>	删除一条数字扩展 IP 访问列表。

(3) 配置命名标准 IP 访问列表

a. 创建一个命名标准 IP 访问列表

命令	解释
全局配置模式	

ip access-list standard <name> no ip access-list standard <name>	创建一条命名标准 IP 访问列表；本命令的 no 操作为删除此命名标准 IP 访问列表。

b. 指定多条 permit 或 deny 规则

命令	解释
命名标准 IP 访问列表配置模式	
[no] {deny permit} {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}}	创建一条命名标准 IP 访问规则 (rule)；本命令的 no 操作为删除此命名标准 IP 访问规则 (rule)。

c. 退出命名标准 IP 访问列表配置模式

命令	解释
命名标准 IP 访问列表配置模式	
exit	退出命名标准 IP 访问列表配置模式。

(4) 配置命名扩展 IP 访问列表

a. 创建一个命名扩展 IP 访问列表

命令	解释
全局配置模式	
ip access-list extended <name> no ip access-list extended <name>	创建一条命名扩展 IP 访问列表；本命令的 no 操作为删除此命名扩展 IP 访问列表。

b. 指定多条 permit 或 deny 规则

命令	解释
命名扩展 IP 访问列表配置模式	
[no] {deny permit} icmp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [<icmp-type> [<icmp-code>]] [precedence <prec>] [tos <tos>][time-range<time-range-name>]	创建一条 icmp 命名扩展 IP 访问规则 (rule)；本命令的 no 操作为删除此命名扩展 IP 访问规则 (rule)。
[no] {deny permit} igmp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [<igmp-type>] [precedence <prec>] [tos <tos>][time-range<time-range-name>]	创建一条 igmp 命名扩展 IP 访问规则 (rule)；本命令的 no 操作为删除此命名扩展 IP 访问规则 (rule)。

<pre>[no] {deny permit} tcp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [ack+fin+psh+rst+urg+syn] [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 tcp 命名扩展 IP 访问规则 (rule); 本命令的 no 操作为删除此命名扩展 IP 访问规则(rule)。
<pre>[no] {deny permit} udp {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 udp 命名扩展 IP 访问规则 (rule); 本命令的 no 操作为删除此命名扩展 IP 访问规则(rule)。
<pre>[no] {deny permit} {eigrp gre igmp ipinip ip ospf < protocol-num >} {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} {{<dIpAddr> <dMask>} any-destination {host-destination <dIpAddr>}} [precedence <prec>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条匹配其他特定 IP 协议或所有 IP 协议的数字扩展 IP 访问规则; 如果此编号数字扩展访问列表不存在则创建此访问列表。

c. 退出命名扩展 IP 访问列表配置模式

命令	解释
命名扩展 IP 访问列表配置模式	
exit	退出命名扩展 IP 访问列表配置模式。

(5) 配置数字标准 MAC 访问列表

命令	解释
全局配置模式	

access-list <num> {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} no access-list <num>	创建一条数字标准 mac 访问列表，如果已有此访问列表，则增加一条规则（rule）表项；本命令的 no 操作为删除一条数字标准 mac 访问列表。

(6) 配置数字扩展 MAC 访问列表

命令	解释
全局配置模式	
access-list <num> {deny permit} {any-source-mac {host-source-mac<host_smac>}{<smac><smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>}{<dmac><dmac-mask>}} [{untagged-eth2 tagged-eth2 untagged-802-3 tagged-802-3} [<offset1> <length1> <value1> [<offset2> <length2> <value2> [<offset3> <length3> <value3> [<offset4> <length4> <value4>]]]] no access-list <num>	创建一条数字扩展 mac 访问列表，如果已有此访问列表，则增加一条规则（rule）表项；本命令的 no 操作为删除一条数字扩展 mac 访问列表。

(7) 配置命名扩展 MAC 访问列表**a. 创建一个命名扩展 MAC 访问列表**

命令	解释
全局配置模式	
mac-access-list extended <name> no mac-access-list extended <name>	创建一条命名扩展 MAC 访问列表；本命令的 no 操作为删除此命名扩展 MAC 访问列表。

b. 指定多条 permit 或 deny 规则表项

命令	解释
命名扩展 MAC 访问列表配置模式	

<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} [cos <cos-val> [<cos-bitmask>]] [vlanId <vid-value> [<vid-mask>]] [ethertype <protocol> [<protocol-mask>]]</pre> <pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} [ethertype <protocol> [<protocol-mask>]]</pre> <pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac<host_dmac>} {<dmac><dmac-mask>}} [vlanid <vid-value> [<vid-mask>]] [ethertype <protocol> [<protocol-mask>]]]</pre>	创建一条匹配普通 MAC 帧的命名扩展 MAC 访问规则(rule); no 操作为删除此命名扩展 MAC 访问规则 (rule)
<pre>[no]{deny permit}{any-source-mac {host-source-mac<host _smac>}}{<smac><smac-mask>}}{any-destination-mac {ho st-destination-mac<host_dmac>}}{<dmac><dmac-mask>}} [untagged-eth2 [ethertype <protocol> [protocol-mask]]]</pre>	创建一条匹配 untagged 以太网 2 帧类型的命名扩展 MAC 访问规则(rule); no 操作为删除此命名扩展 MAC 访问规则 (rule)
<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} [untagged-802-3]</pre>	创建一条匹配 untagged 802.3 帧类型的命名扩展 MAC 访问规则(rule); no 操作为删除此命名扩展 MAC 访问规则 (rule)

[no]{deny permit}{any-source-mac[{host-source-mac<host _smac>}]{<smac><smac-mask>}}{any-destination-mac[{host-destination-mac<host _dmac>}]{<dmac><dmac-mask>}}[tagged-eth2 [cos <cos-val> [<cos-bitmask>]] [vlanId <vid-value> [<vid-mask>]] [ethertype<protocol> [<protocol-mask>]]]	创建一条匹配 tagged 以太网 2 帧类型的命名扩展 MAC 访问规则 (rule); no 操作为删除此命名扩展 MAC 访问规则 (rule)
[no] {deny permit} {any-source-mac {host-source-mac <host _smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host _dmac>} {<dmac> <dmac-mask>}} [tagged-802-3 [cos <cos-val> [<cos-bitmask>]] [vlanId <vid-value> [<vid-mask>]]]	创建一条匹配 tagged 802.3 帧类型的命名扩展 MAC 访问规则 (rule); no 操作为删除此命名扩展 MAC 访问规则 (rule)

c. 退出 MAC 访问表配置模式

命令	解释
命名扩展 MAC 访问列表配置模式	
exit	退出命名扩展 MAC 访问列表配置模式

(8) 配置数字扩展 MAC-IP 访问列表

命令	解释
全局配置模式	
access-list <num> {deny permit} {any-source-mac {host-source-mac<host _smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host _dmac>} {<dmac><dmac-mask>}} icmp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [<icmp-type> [<icmp-code>]] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]	创建一条 MAC-ICMP 数字扩展 mac-ip 访问列表, 如果已有此访问列表, 则增加一条规则 (rule) 表项。

<pre>access-list <num> {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} igmp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [<igmp-type>] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 MAC-IGMP 数字扩展 mac-ip 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>access-list <num> {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} tcp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} [s-port {<port1> range <sPortMin> <sPortMax>}] {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [d-port {<port3> range <dPortMin> <dPortMax>}] [ack+fin+psh+rst+urg+syn] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 MAC-TCP 数字扩展 mac-ip 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
<pre>access-list <num> {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac><dmac-mask>}} udp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} [s-port {<port1> range <sPortMin> <sPortMax>}] {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [d-port {<port3> range <dPortMin> <dPortMax>}] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 MAC-UDP 数字扩展 mac-ip 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。

access-list <num> {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} {eigrp gre igmp ip ipinip ospf {<protocol-num>}} {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]	创建一条匹配其他特定 MAC-IP 协议或所有 MAC-IP 协议的数字扩展 mac-ip 访问列表，如果已有此访问列表，则增加一条规则（rule）表项。
no access-list <num>	删除一条数字扩展 MAC-IP 访问列表。

(9) 配置命名扩展 MAC-IP 访问列表

a. 创建一个命名扩展 MAC-IP 访问列表

命令	解释
全局配置模式	
mac-ip-access-list extended <name> no mac-ip-access-list extended <name>	创建一条命名扩展 MAC-IP 访问列表；本命令的 no 操作为删除此命名扩展 MAC-IP 访问列表。

b. 指定多条 permit 或 deny 规则表项

命令	解释
命名扩展 MAC-IP 访问列表配置模式	
[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} icmp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [<icmp-type> [<icmp-code>]] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]	创建一条 mac-icmp 命名扩展 MAC 访问规则（rule）；no 操作为删除此命名扩展 MAC-IP 访问规则（rule）。

<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} igmp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [<igmp-type>] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 mac-igmp 命名扩展 MAC-IP 访问规则 (rule); no 操作为删除此命名扩展 MAC-IP 访问规则 (rule)。
<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} tcp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} [s-port {<port1> range <sPortMin> <sPortMax>}] {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [d-port {<port3> range <dPortMin> <dPortMax>}] [ack+fin+psh+rst+urg+syn] [precedence <precedence>] [tos <tos>][time-range<time-range-name>]</pre>	创建一条 mac-tcp 命名扩展 MAC-IP 访问规则 (rule); no 操作为删除此命名扩展 MAC-IP 访问规则 (rule)。
<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} udp {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} [s-port {<port1> range <sPortMin> <sPortMax>}] {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [d-port {<port3> range <dPortMin> <dPortMax>}] [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 mac-udp 命名扩展 MAC-IP 访问规则 (rule); no 操作为删除此命名扩展 MAC-IP 访问规则 (rule)。

<pre>[no] {deny permit} {any-source-mac {host-source-mac <host_smac>} {<smac> <smac-mask>}} {any-destination-mac {host-destination-mac <host_dmac>} {<dmac> <dmac-mask>}} {eigrp gre igrp ip ipinip ospf {<protocol-num>}} {{<source> <source-wildcard>} any-source {host-source <source-host-ip>}} {{<destination> <destination-wildcard>} any-destination {host-destination <destination-host-ip>}} [precedence <precedence>] [tos <tos>] [time-range<time-range-name>]</pre>	创建一条 mac-ip 其他协议类型的命名扩展 MAC-IP 访问规则 (rule); no 操作为删除此命名扩展 MAC-IP 访问规则 (rule)。

c. 退出 MAC-IP 访问表配置模式

命令	解释
命名扩展 MAC-IP 访问列表配置模式	
exit	退出命名扩展 MAC-IP 访问列表配置模式

(10) 配置数字标准 IPv6 访问列表

命令	解释
全局配置模式	
<pre>ipv6 access-list <num> {deny permit} {{<sIPv6Addr> <sPrefixlen>} any-source {host-source <sIPv6Addr>}} no ipv6 access-list <num></pre>	创建一条数字标准 IPv6 访问列表，如果已有此访问列表，则增加一条规则 (rule) 表项；本命令的 no 操作为删除一条数字标准 IPv6 访问列表。

(11) 配置数字扩展 IPv6 访问列表

命令	解释
全局配置模式	

<pre> ipv6 access-list <num-ext> {deny permit} icmp {{<sIPv6Prefix/sPrefixlen>} any-source {host-source <sIPv6Addr>}} {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [<icmp-type> [<icmp-code>]] [dscp <dscp>] [flow-label <fl>][time-range<time-range-name>] ipv6 access-list <num-ext> {deny permit} tcp {{<sIPv6Prefix/<sPrefixlen>} any-source {host-source <sIPv6Addr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {{< dIPv6Prefix/<dPrefixlen>} any-destination {host-destination <dIPv6Addr>}} [dPort {<dPort> range <dPortMin> <dPortMax>}] [syn ack urg rst fin psh] [dscp <dscp>] [flow-label <flowlabel>][time-range<time-range-name>] ipv6 access-list <num-ext> {deny permit} udp {{<sIPv6Prefix/<sPrefixlen>} any-source {host-source <sIPv6Addr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {{<dIPv6Prefix/<dPrefixlen>} any-destination {host-destination <dIPv6Addr>}} [dPort {<dPort> range <dPortMin> <dPortMax>}] [dscp <dscp>] [flow-label <flowlabel>][time-range<time-range-name>] ipv6 access-list <num-ext> {deny permit} <next-header> {<sIPv6Prefix/sPrefixlen> any-source {host-source <sIPv6Addr>}} {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [dscp <dscp>] [flow-label <fl>][time-range<time-range-name>] no ipv6 access-list <num> </pre>	创建一条数字扩展 IPV6 访问列表，如果已有此访问列表，则增加一条规则（rule）表项；本命令的 no 操作为删除一条数字标准 IP 访问列表。

(12) 配置命名标准 IPv6 访问列表

a) 创建一个命名标准 IPv6 访问列表

命令	解释
全局配置模式	

ipv6 access-list standard <name> no ipv6 access-list standard <name>	创建一条命名标准 IPv6 访问列表；本命令的 no 操作为删除此命名标准 IPv6 访问列表。

b) 指定多条 permit 或 deny 规则

命令	解释
命名标准 IPv6 访问列表配置模式	
[no] {deny permit} {{<slIPv6Prefix/sPrefixlen>} any-source {host-source <slIPv6Addr>}}	创建一条命名标准 IPv6 访问规则（rule）；本命令的 no 操作为删除此命名标准 IPv6 访问规则（rule）。

c) 退出命名标准 IPv6 访问列表配置模式

命令	解释
命名标准 IPv6 访问列表配置模式	
exit	退出命名标准 IPv6 访问列表配置模式。

(13) 配置命名扩展 IPv6 访问列表

a) 创建一个命名扩展 IPv6 访问列表

命令	解释
全局配置模式	
ipv6 access-list extended <name> no ipv6 access-list extended <name>	创建一条命名扩展 IPv6 访问列表；本命令的 no 操作为删除此命名扩展 IPv6 访问列表。

b) 指定多条 permit 或 deny 规则

命令	解释
命名扩展 IP 访问列表配置模式	
[no] {deny permit} icmp {{<slIPv6Prefix/sPrefixlen>} any-source {host-source <slIPv6Addr>}} {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [<icmp-type> [<icmp-code>]] [dscp <dscp>] [flow-label <fl>][time-range<time-range-name>]	创建一条 icmp 命名扩展 IPv6 访问规则（rule）；本命令的 no 操作为删除此命名扩展 IPv6 访问规则（rule）。

[no] {deny permit} tcp {<slIPv6Prefix/sPrefixlen> any-source {host-source <slIPv6Addr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [syn ack urg rst fin psh] [dscp <dscp>] [flow-label <fl>] [time-range<time-range-name>]	创建一条 tcp 命名扩展 IPv6 访问规则 (rule); 本命令的 no 操作为删除此命名扩展 IPv6 访问规则 (rule)。
[no] {deny permit} udp {<slIPv6Prefix/sPrefixlen> any-source {host-source <slIPv6Addr>}} [s-port {<sPort> range <sPortMin> <sPortMax>}] {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [d-port {<dPort> range <dPortMin> <dPortMax>}] [dscp <dscp>] [flow-label <fl>] [time-range<time-range-name>]	创建一条 udp 命名扩展 IPv6 访问规则 (rule); 本命令的 no 操作为删除此命名扩展 IPv6 访问规则 (rule)。
[no] {deny permit} <proto> {<slIPv6Prefix/sPrefixlen> any-source {host-source <slIPv6Addr>}} {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [dscp <dscp>] [flow-label <fl>] [time-range<time-range-name>]	创建一条匹配其他特定 IPv6 协议的数字扩展 IPv6 访问规则; 如果此编号数字扩展访问列表不存在则创建此访问列表。
[no] {deny permit} {<slIPv6Prefix/sPrefixlen> any-source {host-source <slIPv6Addr>}} {<dIPv6Prefix/dPrefixlen> any-destination {host-destination <dIPv6Addr>}} [dscp <dscp>] [flow-label <fl>] [time-range<time-range-name>]	创建一条命名扩展 IPv6 访问规则 (rule); 本命令的 no 操作为删除此命名扩展 IPv6 访问规则 (rule)。

c) 退出命名扩展 IPv6 访问列表配置模式

命令	解释
命名扩展 IPv6 访问列表配置模式	
exit	退出命名扩展 IPv6 访问列表配置模式。

2. 配置包过滤功能

(1) 全局打开包过滤功能

命令	解释
全局配置模式	
firewall enable	全局打开包过滤功能。

firewall disable	全局关闭包过滤功能。

(2) 配置 ACL deny 优先功能

命令	解释
全局配置模式	
[no] access-list deny-preemption	开启 acl deny 优先功能, 本命令的 no 操作为关闭 acl deny 优先功能。

3. 配置时间范围功能

(1) 创建时间范围名称

命令	解释
全局配置模式	
time-range <time_range_name>	创建一个名为 time_range_name 的时间范围名
no time-range <time_range_name>	停止名为 time_range_name 的时间范围功能

(2) 配置周期性时段

命令	解释
时间范围模式	
absolute-periodic {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <start_time> to {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <end_time>	配置一周内的各种不同要求的时间范围, 每周都循环这个时间
periodic {Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+Sunday} daily weekdays weekend} <start_time> to <end_time>	

[no] absolute-periodic {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <start_time> to {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <end_time>	停止一周内的时间范围配置
[no] periodic {{Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+Sunday} daily weekdays weekend} <start_time> to <end_time>	

(3) 配置绝对性时段

命令	解释
全局配置模式	
absolute start <start_time> <start_data> [end <end_time> <end_data>]	创建一个绝对时间范围
[no] absolute start <start_time> <start_data> [end <end_time> <end_data>]	停止一个绝对时间范围功能

4. 将 access-list 绑定到特定端口的特定方向

命令	解释
物理接口配置模式/Vlan 接口模式	
{ip ipv6 mac mac-ip} access-group <acl-name> {in out} [traffic-statistic] no {ip ipv6 mac mac-ip} access-group <acl-name> {in out}	物理接口模式：在端口的入口或出口方向上应用一条 access-list；no 操作为删除绑定在端口上的 access-list。

5. 显示端口 ACL 的配置及统计信息

命令	解释
特权和配置模式	

show access-group in (interface {Ethernet Ethernet IFNAME})	显示端口上 ACL 绑定情况。

6. 清空指定接口的包过滤统计信息

命令	解释
特权用户模式	
clear access-group (in out) statistic interface { <interface-name> ethernet <interface-name> }	在指定端口清除出口或入口方向包过滤统计信息。

6.1.3 ACL 举例

案例 1:

用户有如下配置需求：交换机的端口 10 连接 10.0.0.0/24 网段，管理员不希望用户使用 ftp。

配置更改:

- a) 创建相应的 ACL
- b) 配置包过滤功能
- c) 绑定 ACL 到端口

配置步骤如下:

```
Switch(config)#access-list 110 deny tcp 10.0.0.0 0.0.0.255 any-destination d-port 21
```

```
Switch(config)#firewall enable
```

```
Switch(config)#interface ethernet1/10
```

```
Switch(Config-If-Ethernet1/10)#ip access-group 110 in
```

```
Switch(Config-If-Ethernet1/10)#exit
```

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

配置结果:

```
Switch#show firewall
```

```
Firewall Status: Enable.
```

```
Switch#show access-lists
```

```
access-list 110(used 1 time(s)) 1 rule(s)
```

```
access-list 110 deny tcp 10.0.0.0 0.0.0.255 any-destination d-port 21
```

```
Switch#show access-group interface ethernet 1/10
```

```
interface name:Ethernet1/10
```

```
IP Ingress access-list used is 110, traffic-statistics Disable.
```

案例 2:

用户有如下配置需求:交换机的端口 10 不允许转发源 MAC 地址是 00-12-11-23-XX-XX 的 802.3 的数据报文。

配置更改:

- 1、创建相应的 MAC ACL
- 2、配置包过滤功能
- 3、绑定 ACL 到端口

配置步骤如下:

```
Switch(config)#access-list 1100 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac untagged-802-3
Switch(config)# access-list 1100 deny 00-12-11-23-00-00 00-00-00-00-ff-ff any
tagged-802
Switch(config)#firewall enable
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#mac access-group 1100 in
Switch(Config-If-Ethernet1/10)#exit
Switch(config)#exit
```

配置结果:

```
Switch#show firewall
Firewall Status: Enable.
Switch #show access-lists
access-list 1100(used 1 time(s))
access-list 1100 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac
untagged-802-3
access-list 1100 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac
Switch #show access-group interface ethernet 1/10
interface name:Ethernet1/10
MAC Ingress access-list used is 1100,traffic-statistics Disable.
```

案例 3:

用户有如下配置需求:交换机的端口 10 连接的网段的 MAC 地址是 00-12-11-23-XX-XX, 并且 IP 为 10.0.0.0/24 网段, 管理员不希望用户使用 ftp, 也不允许外网 ping 此网段的任何一台主机。

配置更改:

- 创建相应的 ACL

- 配置包过滤功能
- 绑定 ACL 到端口

配置步骤如下：

```
Switch(config)#access-list 3110 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac tcp 10.0.0.0 0.0.0.255 any-destination d-port 21
Switch(config)#access-list 3110 deny any-source-mac 00-12-11-23-00-00
00-00-00-00-ff-ff icmp any-source 10.0.0.0 0.0.0.255
```

```
Switch(config)#firewall enable
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#mac-ip access-group 3110 in
Switch(Config-Ethernet1/10)#exit
Switch(config)#exit
```

配置结果：

```
Switch#show firewall
Firewall Status: Enable.
```

```
Switch#show access-lists
access-list 3110(used 1 time(s))
access-list 3110 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac
tcp 10.0.0.0 0.0.0.255 any-destination d-port 21
access-list 3110 deny any-source-mac 00-12-11-23-00-00 00-00-00-00-ff-ff icmp
any-source 10.0.0.0 0.0.0.255
```

```
Switch#show access-group interface ethernet 1/10
interface name: Ethernet1/10
MAC-IP Ingress access-list used is 3110, traffic-statistics Disable.
```

案例 4：

用户有如下配置需求：交换机的 600 端口连接的网段是 IPV6，并且 IPV6 的地址为 2003:1:1:1::0/64 网段，管理员不希望除了 2003:1:1:1:66::0/80 网段用户访问外网。

配置更改：

- 1、 创建相应的 ACL
- 2、 配置包过滤功能
- 3、 绑定 ACL 到端口

配置步骤如下：

```
Switch(config)#ipv6 access-list 600 permit 2003:1:1:1:66::0/80 any-destination
Switch(config)#ipv6 access-list 600 deny 2003:1:1:1::0/64 any-destination
```

```
Switch(config)#firewall enable
```

```
Switch(config)#interface ethernet 1/10
```

```
Switch(Config-If-Ethernet1/10)#ipv6 access-group 600 in
```

```
Switch(Config-If-Ethernet1/10)#exit
```

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

配置结果:

```
Switch#show firewall
```

```
Firewall Status: Enable.
```

```
Switch#show ipv6 access-lists
```

```
Ipv6 access-list 600(used 1 time(s))
```

```
ipv6 access-list 600 deny 2003:1:1:1::0/64 any-source
```

```
ipv6 access-list 600 permit 2003:1:1:1:66::0/80 any-source
```

```
Switch #show access-group interface ethernet 1/10
```

```
interface name:Ethernet1/10
```

```
IPv6 Ingress access-list used is 600, traffic-statistics Disable.
```

案例 5:

用户有以下配置需求: 交换机的端口 1, 2, 5, 7 属于 Vlan100, 管理员不希望 IP 地址为 192.168.0.1 的设备接入到这几个端口。

配置更改:

1. 创建相应的 ACL
2. 配置包过滤功能
3. 绑定 ACL 到端口

配置步骤如下:

```
Switch (config)#firewall enable
```

```
Switch (config)#vlan 100
```

```
Switch (Config-Vlan100)#switchport interface ethernet 1/1;2;5;7
```

```
Switch (Config-Vlan100)#exit
```

```
Switch (config)#access-list 1 deny host-source 192.168.0.1
```

```
Switch (config)#interface ethernet1/1;2;5;7
```

```
Switch (config-if-port-range)#ip access-group 1 in
```

```
Switch (Config-if-Vlan100)#exit
```

配置结果:

```
Switch (config)#show access-group interface vlan 100
```

Interface VLAN 100:

Ethernet1/1: IP Ingress access-list used is 1, traffic-statistics Disable.

Ethernet1/2: IP Ingress access-list used is 1, traffic-statistics Disable.

Ethernet1/5: IP Ingress access-list used is 1, traffic-statistics Disable.

Ethernet1/7: IP Ingress access-list used is 1, traffic-statistics Disable.

6.1.4 ACL 排错帮助

6. 对 ACL 中的表项的检查是自上而下的，只要匹配一条表项，对此 ACL 的检查就马上结束。
- ✎ 端口特定方向上没有绑定 ACL 或没有任何 ACL 表项匹配时，才会使用默认规则。每个端口入口可以绑定一条 MAC-IP ACL，一条 IP ACL，一条 MAC ACL 和一条 IPV6 标准 ACL（通过物理接口配置模式或 Vlan 接口配置模式）。
- ✎ 当同时绑定四条 ACL 且数据包同时匹配其中多条 ACL 时，优先关系从高到低如下，如果优先级相同，则先配置的优先级高：
 - ◆ 入口 IPv6 ACL
 - ◆ 入口 MAC-IP ACL
 - ◆ 入口 IP ACL
 - ◆ 入口 MAC ACL
- ✎ 端口可以成功绑定的 ACL 数目取决于已绑定的 ACL 的内容以及硬件资源限制。
- ✎ 如果 access-list 中包括过滤信息相同但动作矛盾的规则，则其无法绑定到端口并将有报错提示。例如同时配置 permit tcp any-source any-destination 及 deny tcp any-source any-destination。
- ✎ 可以配置 ACL 拒绝某些 ICMP 报文通过以防止“冲击波”等病毒攻击。
- ✎ 如果物理接口的模式为 Trunk，则该端口只能通过物理接口配置模式配置 ACL。
- ✎ 在物理接口配置模式下绑定的 ACL 只能在物理接口配置模式下取消，在 Vlan 接口配置模式下绑定的 ACL 只能在 Vlan 接口模式下取消。
- ✎ 物理接口加入/移出 Vlan 时（Trunk 口除外），该 Vlan 中配置的 ACL 将自动绑定/移除该物理接口；如果目的 Vlan 中配置的 ACL（通过 Vlan 接口配置模式配置）和该端口原有的 ACL 配置（通过物理接口配置模式配置）冲突，将导致端口移动失败。
- ✎ Vlan 中无物理接口时（Trunk 口除外），将删除 ACL 在 Vlan 中的配置，以后如果再有物理端口移入 Vlan，将无 ACL 应用于该物理接口。端口模式转换：access->trunk，将取消通过 Vlan 接口配置模式绑定到该物理接口的 ACL；trunk->access,将 Vlan 1 接口配置的 ACL 绑定到该物理接口，如果绑定失败，端口模式转换将执行失败。

6.2 802.1x

6.2.1 802.1x 介绍

802.1x协议起源于802.11协议，后者是IEEE的无线局域网协议，制订802.1x协议的初衷是为了解决无线局域网用户的接入认证问题。IEEE 802 LAN协议定义的局域网并不提供接入认证，只要用户能接入局域网控制设备（如LAN Switch），就可以访问局域网中的设备或资源。这在早期企业网有线LAN应用环境下并不存在明显的安全隐患。

随着移动办公及驻地网运营等应用的大规模发展，服务提供者需要对用户的接入进行控制和配置。尤其是WLAN的应用和LAN接入在电信网上大规模开展，有必要对端口加以控制以实现用户级的接入控制，802.1x就是IEEE LAN/WAN 委员会为了解决基于端口的网络接入控制（Port-Based Network Access Control）而定义的一个标准，该标准目前已经在无线局域网和以太网中被广泛应用。

“基于端口的网络接入控制”是指在局域网接入设备的端口这一级对所接入的用户设备进行认证和控制。连接在端口上的用户设备如果能通过认证，就可以访问局域网中的资源；如果不能通过认证，则无法访问局域网中的资源。

6.2.1.1 802.1x 认证体系结构

使用802.1x的系统为典型的Client/Server体系结构，包括三个实体，如下图所示，分别为：Supplicant system（客户端）、Authenticator system（接入控制单元）以及Authentication server system（认证服务器）。

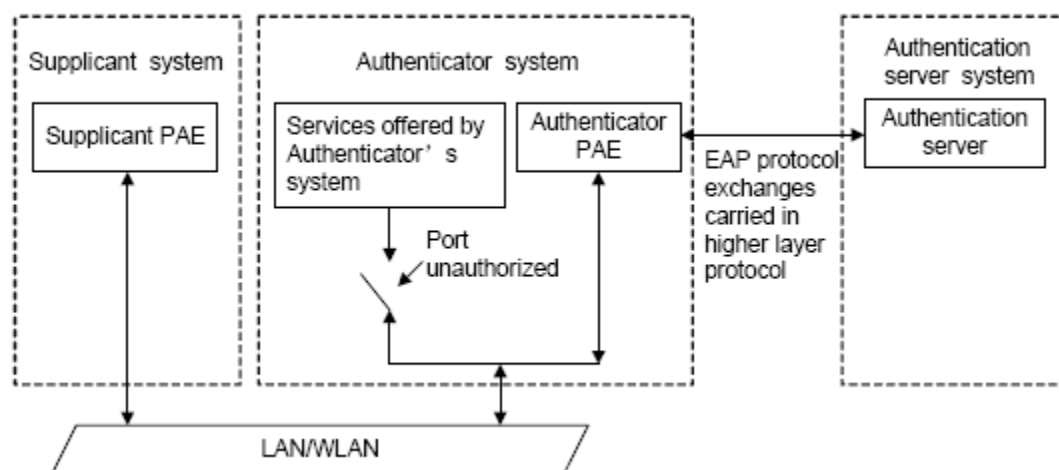


图 6-1 802.1x 认证体系结构

- 客户端是位于局域网段一端的一个实体，由该链路另一端的接入控制单元对其进行认证。客户端一般为一个用户终端设备，用户通过启动客户端软件发起802.1x认证。客户端必须支持EAPOL（Extensible Authentication Protocol over LAN，局域网上的可扩展认证协议）。

- ✎ 接入控制单元是位于局域网段一端的另一个实体，对所连接的客户端进行认证。接入控制单元通常为支持802.1x 协议的网络设备，它为客户端提供接入局域网的端口，该端口可以是物理端口，也可以是逻辑端口。
- ✎ 认证服务器是为接入控制单元提供认证服务的实体。认证服务器用于实现对用户进行认证、授权和计费，通常为RADIUS(Remote Authentication Dial-In User Service, 远程认证拨号用户服务)服务器。该服务器可以存储有关用户的信息。包括用户名、密码以及其它参数，例如用户所属的VLAN、端口等。

三个实体涉及以下三个基本概念：端口 PAE、受控端口和受控方向。

1. PAE

PAE (Port Access Entity, 端口访问实体) 是 802.1x 认证机制中负责执行算法和协议操作的实体。

- ✎ 客户端PAE负责响应接入控制单元的认证请求，向接入控制单元提交用户的认证信息。客户端PAE也可以主动向接入控制单元发送认证请求和下线请求。
- ✎ 接入控制单元PAE利用认证服务器对需要接入局域网的客户端执行认证，并根据认证结果对受控端口的授权/非授权状态进行相应地控制。授权状态是指允许用户访问网络资源；非授权状态是指仅允许EAPOL报文收发，不允许用户访问网络资源。

2. 受控/非受控端口

接入控制单元为客户端提供接入局域网的端口，这个端口被划分为两个逻辑端口：受控端口和非受控端口。

- ✎ 非受控端口始终处于双向连通状态，主要用来传递EAPOL协议帧，保证客户端始终能够发出或接收认证报文。
- ✎ 受控端口在授权状态下处于双向连通状态，用于传递业务报文；在非授权状态下禁止从客户端接收任何报文。
- ✎ 受控端口和非受控端口是同一端口的两个部分；任何到达该端口的帧，在受控端口与非受控端口上均可见。

3. 受控方向

在非授权状态下，受控端口可以被设置成单向受控或双向受控。

- ✎ 实行双向受控时，禁止帧的发送和接收。
- ✎ 实行单向受控时，禁止从客户端接收帧，但允许向客户端发送帧。

说明：目前，本交换机只支持单向受控。

6.2.1.2 802.1x 工作机制

IEEE 802.1x 认证系统使用 EAP (Extensible Authentication Protocol, 可扩展认证协议)，来实现客户端、接入控制单元和认证服务器之间认证信息的交换。

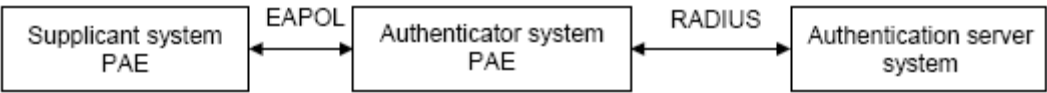


图 6-2 802.1x 认证工作机制

- 在客户端PAE与接入控制单元PAE之间，EAP协议报文使用EAPOL封装格式，直接承载于LAN环境中。
- 在接入控制单元PAE与RADIUS服务器之间，可以使用两种方式来交换信息。一种是EAP协议报文使用EAPOR（EAP over RADIUS）封装格式承载于RADIUS协议中；另一种是EAP协议报文由接入控制单元PAE进行终结，采用包含PAP（Password Authentication Protocol，密码验证协议）或CHAP（Challenge Handshake Authentication Protocol，质询握手验证协议）属性的报文与RADIUS服务器进行认证交互。
- 当用户完成认证后，认证服务器会把用户的相关信息传递给接入控制单元，接入控制单元PAE根据RADIUS服务器的认证结果（Accept 或Reject）决定受控端口的授权/非授权状态。

6.2.1.3 EAPOL 消息的封装

1. EAPOL 数据包的格式

EAPOL 是 802.1x 协议定义的一种报文封装格式，主要用于在客户端和接入控制单元之间传送 EAP 协议报文，以允许 EAP 协议报文在 LAN 上传送。在 IEEE 802/Ethernet LAN 环境中，EAPOL 数据包的格式如下图所示，EAPOL 数据包的起始位置是 MAC 帧的 Type/Length 域。

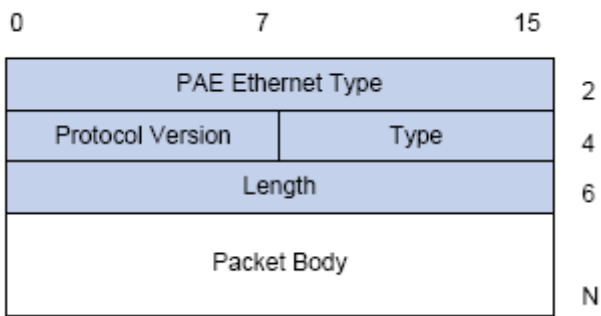


图 6-3 EAPOL 数据包格式

- PAE Ethernet Type: 表示协议类型，为 0x888E。
- Protocol Version: 表示 EAPOL 数据包的发送方所支持的协议版本号。
- Type: 表示 EAPOL 数据包类型，具体类型包括：
- EAP-Packet（值为0x00）：认证信息帧，用于承载EAP报文。该帧能够穿越（Pass through）接入控制单元，在客户端和认证服务器之间传递EAP报文。
 - EAPOL-Start（值为0x01）：认证发起帧。
 - EAPOL-Logoff（值为0x02）：退出请求帧。
 - EAPOL-Key（值为0x03）：密钥信息帧。
 - EAPOL-Encapsulated-ASF-Alert（值为0x04）：用于支持ASF（Alert Standard Forum）的Alerting 消息。该帧用于封装与网管相关信息，例如各种警告信息，由

设备端终结。

Length: 表示数据长度，也就是“Packet Body”字段的长度，单位为字节。如果为 0，则表示没有后面的数据域。

Packet Body: 表示数据内容，根据不同的 Type 有不同的格式。

2. EAP 数据包的格式

当 EAPOL 数据包格式 Type 域为 EAP-Packet 时，Packet Body 为 EAP 数据包结构，如下图所示。

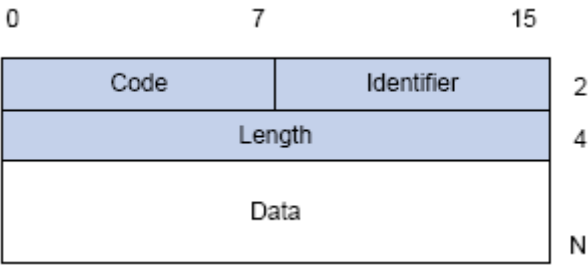


图 6-4 EAP 数据包格式

Code: 指明 EAP 数据包的类型，共有 4 种：Request (1)、Response (2)、Success (3)、Failure (4)。

- ☞ Success 和 Failure 类型的包没有 Data 域，相应的 Length 域的值 4。
- ☞ Request 和 Response 类型数据包的 Data 域的格式如下图所示。Type 为 EAP 的认证类型，Type data 的内容由类型决定。例如，Type 值为 1 时代表 Identity，用来查询对方的身份；Type 值为 4 时，代表 MD5-Challenge，类似于 PPP CHAP 协议，包含质询消息。

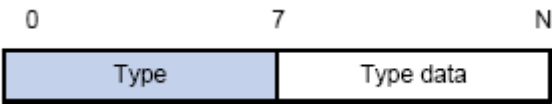


图 6-5 Request 和 Response 类型数据包的 Data 域的格式

Identifier: 辅助进行 Request 和 Response 消息的匹配。

Length: EAP 数据包的长度，包含 Code、Identifier、Length 和 Data 域，单位为字节。

Data: EAP 数据包的内容，由 Code 类型决定。

6.2.1.4 EAP 属性的封装

RADIUS 为支持 EAP 认证增加了两个属性：EAP-Message（EAP 消息）和 Message-Authenticator（消息认证码）。RADIUS 协议的报文格式请参见“AAA-RADIUS-HWTACACS 操作”的 RADIUS 协议简介部分。

1. EAP-Message

如图所示，这个属性用来封装 EAP 数据包，类型代码为 79，String 域最长 253 字节，如果 EAP 数据包长度大于 253 字节，可以对其进行分片，依次封装在多个 EAP-Message 属性中。

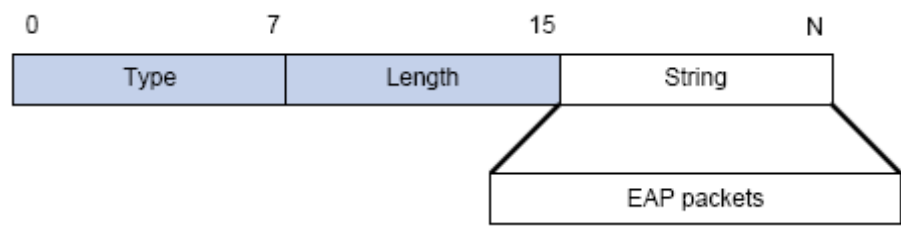


图 6-6 EAP-Message 属性封装

2. Message-Authenticator

如图所示，这个属性用于在使用 EAP、CHAP 等认证方法的过程中，避免接入请求包被窃听。在含有 EAP-Message 属性的数据包中，必须同时也包含 Message-Authenticator，否则该数据包会被认为无效而被丢弃。

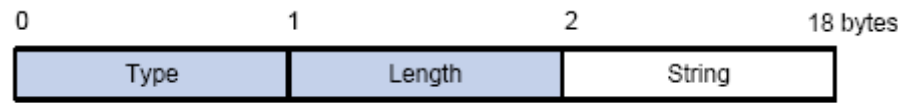


图 6-7 Message-Authenticator 属性

6.2.1.5 基于 802.1x 的 Web 认证代理功能

原有神州数码 802.1x 认证系统的整体思路采用 IEEE 802.1x 认证系统的体系结构、工作机制、基本业务流程，客户端使用神州数码专有客户端进行认证，设备端采用神州数码二层交换机，认证服务器使用 RADIUS 服务器，认证消息格式采用 EAP 协议。在客户端与认证代理交换机之间使用 EAPOL 封装技术，即将 EAP 报文封装在以太网帧之中进行认证通信；而认证代理交换机与认证服务器间使用 EAPOR 封装格式，即将 EAP 报文承载在 Radius 协议上进行认证通信；也可以由设备端进行中介，而在设备端与 RADIUS 服务器之间通过 RADIUS 协议传送 PAP 协议报文或 CHAP 协议报文。

神州数码 802.1x 认证系统中，用户为了实现身份认证和网络准入，就必须安装神州数码认证客户端软件，通过认证客户端登录触发认证流程，实现与 DCBI 服务器的认证通信。但是现在有些客户工作不希望安装客户端软件，希望可以简单的通过浏览器就实现认证。为了迎合用户新需求，实现认证客户端的平台无关性，我们设计提出了基于原有 802.1x 系统的 Web 认证功能。

神州数码 Web 认证基本仍采用标准的 802.1x 认证系统，使用浏览器中运行的 Java Applet 替代原有的客户端软件，设备端采用神州数码三层交换机，认证服务器使用标准 RADIUS 服务器，认证消息承载在 EAP 报文中进行通信。由于客户端采用 Java Applet 技术，不能发送以太网帧，故而客户端与 web 认证代理交换机的认证通信 EAP 报文不能采用 EAPOL 封装技术，即不能将 EAP 报文封装在以太网帧中发送。因此作为替代采用 EAPOU 技术，即将 EAP 认证报文承载于 UDP 协议之上，实现 web 客户端与 web 认证代理交换机之间的认证通信，而在认证代理交换机与认证服务器之间仍然采用标准的 EAPOR 协议。

6.2.1.6 802.1x 认证方式

认证过程可以由客户端主动发起，也可以由设备端发起。一方面当设备端探测到有未经过认证的用户使用网络时，就会主动向客户端发送 EAP-Request/Identity 报文，发起认证；另一方面客户端可以通过客户端软件向设备端发送 EAPOL-Start 报文，发起认证。

802.1x 系统支持 EAP 中继方式和 EAP 终结方式与远端 RADIUS 服务器交互完成认证。以下对这两种认证方式的过程描述，都以客户端主动发起认证为例。

6.2.1.6.1 EAP 中继方式

这种方式是 IEEE 802.1x 标准规定的，将 EAP（扩展认证协议）承载在其它高层协议中，如 EAP over RADIUS，以便扩展认证协议报文穿越复杂的网络到达认证服务器。一般来说，EAP 中继方式需要 RADIUS 服务器支持 EAP 属性：EAP-Message 和 Message-Authenticator。

EAP 是一个通用的用来传输实际认证协议的认证框架，而不是一个特殊的认证机制。EAP 提供一些公共的功能，并且允许协商所希望的认证机制，这些机制被叫做 EAP 方法（Method）。EAP 的好处就是当一个新的认证协议发展出来的时候，基础的 EAP 机制不需要随着改变，在下图中给出了 EAP 认证方法协议栈。

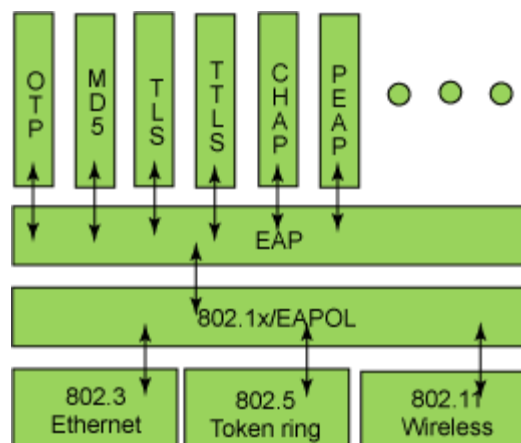


图 6-8 EAP 认证方法协议栈

现在已经开发出 50 种以上的 EAP 认证方法，而各种不同 EAP 认证方法间的差异在于认证机制与密钥管理的不同。其中比较常见的四种 EAP 认证方法包括：

- ☞ **EAP-MD5**
- ☞ **EAP-TLS**（Transport Layer Security，传输层安全）
- ☞ **EAP-TTLS**（Tunneled Transport Layer Security，隧道传输层安全）
- ☞ **PEAP**（Protected Extensible Authentication Protocol，受保护的扩展认证协议）

下面分别介绍这四种 EAP 认证方法。

注意：

- ☞ 交换机作为穿越（Pass-through）接入控制单元，不检查具体 EAP 方法相关的内容，因而可以支持以上常见的 EAP 方法，并能够支持未来扩展的各种 EAP 认证方法。

- ☞ EAP 中继方式下，如果要采用 EAP-MD5、EAP-TLS、EAP-TTLS 或者 PEAP 这四种认证方法之一，需要保证在客户端和 RADIUS 服务器上选择一致的认证方法。

1. EAP-MD5 认证方法

EAP-MD5 是一个 IETF 开放标准，但提供最少的安全，MD5 Hash 函数容易受到字典攻击。

下面给出采用 EAP-MD5 认证方法的基本业务流程，如图所示。

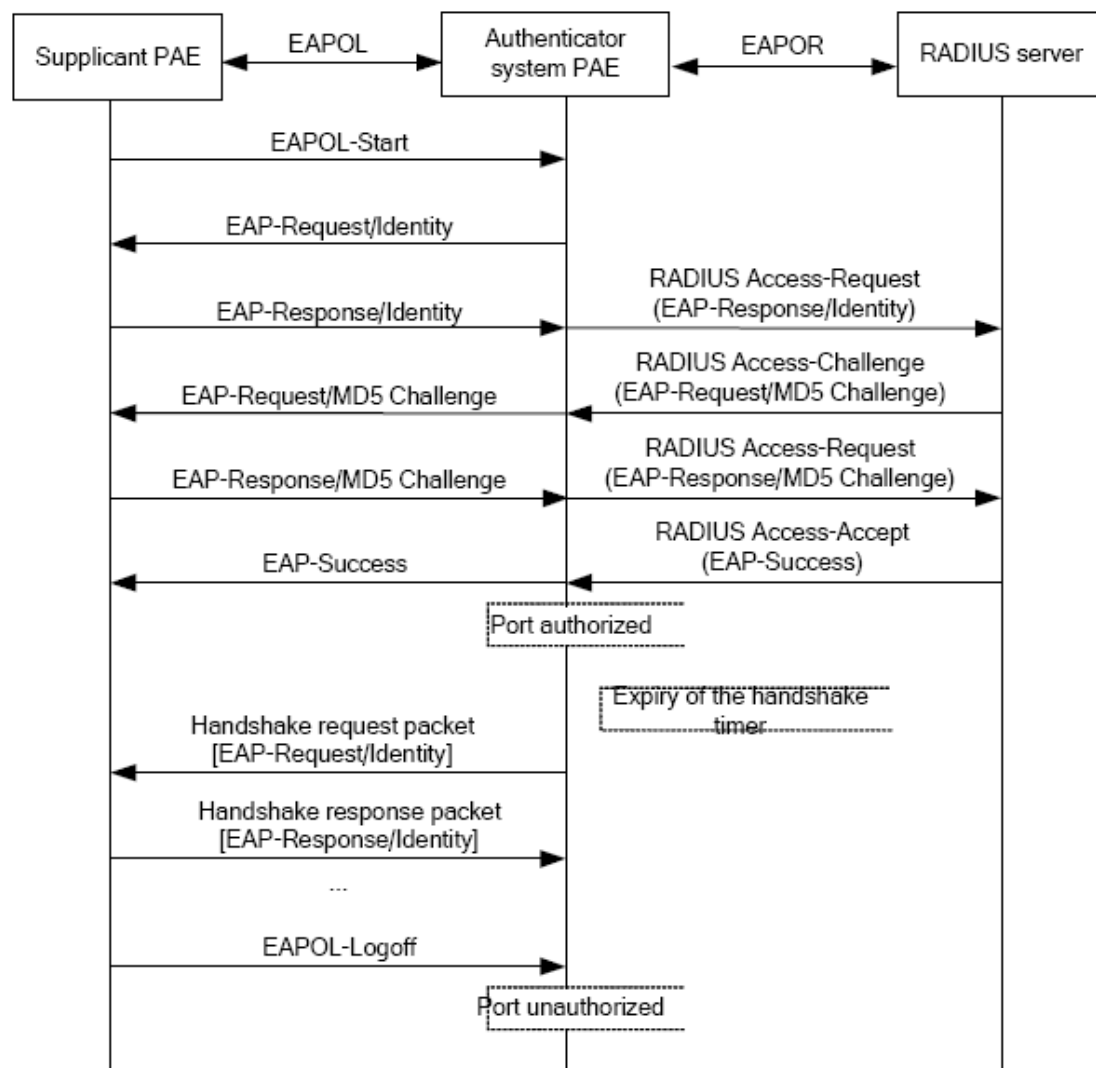


图 6-9 802.1x EAP-MD5 认证流程

2. EAP-TLS 认证方法

EAP-TLS 是由微软公司基于 EAP 和 TLS 协议而提出来的。它使用 PKI 来保护客户端与 Radius 认证服务器之间的身份认证以及生成动态的会话密钥，要求客户端和 Radius 认证服务器都要拥有数字证书以进行双向认证。它是无线局域网最早采纳的 EAP 认证方法。由于需要每一个用户都要拥有数字证书，因而在实际部署时由于维护困难而很少被使用，但它仍被认为是最安全的 EAP 标准之一，而且广泛地被无线局域网硬件和软件制造厂商所支

持。

下面给出采用 EAP-TLS 认证方法的基本业务流程，如图所示。

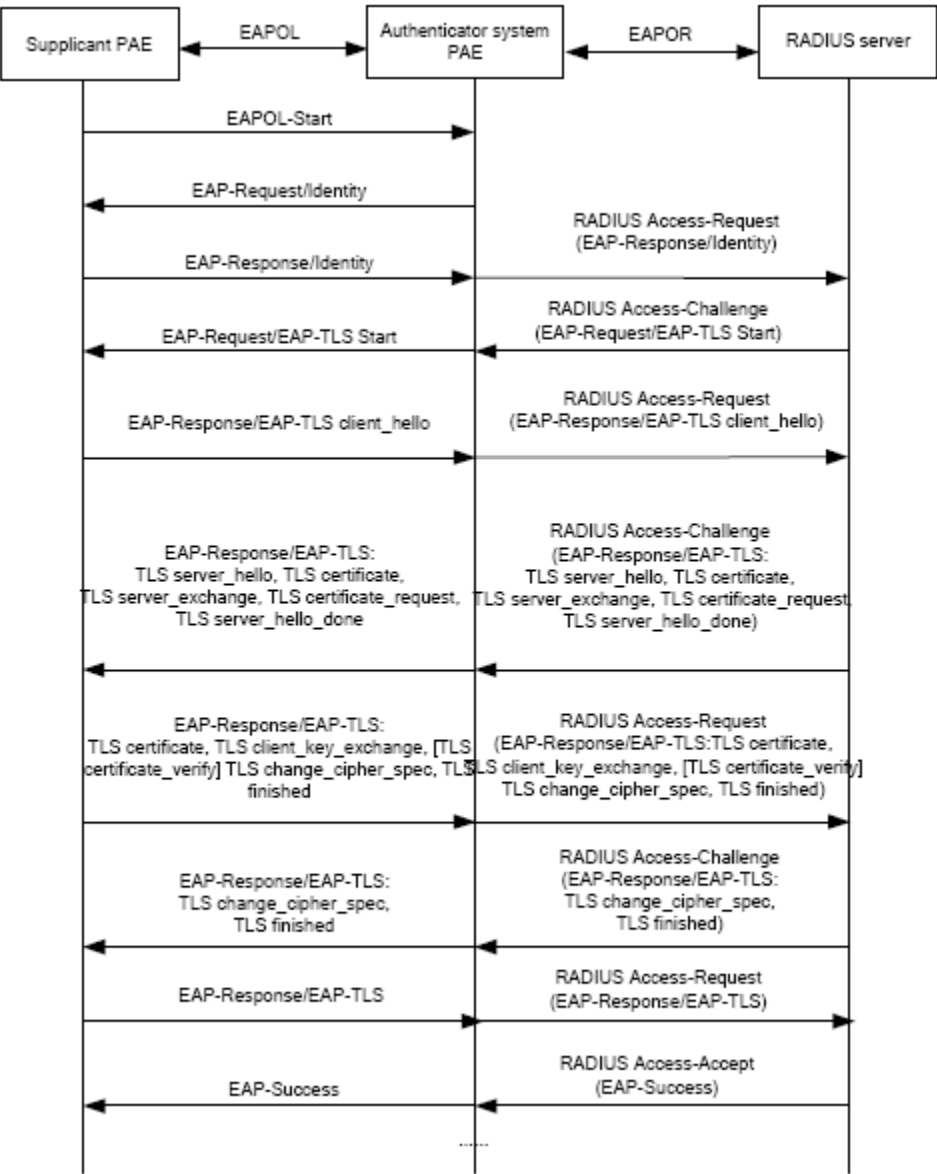


图 6-10 802.1x EAP-TLS 认证流程

3. EAP-TTLS 认证方法

EAP-TTLS 是由 Funk Software 和 Certicom 合作开发的。它可以提供 EAP-TLS 相同的认证强度，但不要求每个用户都要拥有数字证书，而只要求 Radius 认证服务器要拥有数字证书。用户身份的验证是通过密码来进行的，该密码是在利用认证服务器的证书所建立的一个安全的加密的隧道中传输。在 TTLS 隧道内部可以转发任意类型的认证请求，例如 EAP、PAP、MS-CHAPV2 等。

4. PEAP 认证方式

EAP-PEAP 是由 Cisco、微软和 RSA Security 联合提出的开放标准的建议。它早已被运用在产品中，而且提供很好的安全。它在协议设计和安全性方面与 EAP-TTLS 相似，只需要一份服务器端的 PKI 证书来建立一个安全的 TLS 隧道以保护用户认证。

下面给出采用 PEAP 认证方法的基本业务流程，如图所示。

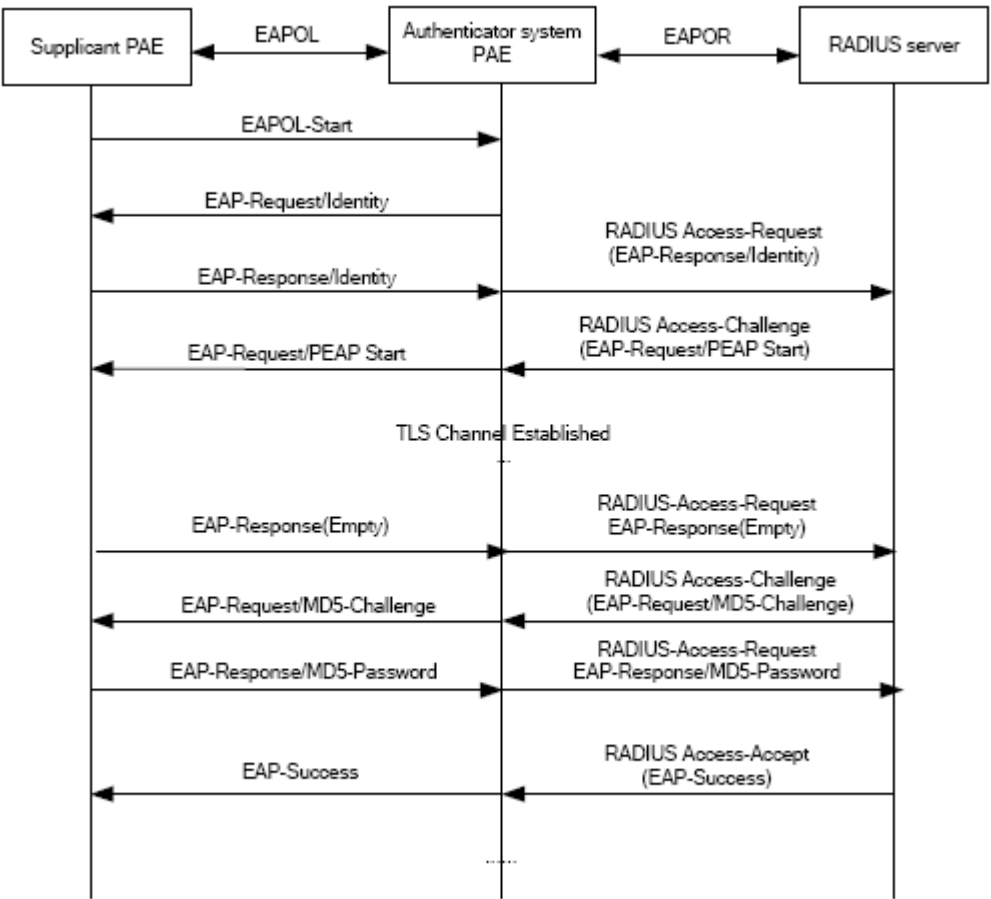


图 6-11 802.1x PEAP 认证流程

6.2.1.6.2 EAP 终结方式

这种方式将 EAP 报文在接入控制单元终结并映射到 RADIUS 报文中，利用标准 RADIUS 协议完成认证、授权和计费。基本业务流程如下图所示。

对于 EAP 终结方式，接入控制单元与 RADIUS 服务器之间可以采用 PAP 或者 CHAP 认证方法。以下以 CHAP 认证方法为例介绍基本业务流程，如图所示。

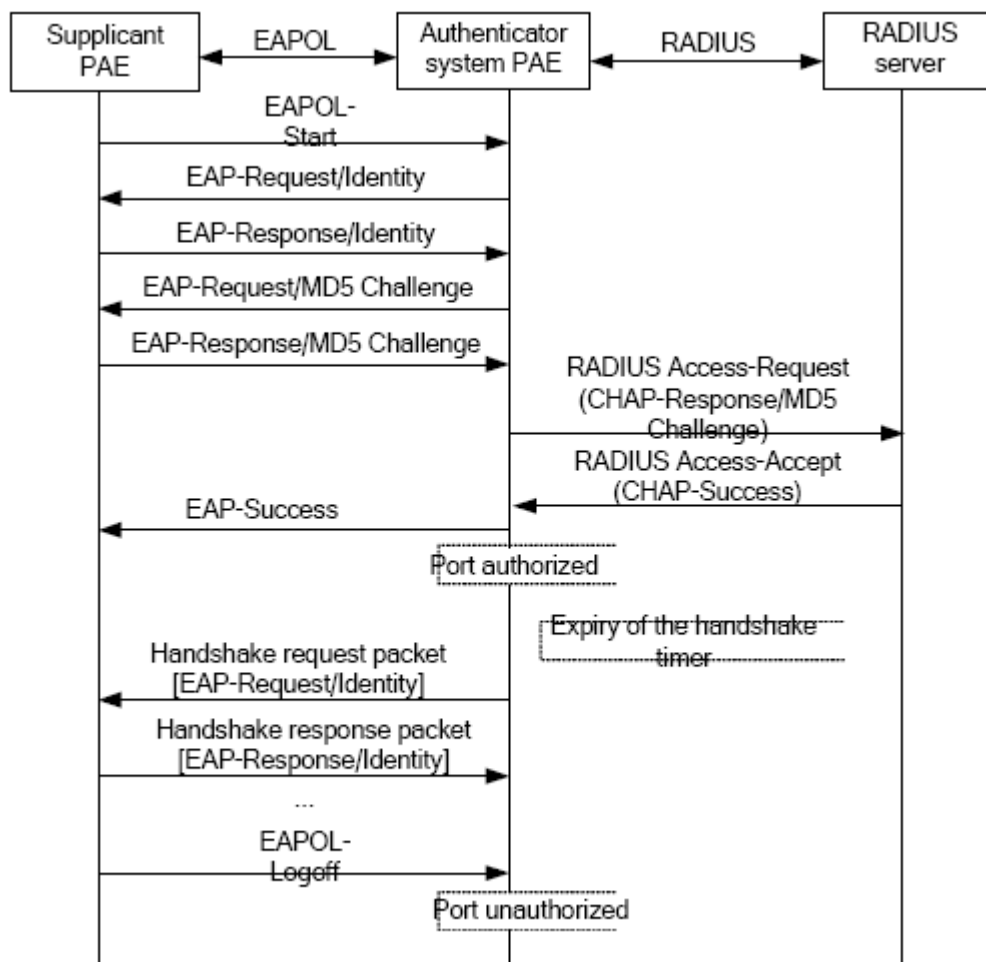


图 6-12 802.1x EAP 终结方式认证流程

EAP 终结方式与 EAP 中继方式的认证流程相比，不同之处在于用来对用户密码信息进行加密处理的随机加密字由设备端生成，之后接入控制单元会把用户名、随机加密字和客户端加密后的密码信息一起送给 RADIUS 服务器，进行相关的认证处理。

6.2.1.7 802.1x 的扩展和优化

设备在 802.1x 的 EAP 中继方式和 EAP 终结方式的实现中，不仅支持协议所规定的端口接入认证方式，还对其进行了扩展、优化：

- (2) 支持一个物理端口下挂接多个用户的应用场合；
- (3) 接入控制方式（即对用户的认证方式）可以采用基于端口、基于 MAC 和基于用户（IP 地址+MAC 地址+端口）三种方式：
 - 当采用基于端口方式时，只要该端口下的第一个用户认证成功后，其它接入用户无须认证就可使用网络资源，但是当第一个用户下线后，其它用户也会被拒绝使用网络。
 - 当采用基于 MAC 方式时，对于同一物理端口下的所有接入用户均需要单独认证，只有通过了认证的用户才能够接入网络，没有通过认证的用户不能接入网络。当某个用户下线时，也只有该用户无法使用网络。

- 当采用基于用户（IP地址+MAC地址+端口）方式时，它允许用户在认证之前就能够访问有限资源。对基于用户的接入控制方式，存在标准控制与高级控制两种类型：基于用户的标准控制类型不对有限资源的访问进行限制，该端口下的所有用户在没有经过认证之前都可以访问有限资源，当认证通过后可以访问所有资源；但基于用户的高级控制类型会对有限资源的访问进行限制，在该端口下只有特定用户在没有经过认证之前能够访问有限资源，当这些用户认证通过后可以访问所有资源。

注意：当使用私有的客户端时，推荐采用基于用户的高级接入控制方式，采用这种方式可以有效地防止 ARP 欺骗。

交换机最大认证用户数可以达到 4000 人，推荐使用认证用户数不超过 2000 人。

6.2.1.8 VLAN 分配特性

1. Auto VLAN

Auto VLAN 特性可以使 RADIUS 服务器根据用户信息和用户接入设备信息来动态改变接入端口所属的 VLAN。当 802.1x 用户在服务器上通过认证时，RADIUS 服务器会把授权信息传送给设备端。如果 RADIUS 服务器上配置了下发 VLAN 功能，则在 Access-Accept 报文中应当含有以下属性：

4. Tunnel-Type = VLAN (13)
5. Tunnel-Medium-Type = 802 (6)
6. Tunnel-Private-Group-ID = VLANID

这里的 VLANID 是指 VLAN 的 VID，取值范围为 1~4094。例如 Tunnel-Private-Group-ID = 30 表示 VLAN 30。

当交换机接收到下发的 Auto VLAN 信息后，当前 Access 端口离开用户配置的 VLAN 并加入 Auto VLAN 中。

Auto VLAN 并不改变端口的配置，也不影响端口的配置。但是，Auto VLAN 的优先级高于用户配置的 VLAN，即通过认证后起作用的 VLAN 是 Auto VLAN，用户配置的 VLAN 在用户下线后生效。

说明：目前 Auto VLAN 只能在基于端口的接入控制方式，以及链路类型为 Access 类型的端口上生效。

2. Guest VLAN

Guest VLAN 特性用来允许未认证用户访问某些特定资源。

用户认证端口在通过 802.1x 认证之前属于一个缺省 VLAN（即 Guest VLAN），用户访问该 VLAN 内的资源不需要认证，但此时不能够访问其他网络资源；认证成功后，端口离

开 Guest VLAN，用户可以访问其他的网络资源。

用户在 Guest VLAN 中可以获取 802.1x 客户端软件，升级客户端，或执行其他一些应用升级程序（例如防病毒软件、操作系统补丁程序等）。如果因为没有专用的认证客户端或者客户端版本过低等原因，导致在一定的时间内端口上无客户端认证成功，接入设备会把该端口加入到 Guest VLAN。

开启 802.1x 特性并正确配置 Guest VLAN 后，当设备从某一端口发送触发认证报文（EAP-Request/Identity）超过设定的最大次数而没有收到客户端的任何回应报文时，与 Auto VLAN 类似，该端口将被加入到 Guest VLAN 内。此时 Guest VLAN 中端口下的用户发起认证，如果认证失败，该端口将会仍然处在 Guest VLAN 内；如果认证成功，分为以下两种情况：

- ☞ 认证服务器下发一个 Auto VLAN，这时端口离开 Guest VLAN，加入下发的 Auto VLAN 中。用户下线后，端口会被重新划分到所配置的 Guest Vlan 内。
- ☞ 认证服务器不下发 VLAN，这时端口离开 Guest VLAN，加入配置的 VLAN 中。用户下线后，端口会被重新划分到所配置的 Guest Vlan 内。

6.2.2 802.1x 配置

802.1x 配置任务序列如下：

1. 使能交换机的 802.1x 功能
2. 接入控制单元的属性配置
 - 1) 配置端口的接入控制方式
 - 2) 配置交换机 802.1x 的扩展功能
 - 3) 配置端口的 IPv6 透传功能
3. 与 Supplicant (用户接入设备)相关的属性配置（可选）

1.使能交换机的 802.1x 功能

命令	解释
全局配置模式	
dot1x enable no dot1x enable	使能交换机全局及端口的 802.1x 功能；本命令的 no 操作为关闭 802.1x 功能。
dot1x privateclient enable no dot1x privateclient enable	使能交换机强制客户端软件使用神州数码私有 802.1x 认证报文格式；本命令的 no 操作为关闭该功能，允许客户端软件使用标准 802.1x 认证报文格式。
dot1x user free-resource <prefix> <mask> no dot1x user free-resource	设置用户能够访问的有限资源；本命令的 no 操作为删除有限资源。

dot1x unicast enable	打开交换机全局 802.1x 单播透传功能；本命令的 no 操作关闭 802.1x 单播透传功能。
no dot1x unicast enable	

2. 接入控制单元的属配置

1) 配置端口的接入控制方式

命令	解释
端口配置模式	
dot1x port-method {macbased userbased { advanced}} no dot1x port-method	设置端口的接入控制方式；本命令的 no 操作用来恢复缺省的接入控制方式。
dot1x max-user userbased <number> no dot1x max-user userbased	设置指定端口最多允许接入的用户数，用于端口接入控制方式为 userbased 的情况；本命令的 no 操作为恢复缺省的最多允许 10 个用户。

2) 配置交换机 802.1x 的扩展功能

命令	解释
全局配置模式	
dot1x macfilter enable no dot1x macfilter enable	打开交换机 802.1x 的地址过滤功能；本命令的 no 操作为关闭 802.1x 的地址过滤功能。
dot1x macbased port-down-flush no dot1x macbased port-down-flush	打开该命令，当基于 mac 进行 dot1x 认证的端口 down 时，删除该端口上已经认证通过的用户；本命令的 no 操作不对该情况下已认证用户进行下线操作。
dot1x accept-mac <mac-address> [interface <interface-name>] no dot1x accept-mac <mac-address> [interface <interface-name>]	添加 802.1x 的地址过滤表的表项；本命令的 no 操作为删除 802.1x 的地址过滤表的表项。
dot1x eapoe enable no dot1x eapoe enable	打开交换机 EAP 中继的认证方式；本命令的 no 操作为采用 EAP 本地终结的认证方式。

3) 配置端口的 IPv6 透传功能

命令	解释
端口配置模式	

dot1x ipv6 passthrough no dot1x ipv6 passthrough	使能交换机端口的 IPv6 透传功能, 用于端口接入控制方式为 userbased 的情况; 本命令的 no 操作为关闭该功能。

3.与 Supplicant (用户接入设备)相关的属性配置

命令	解释
全局配置模式	
dot1x max-req <count> no dot1x max-req	设置交换机在没有收到 supplicant 回应而重新启动认证前, 发送 EAP-request/MD5 帧的最大次数; 本命令的 no 操作用来恢复缺省值。
dot1x re-authentication no dot1x re-authentication	设置允许对 supplicant 进行周期性重认证; 本命令的 no 操作用来关闭该功能。
dot1x timeout quiet-period <seconds> no dot1x timeout quiet-period	设置在端口认证失败后保持静默状态的时间; 本命令的 no 操作用来恢复缺省值。
dot1x timeout re-authperiod <seconds> no dot1x timeout re-authperiod	设置交换机对 supplicant 重新认证的时间间隔; 本命令的 no 操作用来恢复缺省值。
dot1x timeout tx-period <seconds> no dot1x timeout tx-period	设置交换机对 supplicant 重发 EAP-request/identity 帧的时间间隔; 本命令的 no 操作用来恢复缺省值。
dot1x re-authenticate [interface <interface-name>]	设置对所有端口或某个指定端口进行 802.1x 重认证 (不须等待超时)。

6.2.3 802.1x 应用举例

6.2.3.1 Guest Vlan 应用举例

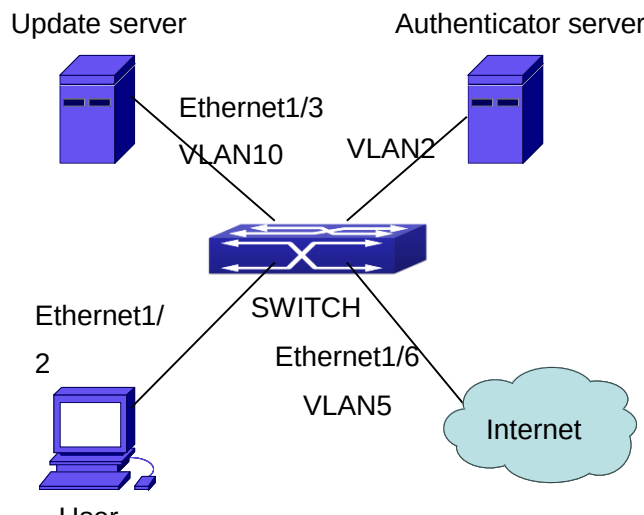


图 6-13 Guest VLAN 网络拓扑结构

如图 6-13 所示，1 台交换机通过 802.1x 认证接入网络，认证服务器为 RADIUS 服务器。User 接入交换机的端口 Ethernet1/2 下，端口 Ethernet 1/2 在 VLAN100 内；认证服务器在 VLAN2 内；Update Server 是用于客户端软件下载和升级的服务器，在 VLAN10 内；交换机连接 Internet 网络的端口 Ethernet 1/6 在 VLAN5 内。

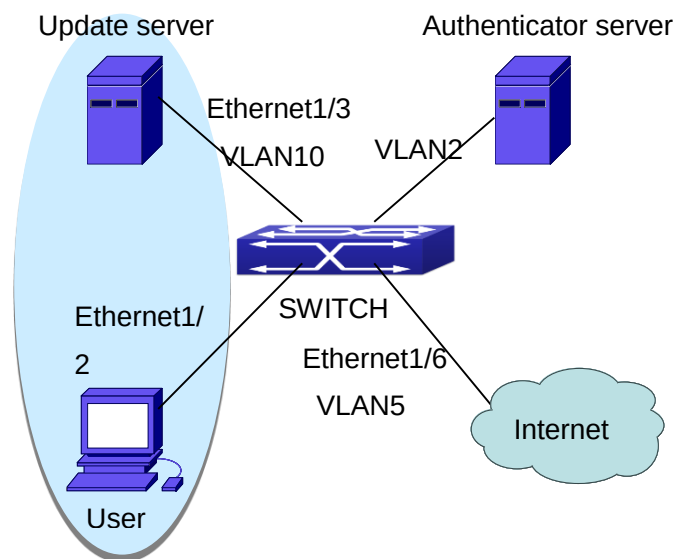


图 6-14 用户加入 Guest VLAN

如图 6-14 所示，在交换机端口 Ethernet1/2 上开启 802.1x 特性，并配置 VLAN10 为该端口 Guest VLAN，当用户没有认证或者认证失败时，端口 Ethernet1/2 被加入到 VLAN10 中，用户可以访问 Update Server。

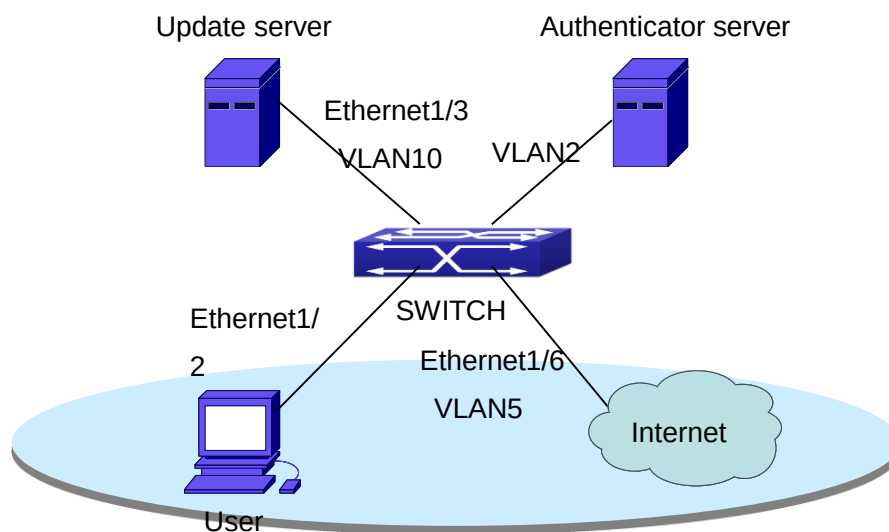


图 6-15 用户上线，VLAN 下发

如图 6-15 所示，当用户认证成功上线，认证服务器下发 VLAN5，此时，用户和端口 Ethernet1/6 都在 VLAN5 内，用户可以访问 Internet。

配置步骤如下：

配置 RADIUS 服务器。

```
Switch(config)#radius-server authentication host 10.1.1.3
```

```
Switch(config)#radius-server accounting host 10.1.1.3
```

```
Switch(config)#radius-server key test
```

```
Switch(config)#aaa enable
```

```
Switch(config)#aaa-accounting enable
```

创建 VLAN100。

```
Switch(config)#vlan 100
```

使能全局 802.1x 功能。

```
Switch(config)#dot1x enable
```

使能端口 Ethernet1/2 上的 802.1x 功能。

```
Switch(config)#interface ethernet 1/2
```

```
Switch(Config-Ethernet1/2)#dot1x enable
```

配置端口的链路类型为 **access** 模式。

```
Switch(Config-Ethernet1/2)#switch-port mode access
```

配置端口上接入控制方式为 macbased。


```
Switch(Config-Ethernet1/2)#dot1x port-method macbased
```

通过命令 **show running-config** 或者 **show interface ethernet 1/2** 可以查看 Guest VLAN 配置情况。在没有用户上线、用户认证失败或用户成功下线等情况下发送触发认证报文（EAP-Request/Identity）超过设定的最大次数时，通过命令 **show vlan id 100** 可以查看端口配置的 Guest VLAN 是否生效。

6.2.3.2 802.1x 与 IPv4 Radius 应用举例

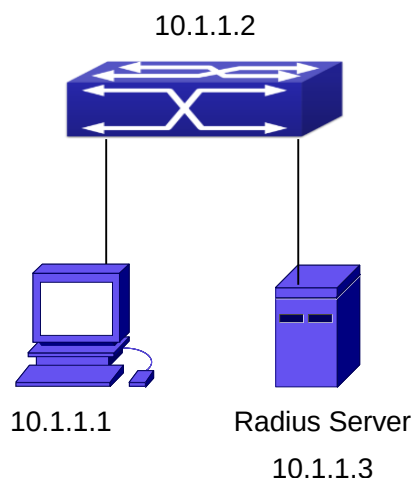


图 6-16 IEEE802.1x 配置举例拓扑图

计算机连接到交换机的端口 Ethernet1/2 上，端口 Ethernet1/2 开启 IEEE802.1x 认证功能，接入方式采用缺省的基于 MAC 地址认证方式。交换机的 IP 地址设置为 10.1.1.2，并将除端口 Ethernet1/2 以外的任意一个端口与 RADIUS 认证服务器相连接，RADIUS 认证服务器的 IP 地址设置为 10.1.1.3，认证、计费端口为缺省端口 1812 和端口 1813。计算机上安装 IEEE802.1x 认证客户端软件，并通过使用此软件来实现 IEEE802.1x 认证。

配置步骤如下：

```
Switch(config)#interface vlan 1
Switch(Config-if-vlan1)#ip address 10.1.1.2 255.255.255.0
Switch(Config-if-vlan1)#exit
Switch(config)#radius-server authentication host 10.1.1.3
Switch(config)#radius-server accounting host 10.1.1.3
Switch(config)#radius-server key test
Switch(config)#aaa enable
Switch(config)#aaa-accounting enable
Switch(config)#dot1x enable
Switch(config)#interface ethernet 1/2
Switch(Config-Ethernet1/2)#dot1x enable
```

```
Switch(Config-Ethernet1/2)#exit
```

6.2.3.3 802.1x 与 IPv6 Radius 应用举例

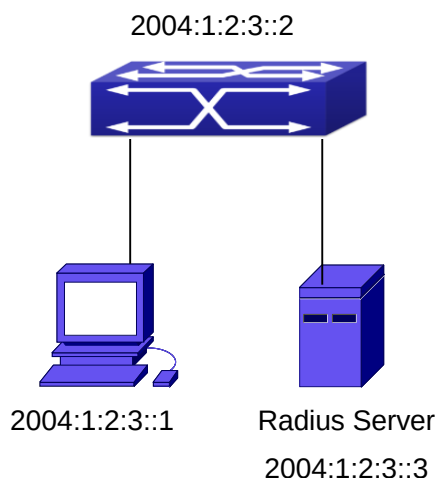


图 6-17 IPv6 Radius 配置举例拓扑图

计算机连接到交换机的端口 1/2 上，端口 1/2 开启 IEEE802.1x 认证功能，接入方式采用缺省的基于 MAC 地址认证方式。交换机的 IP 地址设置为 2004:1:2:3::2，并将除端口 1/2 以外的任意一个端口与 RADIUS 认证服务器相连接，RADIUS 认证服务器的 IP 地址设置为 2004:1:2:3::3，认证、计费端口为缺省端口 1812 和端口 1813。计算机上安装 IEEE802.1x 认证客户端软件，并通过使用此软件来实现 IEEE802.1x 认证。

配置步骤如下：

```
Switch(config)#interface vlan 1
Switch(Config-if-vlan1)#ipv6 address 2004:1:2:3::2/64
Switch(Config-if-vlan1)#exit
Switch(config)#radius-server authentication host 2004:1:2:3::3
Switch(config)#radius-server accounting host 2004:1:2:3::3
Switch(config)#radius-server key test
Switch(config)#aaa enable
Switch(config)#aaa-accounting enable
Switch(config)#dot1x enable
Switch(config)#interface ethernet 1/2
Switch(Config-If-Ethernet1/2)#dot1x enable
Switch(Config-If-Ethernet1/2)#exit
```

6.2.3.4 基于 802.1x 的 Web 认证代理应用举例

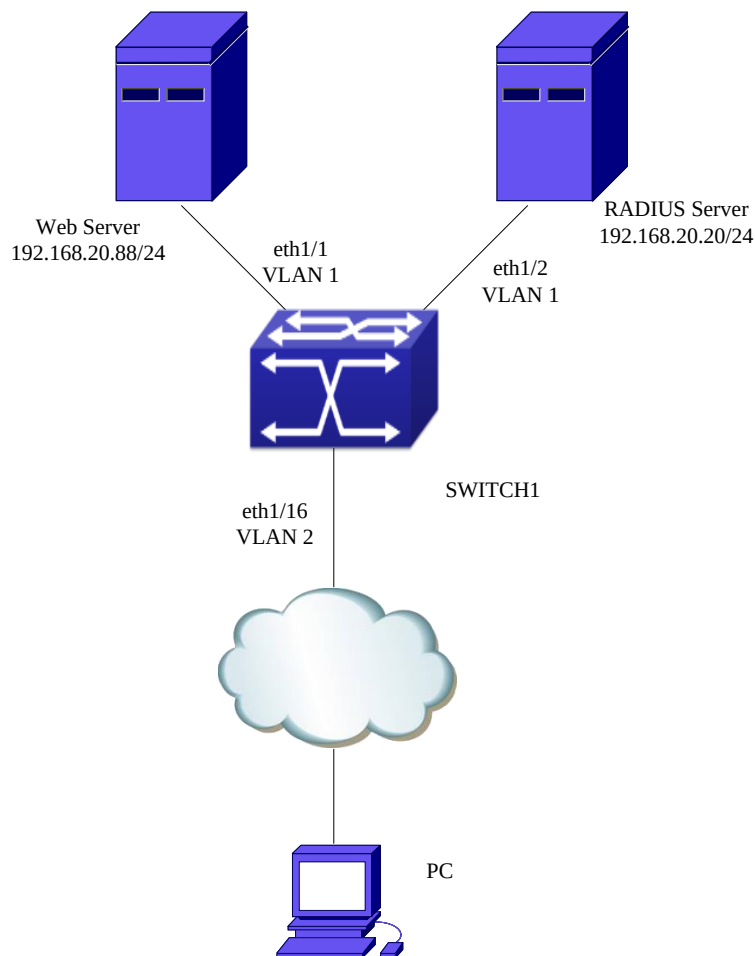


图 6-18 基于 802.1x 的 Web 认证典型配置案例

在上述网络拓扑图中，SWITCH1 的端口 Ethernet1/1 与 Web 服务器（IP 地址为 192.168.20.20/24）相连，SWITCH1 的端口 Ethernet1/2 与 RADIUS 认证服务器（IP 地址为 192.168.20.88/24，认证端口为 1812）相连，PC 通过一团不可知的网络云与 SWITCH1 的端口 Ethernet1/16 相连。Web 服务器和认证服务器处于 VLAN 1 中，PC 处于 VLAN 2 中。可通过下面的配置启动基于 802.1x 的 Web 认证。重认证机制默认是关闭的，要打开重认证功能要执行相应的 802.1x 配置命令。

SWITCH1 配置任务序列：

```
Switch(config)#dot1x enable
```

```
Switch(config)#interface ethernet 1/16
```

```
Switch(Config-If-Ethernet1/16)#dot1x enable
```

```
Switch(Config-If-Ethernet1/16)#dot1x port-method macbased
```

6.2.4 802.1x 排错帮助

用户在使用 802.1x 时，通常会遇到端口无法配置 802.1x；或者 802.1x 认证状态为 auto，

用户运行 802.1x 的 supplicant 软件后，端口仍不能改变为认证通过状态的情况。可能的原因及解决建议如下：

19. 如果出现端口无法配置 802.1x 的情况，请检查交换机的认证端口是否运行了端口 mac 绑定，或者端口已经配置为聚合端口。如果要打开端口的 802.1x 认证功能，则必须关闭该端口下的上述功能。
20. 如果交换机配置正确，但认证仍无法通过，请检查交换机与 RADIUS 服务器，交换机与 802.1x 客户机之间是否相互连通；检查交换机端口 VLAN 的配置。
21. 通过查看 RADIUS 服务器的事件日志，判断问题的起因。在事件日志中对登录不成功的不仅有记录还有不成功原因的提示。如事件日志显示 authenticator 的登录口令不对，则修改 radius-server key 参数；如果事件日志中显示没有该 authenticator，则需在 RADIUS 服务器中增加该 authenticator；如事件日志中提示没有该登录用户，说明用户的登录名和口令有误，输入正确的用户名和口令。
22. 基于 802.1x 的 Web 认证代理默认是关闭的。打开基于 802.1x 的 Web 认证代理后，可以同时打开调试开关 debug dot1x 来查看调试信息。
23. 使用 show dot1x 时如果端口的状态显示没有被关闭，只能说明没有被基于 802.1x 的 Web 认证代理功能关闭。
24. 基于 802.1x 的 Web 认证代理的整个交换机最多能实现 1024 个认证成功用户同时在线，如果超过了此限制，将会返回提示信息。
25. 当用户使用 Web 认证无法成功时，检查是否启用了 dot1x privateclient enable 命令，若已启用，则需要关闭此私有认证功能。

6.3 端口、VLAN 中 MAC、IP 数量限制

6.3.1 端口、VLAN 中 MAC、IP 数量限制功能简介

MAC 地址表是标识目的 MAC 地址与交换机端口之间映射关系的表，其 MAC 地址分为静态 MAC 地址和动态 MAC 地址。静态 MAC 地址由用户配置，具有最高优先级（不能被动态 MAC 地址覆盖）且永久生效；动态 MAC 地址由交换机在转发数据帧的过程中学习，且在有限时间内生效。当交换机接收到需要转发的数据帧时，首先学习数据帧的源 MAC 地址，与接收端口建立映射关系；然后根据目标 MAC 地址查询 MAC 地址表，如果命中相关表项，交换机将数据帧从相应端口转发；否则，交换机将数据帧在其所属 VLAN 内广播。

通常交换机支持静态配置和动态学习 MAC 地址的功能，每个端口可以静态配置和动态学习多个 MAC 地址，从而实现端口之间已知 MAC 地址数据流的转发。当 MAC 地址老化后，则进行广播处理。在我们目前的交换机中，端口上对 MAC 地址的数量没有限制，允许一个端口配置或者学习多个 MAC 地址，直到硬件表项填满为止。为了限制端口过量的 MAC

地址，我们需要限制端口的 MAC 地址数量。

对于每一个 INTERFACE VLAN 中的 IP 数量也没有限制，IP 数量限制也就是接口上用户数量的限制，也就是 ARP、ND 表项的限制，这些表项没有相关的配置命令可以控制下发的数量。为了提高产品的安全性能和可控性，我们需要将端口上的 MAC 地址的数量和每个 INTERFACE VLAN 中 ARP、ND 数量进行控制，端口只允许配置的最大数量的动态 MAC 地址存在。每一个 VLAN 上的用户数量不能超过配置允许的用户数量。

对 MAC、ARP 表项进行限制可以在一定程度上防止 DOS 攻击。当恶意用户频繁发动 MAC、ARP 欺骗时，很容易把交换机的 MAC、ARP 表项填满，从而导致 DOS 攻击的发生。对数量进行限制，可以预防 DOS 攻击的发生。

综上节所述，开发端口、VLAN 中的 MAC、IP 数量限制的功能具有重大意义。交换机将可以通过配置命令控制端口的 MAC 地址的数量，同时可以通过配置命令控制端口和 VLAN 中的 ARP、ND 表项的数量。

端口动态 MAC、IP 数量的限制：

- 1.动态 MAC 数量限制，对于交换机已经动态学习到的 MAC 地址，如果大于等于允许学习的最大动态 MAC 数量时，则关闭该端口的 MAC 学习功能，如果少于允许学习的最大动态 MAC 数量时，这时仍然可以继续学习。
- 2.动态 IP 数量限制，对于交换机已经动态学习到的 ARP、ND，如果大于等于允许学习的最大动态 ARP、ND 数量时，则可以关闭该端口的 ARP、ND 学习功能，如果少于允许学习的最大动态 ARP、ND 数量时，这时仍然可以继续学习。

接口 MAC、ARP、ND 数量的限制：

- 1.动态 MAC 数量限制，对于交换机 VLAN 中已经动态学习到的 MAC 地址，如果大于允许学习的最大动态 MAC 数量时，则可以关闭该 VLAN 中所有端口的 MAC 学习功能，如果少于允许学习的最大动态 MAC 数量时，这时 VLAN 中所有端口仍然可以继续学习。（特殊端口除外）
- 2.动态 IP 数量限制，对于交换机已经动态学习到的 ARP、ND，如果大于等于允许学习的最大动态 ARP、ND 数量时，则该 VLAN 不会学习新的 ARP、ND，如果少于允许学习的最大动态 ARP、ND 数量时，这时仍然可以继续学习。

6.3.2 端口、VLAN 中 MAC、IP 数量限制功能配置任务序列

- 1) 启动端口上 MAC、IP 数量限制功能
- 2) 启动 VLAN 中 IP 数量限制功能
- 3) 配置查询动态 MAC 的超时时间
- 4) 配置端口违背模式
- 5) 显示和调试端口上 MAC、IP 数量限制相关信息

- 1) 启动端口上 MAC、IP 数量限制功能

命令	解释

端口配置模式	
switchport mac-address dynamic maximum <value> no switchport mac-address dynamic maximum	启动和关闭端口 MAC 数量限制功能
switchport arp dynamic maximum <value> no switchport arp dynamic maximum	启动和关闭端口 ARP 数量限制功能
switchport nd dynamic maximum <value> no switchport nd dynamic maximum	启动和关闭端口 ND 数量限制功能

2) 启动 VLAN 中 IP 数量限制功能

命令	解释
VLAN 配置模式	
接口配置模式	
ip arp dynamic maximum <value> no ip arp dynamic maximum	启动和关闭 VLAN 中 ARP 数量限制功能
ipv6 nd dynamic maximum <value> no ipv6 nd dynamic maximum	启动和关闭 VLAN 中 NEIGHBOR 数量限制功能

3) 配置查询动态 MAC 的超时时间

命令	解释
全局配置模式	
mac-address query timeout <seconds>	设置定时查询动态 MAC 的超时时间

4) 配置端口违背模式

命令	解释
端口配置模式	
switchport mac-address violation {protect shutdown} [recovery <5-3600>] no switchport mac-address violation	设置端口违背模式；本命令的 no 操作为恢复违背模式为 protect。

5) 显示和调试端口上 MAC、IP 数量限制相关信息

命令	解释
特权配置模式	

show mac-address dynamic count {vlan <vlan-id> interface ethernet <portName> }	显示相应端口、VLAN 中的动态 MAC 数量
show arp-dynamic count {vlan <vlan-id> interface ethernet <portName> }	显示相应端口、VLAN 中的动态 ARP 数量
show nd-dynamic count {vlan <vlan-id> interface ethernet <portName> }	显示相应端口、VLAN 中的动态 NEIGHBOR 数量
debug switchport mac count no debug switchport mac count	端口 MAC 数量限制时，各种调试信息
debug switchport arp count no debug switchport arp count	端口 ARP 数量限制时，各种调试信息
debug switchport nd count no debug switchport nd count	端口 NEIGHBOR 数量限制时，各种调试 信息
debug vlan mac count no debug vlan mac count	VLAN 中 MAC 数量限制时，各种调试信息
debug ip arp count no debug ip arp count	VLAN 中 ARP 数量限制时，各种调试信息
debug ipv6 nd count no debug ipv6 nd count	VLAN 中 NEIGHBOR 数量限制时，各种 调试信息

6.3.3 端口、VLAN 中 MAC、IP 数量限制功能典型案例

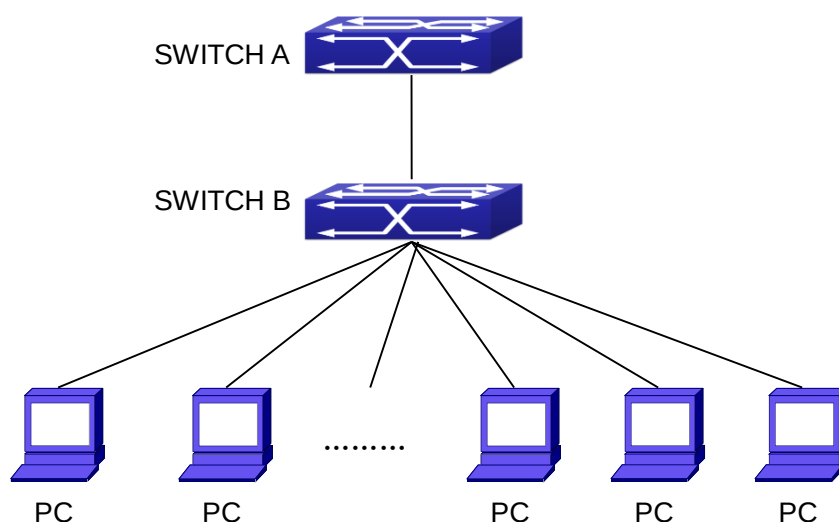


图 6-19 端口、VLAN 中 MAC、IP 数量限制典型配置案例

如上图所示的网络拓扑中, SWITCH B 下面链接很多 PC 用户, 在没有启动端口 MAC、VLAN

中 IP 数量限制功能前，在系统硬件允许的情况下，SWITCH A，SWITCH B 可以得到所有 PC 的 MAC、ARP、ND 表项，对 MAC、ARP 表项进行限制可以在一定程度上防止 DOS 攻击。当恶意用户频繁发动 MAC、ARP 欺骗时，很容易把交换机的 MAC、ARP 表项填满，从而导致 DOS 攻击的发生。对 MAC、ARP、ND 表项数量进行限制，可以预防 DOS 攻击的发生。

设置 SWITCH A 端口 Ethernet1/1 上最大可以学习 20 个动态 MAC 地址、20 个动态 ARP 地址、10 个 NEIGHBOR 表项，VLAN 1 中允许 30 个动态 MAC 地址，30 个动态 ARP 地址、20 个 NEIGHBOR 表项。

SWITCH A 配置任务序列：

```
Switch (config)#interface ethernet 1/1
```

```
Switch (Config-If-Ethernet1/1)#switchport mac-address dynamic maximum 20
```

```
Switch (Config-If-Ethernet1/1)#switchport arp dynamic maximum 20
```

```
Switch (Config-If-Ethernet1/1)#switchport nd dynamic maximum 10
```

6.3.4 端口、VLAN 中 MAC、IP 数量限制功能排错帮助

端口、VLAN 中 MAC、IP 数量限制功能默认情况下是关闭的，如果需要限制网络中接入的用户的数量可以打开该功能。如果出现端口无法配置 MAC 地址数量限制功能的情况，请检查交换机的端口是否运行了 Spanning-tree, dot1x, 端口汇聚或者端口已经配置为 MAC 绑定端口。MAC 数量限制在端口上与这些配置是互斥的，如果该端口要打开 MAC 地址数量限制功能，就必须首先确认端口下的上述功能已经被关闭。

如果在正常配置下，启动了端口、VLAN 中 MAC、IP 数量限制功能后，可以使用 debug 各个相关限制的调试命令查看数量限制的具体信息，以及数量限制功能是否正确，如果有什么问题可以将结果发送给技术服务中心。

6.4 AM

6.4.1 AM 功能简介

AM（Access Management—访问管理）是指当交换机收到 IP 报文或 ARP 报文时，它用收到报文的信息（源 IP 地址或者源 MAC-IP 地址）与配置硬件地址池相比较，如果在配置硬件地址池中找到与收到的报文相匹配的信息（源 IP 地址或者源 MAC-IP 地址）则转发该报文，否则丢弃。之所以在基于源 IP 地址的访问管理上增加基于源 MAC-IP 的访问管理，是因为对主机而言，IP 地址是可变的。如果只有 IP 绑定，用户可以把主机 IP 地址改为转发 IP，从而使本主机发出的报文能够被交换机转发。而 MAC-IP 可以与主机唯一绑定，所以为了防止用户恶意修改主机 IP 地址来使本主机发出的报文能被交换机转发，MAC-IP 的绑定是必要的。

通过 AM 访问管理的端口绑定特性，网络管理员可以将合法用户的 IP（MAC-IP）地址绑定到指定的端口上。进行绑定操作后，只有指定 IP（MAC-IP）地址的用户发出的报文才能通过该端口转发，增强了用户对网络安全的监控。

6.4.2 AM 功能配置任务序列

1. 启动 AM 功能
2. 启动某端口的 AM 功能
3. 配置转发 IP
4. 配置转发 MAC-IP
5. 删除所有已配置的转发 IP 或 MAC-IP 或全部
6. 显示 AM 相关配置信息

1. 启动 AM 功能

命令	解释
全局配置模式	
am enable no am enable	全局启动或关闭 AM 功能。

2. 启动某端口的 AM 功能

命令	解释
端口配置模式	
am port no am port	启动/关闭端口的 AM 功能，端口 AM 功能启动后，默认禁止所有的 IP 报文与 ARP 报文转发。

3. 配置转发 IP

命令	解释
端口配置模式	
am ip-pool <ip-address> <num> no am ip-pool <ip-address> <num>	配置端口的转发 IP。

4. 配置转发 MAC-IP

命令	解释
端口配置模式	

am mac-ip-pool <mac-address> <ip-address> no am mac-ip-pool <mac-address> <ip-address>	配置端口的转发 MAC-IP。

5. 删除所有已配置的转发 IP 或 MAC-IP 或全部

命令	解释
全局配置模式	
no am all [ip-pool mac-ip-pool]	删除全部用户配置的 MAC-IP 地址池或 IP 地址池或两个地址池都删除。

6. 显示 AM 相关配置信息

命令	解释
全局配置模式	
show am [interface <interface-name>]	显示全部或某一特定端口的 AM 配置信息。

6.4.3 AM 功能典型案例

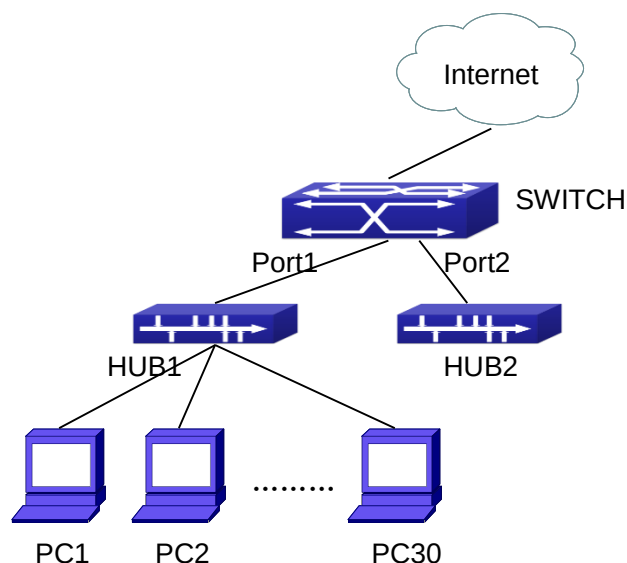


图 6-20 AM 功能典型配置案例

在上述网络拓扑图中，30 台 PC 通过 HUB1 汇集后与交换机的端口 1 相连，30 台 PC 的 IP 地址范围 100.10.10.1~100.10.10.30，出于安全考虑，系统管理员认为只有 IP 地址范围在 100.10.10.1~100.10.10.10 内的用户为合法用户，交换机只能对合法用户发来的数据包进行转发，对 IP 地址不在该范围内的用户发来的数据包，交换机不能转发，直接丢弃。

基于上述要求，在交换机上可做如下配置：

```
Switch(config)#am enable
```

```
Switch(config)#interface ethernet1/1
Switch(Config-If-Ethernet 1/1)#am port
Switch(Config-If-Ethernet 1/1)#am ip-pool 10.10.10.1 10
```

6.4.4 AM 功能排错帮助

AM 功能默认是关闭的，打开 AM 功能后，可以进行 AM 的相关配置。

可以使用 `show am` 来查看当前 AM 配置，包括 AM 是否打开，各个端口所配置的 AM 信息等，也可以使用 `show am [interface <interface-name>]` 来查看具体端口的 AM 配置信息。

在操作过程中，如果有操作错误，系统将会给出具体的错误提示信息。

6.5 安全特性

6.5.1 安全特性介绍

在介绍安全特性之前，先介绍一下 DoS 的概念。DoS 是 Denial of Service 的缩写，意为拒绝服务。DoS 攻击是网络上一种简单但很有效的破坏性攻击手段，服务器会由于不停顿地处理攻击者的数据包，从而正常用户发送的数据包会被丢弃，得不到处理，从而造成了服务器的拒绝服务，更严重的会导致服务器敏感数据泄漏。

安全特性是指利用协议检查来防范 DoS 攻击等安全应用，协议检查允许用户基于给定条件丢弃满足条件的报文。安全特性为用户提供了若干种简单有效的防御 DoS 攻击的方法，并且不影响交换机的线性转发性能。

6.5.2 安全特性配置

6.5.2.1 防 IP Spoofing 功能配置任务序列

使能 IP Spoofing 功能

命令	解释
全局配置模式	
[no] dosattack-check srcip-equal-dstip enable	开启(关闭)检查 IP 源地址等于目的地址的功能。

6.5.2.2 防 TCP 非法标志攻击功能配置任务序列

1. 使能 TCP 非法标志攻击功能

2. 使能检查 IPv4 分片功能

命令	解释
全局配置模式	
[no] dosattack-check tcp-flags enable	开启(关闭)检查 TCP 标志功能。
[no] dosattack-check ipv4-first-fragment enable	开启(关闭)检查 IPv4 分片功能, 单独使用该命令无作用, 若不开启该功能, 交换机不会丢弃包含了非法 TCP 标志的 IPv4 分片报文。

6.5.2.3 防端口欺骗功能配置任务序列

1. 使能防端口欺骗功能

命令	解释
全局配置模式	
[no] dosattack-check srcport-equal-dstport enable	开启(关闭)防端口欺骗功能。
[no] dosattack-check ipv4-first-fragment enable	开启(关闭)检查 IPv4 分片功能, 单独使用该命令无作用, 若不开启该功能, 交换机不会丢弃源端口等于目的端口的 IPv4 分片报文。

6.5.2.4 防 TCP 碎片攻击功能配置任务序列

1. 使能防 TCP 碎片攻击功能

2. 设置允许通过的最小 TCP 报文段长度

命令	解释
全局配置模式	
[no] dosattack-check tcp-fragment enable	开启(关闭)防 TCP 碎片攻击功能。
dosattack-check tcp-segment <size>	设置允许通过的最小 TCP 报文段长度, 单独使用该命令无作用, 需要开启 dosattack-check tcp-fragment enable。

6.5.2.5 防 ICMP 碎片攻击功能配置任务序列

1. 使能防 ICMP 碎片攻击功能

2. 设置允许通过的最大的 ICMPv4 净荷长度
3. 设置允许通过的最大的 ICMPv6 净荷长度

命令	解释
全局配置模式	
[no] dosattack-check icmp-attacking enable	开启防(关闭)ICMP 碎片攻击功能。
dosattack-check icmpV4-size <size>	设置允许通过的最大的 ICMPv4 净荷长度，单独使用该命令无作用，需要开启 dosattack-check icmp-attacking enable。
dosattack-check icmpV6-size <size>	设置允许通过的最大的 ICMPv6 净荷长度，单独使用该命令无作用，需要开启 dosattack-check icmp-attacking enable。

6.5.3 安全特性举例

案例：

用户有如下配置需求：交换机不转发源 IP 地址等于目的 IP 地址的数据报，和源端口等于目的端口的数据报，且在 IPv4 网络内只允许使用默认选项的 ping 命令，即 ICMP request 报文不可分片，且净荷长度一般小于 100。

配置步骤如下：

```
Switch(config)#dosattack-check srcip-equal-dstip enable
Switch(config)#dosattack-check srcport-equal-dstport enable
Switch(config)#dosattack-check ipv4-first-fragment enable
Switch(config)#dosattack-check icmp-attacking enable
Switch(config)#dosattack-check icmpV4-size 100
```

6.6 TACACS+

6.6.1 TACACS+简介

TACACS+终端访问控制器访问控制协议是类似于radius协议的一个用于控制终端接入网络的协议。该协议同样提供Authentication(认证), Authorrization(授权), Accounting(记帐)三个独立的功能。与RADIUS 相比，TACACS+采用TCP协议作为传输层，并且对报文主体（除标准报文头）加密，因而具有更加可靠的传输和加密特性，更加适合于安全控制。

根据TACACS+(version 1.78)协议的特点。我们在交换机上提供了TACACS+的认证功能。用户在登录交换机后（例如telnet）可以利用TACACS+来进行用户名和密码的验证。

6.6.2 TACACS+配置

1. 配置 TACACS+认证密钥
2. 配置 TACACS+服务器
3. 配置 TACACS+认证超时时间
4. 配置 TACACS+ NAS-IP 地址

1. 配置 TACACS+认证密钥

命令	解释
全局配置模式	
tacacs-server key {0 7} <string> no tacacs-server key	设置 TACACS+服务器的密钥；本命令的 no 操作为删除 TACACS+服务器的密钥。

2. 配置 TACACS+ Server

命令	解释
全局配置模式	
tacacs-server authentication host <ip-address> [port <port-number>] [timeout <seconds>] [key {0 7} <string>] [primary] no tacacs-server authentication host <ip-address>	设置 TACACS+服务器 IP 地址、监听端口号、认证服务器超时时间、密钥字符串；本命令的 no 操作为删除 TACACS+认证服务器。

3. 配置 TACACS+认证超时时间

命令	解释
全局配置模式	
tacacs-server timeout <seconds> no tacacs-server timeout	配置 TACACS+服务器认证超时时间；本命令的 no 操作为恢复缺省配置。

4. 配置 TACACS+ NAS-IP 地址

命令	解释
全局配置模式	
tacacs-server nas-ipv4 <ip-address> no tacacs-server nas-ipv4	设置交换设备发送 TACACS+报文的 IP 源地址。

6.6.3 TACACS+典型案例

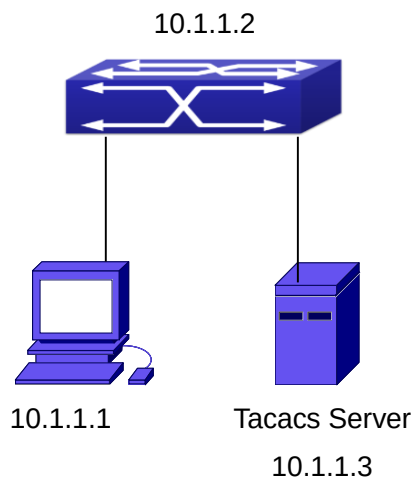


图 6-21 TACACS+配置示意图

计算机连接到交换机，交换机的 IP 地址设置为 10.1.1.2，并与一个 TACACS+认证服务器相连接，TACACS+认证服务器的 IP 地址设置为 10.1.1.3，认证端口为缺省端口 49。交换机的 telnet 登录认证方式设置为 tacacs local。计算机利用 telnet 登录交换机，交换机通过使用 TACACS+认证服务器实现对 telnet 用户的认证。

```
Switch(config)#interface vlan 1
Switch(Config-if-vlan1)#ip address 10.1.1.2 255.255.255.0
Switch(Config-if-vlan1)#exit
Switch(config)#tacacs-server authentication host 10.1.1.3
Switch(config)#tacacs-server key test
Switch(config)#authentication line vty login tacacs
```

6.6.4 TACACS+排错帮助

在配置、使用 TACACS+协议时，可能会由于物理连接、配置错误等原因导致 TACACS+认证失败。因此，用户应注意以下要点：

- ☞ 首先应该保证到 TACACS+服务器的物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 再次，保证交换机配置的 TACACS+密钥同 TACACS+服务器上配置的密钥一致；
- ☞ 然后，确保配置了正确的 TACACS+服务器。

6.7 RADIUS

6.7.1 RADIUS 简介

6.7.1.1 AAA 与 RADIUS 简介

AAA是Authentication， Authorization and Accounting（认证、授权和计费）的简称，它提供了网络安全管理的一致性框架，该框架通过认证、授权和计费三大功能满足了安全网络的访问控制需求：哪些用户可以访问网络设备，访问用户拥有哪些访问权限以及用户使用网络资源的统计计费。

RADIUS（Remote Authentication Dial-In User Service，远程认证拨号用户服务）是一种分布式的、客户端/服务器结构的信息交互协议。RADIUS客户端一般在网络设备上启用，与802.1x协议结合实现认证、授权和计费的所有主要功能，而RADIUS服务器负责维护用户认证、授权和计费信息数据库，通过RADIUS协议与客户端进行通信，完成用户认证和授权信息的下发以及计费信息的统计。RADIUS协议是实现AAA框架最常用的一种协议。

6.7.1.2 RADIUS 协议的报文结构

RADIUS 协议采用 UDP 报文来承载协议报文，报文格式如下：

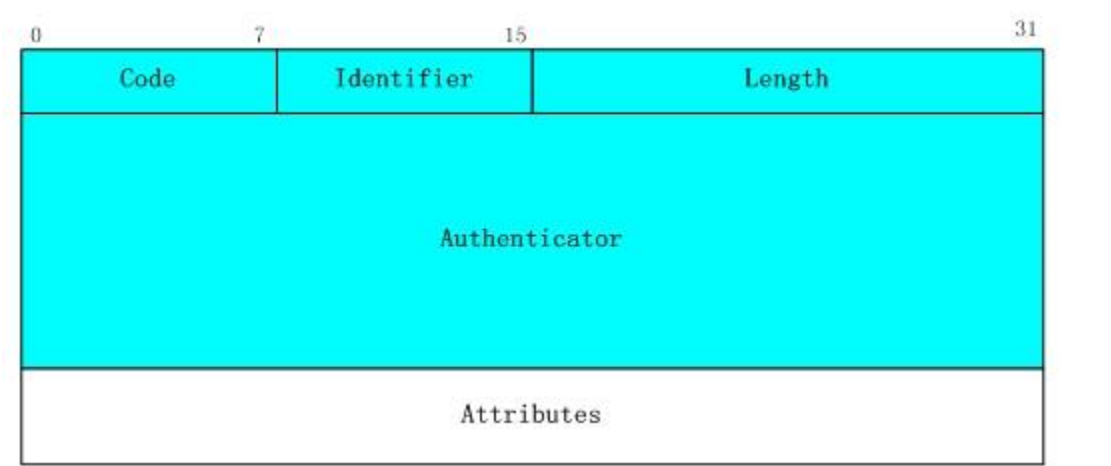


图 7-1 RADIUS 报文结构

Code 域（1 字节）：表示 RADIUS 报文的类型，取值如下，

- 1 Access-Request 认证请求报文
- 2 Access-Accept 认证接收报文
- 3 Access-Reject 认证拒绝报文
- 4 Accounting-Request 计费请求报文
- 5 Accounting-Response 计费回应报文
- 11 Access-Challenge 认证Challenge报文

Identifier 域（1 字节）：用作报文标识，用于匹配请求与应答报文。

Length 域（2 字节）：表示整个 RADIUS 报文的长度，包括 Code、Identifier、Length、

Authenticator 和 Attributes 域。

Authenticator 域（16 字节）：用于验证 RADIUS 服务器回应的报文，还能用于密码隐藏算法。分为 Request Authenticator 和 Response Authenticator。

Attribute 域：用于携带指明的认证和授权以及计费等相关信息，提供请求和相应报文的详细配置信息。Attribute 域采用（Type、Length、Value）三元组的形式构造。

☞ **Type 字段（1 字节），**表示属性的类型值，如下表列举出常用属性。

属性	属性类型	属性	属性类型
1	User-Name	23	Framed-IPX-Network
2	User-Password	24	State
3	CHAP-Password	25	Class
4	NAS-IP-Address	26	Vendor-Specific
5	NAS-Port	27	Session-Timeout
6	Service-Type	28	Idle-Timeout
7	Framed-Protocol	29	Termination-Action
8	Framed-IP-Address	30	Called-Station-Id
9	Framed-IP-Netmask	31	Calling-Station-Id
10	Framed-Routing	32	NAS-Identifier
11	Filter-Id	33	Proxy-State
12	Framed-MTU	34	Login-LAT-Service
13	Framed-Compression	35	Login-LAT-Node
14	Login-IP-Host	36	Login-LAT-Group
15	Login-Service	37	Framed-AppleTalk-Link
16	Login-TCP-Port	38	Framed-AppleTalk-Network
17	(unassigned)	39	Framed-AppleTalk-Zone
18	Reply-Message	40-59	(reserved for accounting)
19	Callback-Number	60	CHAP-Challenge
20	Callback-Id	61	NAS-Port-Type
21	(unassigned)	62	Port-Limit
22	Framed-Route	63	Login-LAT-Port

☞ **Length 字段（1 字节），**表示属性长度（包括 Type、Length 和 Value 字段），单位为字节。

☞ **Value 字段，**表示属性的具体信息，其格式和内容由类型域和长度域决定。

6.7.2 RADIUS 配置

1. 使能交换机的 AAA 认证和计费功能
2. 配置 RADIUS 认证密钥
3. 配置 RADIUS 服务器
4. 配置 RADIUS 服务的参数
5. 配置 RADIUS NAS-IP 地址

1. 使能交换机的 AAA 认证和计费功能

命令	解释
全局配置模式	
aaa enable no aaa enable	使能交换机的 AAA 认证功能；本命令的 no 操作为关闭交换机的 AAA 认证功能。
aaa-accounting enable no aaa-accounting enable	使能交换机的计费功能；本命令的 no 操作为关闭交换机的计费功能。
aaa-accounting update {enable disable}	打开或关闭计费更新功能。

2. 配置 RADIUS 认证密钥

命令	解释
全局配置模式	
radius-server key {0 7} <string> no radius-server key	设置 RADIUS 服务器的密钥；本命令的 no 操作为删除 RADIUS 服务器的密钥。

3. 配置 RADIUS Server

命令	解释
全局配置模式	
radius-server authentication host {<ipv4-address> <ipv6-address>} [port <port-number>][key {0 7} <string>] [primary] [access-mode {dot1x telnet}] no radius-server authentication host {<ipv4-address> <ipv6-address>}	设置 RADIUS 认证服务器 IPv4 地址或者 IPv6 地址和监听端口号、加密密钥、是否为主用服务器以及使用模式；本命令的 no 操作为删除 RADIUS 服务器。

radius-server accounting host {<ipv4-address> <ipv6-address>} [port <port-number>] [key {0 7} <string>] [primary] no radius-server accounting host {<ipv4-address> <ipv6-address>}	配置 RADIUS 计费服务器的 IPv4 或者 IPv6 地址和监听端口号、是否为主用服务器；本命令的 no 操作为删除 RADIUS 计费服务器。

4. 配置 RADIUS 服务参数

命令	解释
全局配置模式	
radius-server dead-time <minutes> no radius-server dead-time	配置 RADIUS 服务器 down 机后的恢复时间；本命令的 no 操作为恢复缺省配置。
radius-server retransmit <retries> no radius-server retransmit	配置 RADIUS 重传次数；本命令的 no 操作为恢复缺省配置。
radius-server timeout <seconds> no radius-server timeout	配置 RADIUS 服务器的超时定时器；本命令的 no 操作为恢复缺省配置。
radius-server accounting-interim-update timeout <seconds> no radius-server accounting-interim-update timeout	设置计费更新报文发送的时间间隔；本命令的 no 操作为恢复缺省配置。

5. 配置 RADIUS NAS-IP 地址

命令	解释
全局配置模式	
radius nas-ipv4 <ip-address> no radius nas-ipv4	设置交换设备发送 RADIUS 报文的 IP 源地址。
radius nas-ipv6 <ipv6-address> no radius nas-ipv6	设置交换设备发送 RADIUS 报文的 IPv6 源地址。

6.7.3 RADIUS 典型案例

6.7.3.1 IPv4 Radius 应用举例

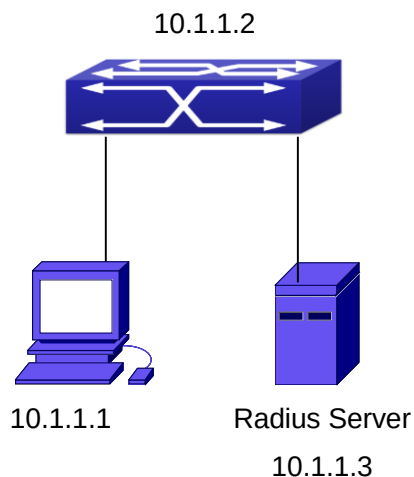


图 6-22 IEEE802.1x 配置举例拓扑图

交换机的 IP 地址设置为 10.1.1.2，并将除端口 **Ethernet1/2** 以外的任意一个端口与 RADIUS 认证服务器相连接，RADIUS 认证服务器的 IP 地址设置为 10.1.1.3，认证、计费端口为缺省端口 1812 和端口 1813。

配置步骤如下：

```
Switch(config)#interface vlan 1
Switch(Config-if-vlan1)#ip address 10.1.1.2 255.255.255.0
Switch(Config-if-vlan1)#exit
Switch(config)#radius-server authentication host 10.1.1.3
Switch(config)#radius-server accounting host 10.1.1.3
Switch(config)#radius-server key test
Switch(config)#aaa enable
Switch(config)#aaa-accounting enable
```

6.7.3.2 IPv6 Radius 应用举例

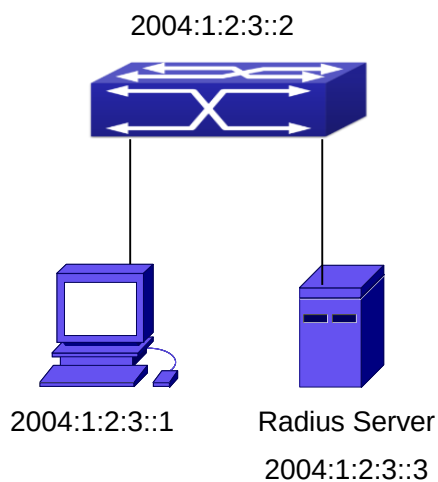


图 6-23 IPv6 Radius 配置举例拓扑图

交换机的 IP 地址设置为 2004:1:2:3::2, 并将除端口 1/2 以外的任意一个端口与 RADIUS 认证服务器相连接, RADIUS 认证服务器的 IP 地址设置为 2004:1:2:3::3, 认证、计费端口为缺省端口 1812 和端口 1813。

配置步骤如下:

```
Switch(config)#interface vlan 1
Switch(Config-if-vlan1)#ipv6 address 2004:1:2:3::2/64
Switch(Config-if-vlan1)#exit
Switch(config)#radius-server authentication host 2004:1:2:3::3
Switch(config)#radius-server accounting host 2004:1:2:3::3
Switch(config)#radius-server key test
Switch(config)#aaa enable
Switch(config)#aaa-accounting enable
```

6.7.4 RADIUS 排错帮助

在配置、使用 RADIUS 协议时, 可能会由于物理连接、配置错误等原因导致 RADIUS 认证失败。因此, 用户应注意以下要点:

- ☞ 首先应该保证到 RADIUS 服务器的物理连接的正确无误;
- ☞ 其次, 保证接口和链路协议是 UP (使用 show interface 命令);
- ☞ 再次, 保证交换机配置的 RADIUS 密钥同 RADIUS 服务器上配置的密钥一致;
- ☞ 然后, 确保配置了正确的 RADIUS 服务器。

如果使用检查都尝试仍无法解决 RADIUS 认证的问题, 那么请使用 debug aaa 等调试命令, 然后将 3 分钟内的 DEBUG 信息拷贝下来, 发送给本公司技术服务中心。

6.8 SSL

6.8.1 SSL 简介

随着计算机网络技术向整个经济社会各层次延伸, 整个社会表现对 Internet、Intranet、Extranet 等使用的更大的依赖性。随着企业间信息交互的不断增加, 任何一种网络应用和增值服务的使用程度将取决于所使用网络的信息安全有无保障, 网络安全已成为现代计算机网络应用的最大障碍, 也是急需解决的难题之一。

由于 Web 上有时要传输重要或敏感的数据, 因此 Netscape 公司在推出 Web 浏览器首版的同时, 提出了安全通信协议 SSL (Secure Socket Layer), 目前已有 2.0 和 3.0 版本。2.0 版本存在安全漏洞, 现在已经不使用了, 我们的交换机也不支持该版本。SSL 采用公开

密钥技术。其目标是保证两个应用间通信的保密性和可靠性,可在服务器和客户机两端同时实现支持。目前,利用公开密钥技术的 SSL 协议,并已成为 Internet 上保密通讯的工业标准。现行 Web 浏览器普遍将 HTTP 和 SSL 相结合,从而实现安全通信。

安全通信协议(SSL)是在 Internet 基础上提供的一种保证私密性的安全协议。它能使客户/服务器应用之间的通信不被攻击者窃听,并且始终对服务器进行认证,还可选择对客户进行认证。SSL 协议要求建立在可靠的传输层协议(例如: TCP)之上。SSL 协议的优势在于它是与应用层协议独立无关的。高层的应用层协议(例如: HTTP, FTP, TELNET.....)能透明的建立于 SSL 协议之上。SSL 协议在应用层协议通信之前就已经完成加密算法、通信密钥的协商以及服务器认证工作。在此之后应用层协议所传送的数据都会被加密,从而保证通信的私密性。

通过以上叙述, SSL 协议提供的安全信道有以下三个特性:

- ☞ 私密性。因为在握手协议定义了会话密钥后,所有的消息都被加密。
- ☞ 确认性。因为尽管会话的客户端认证是可选的,但是服务器端始终是被认证的。
- ☞ 可靠性。因为传送的消息包括消息完整性检查(使用MAC)。

6.8.1.1 SSL 基本原理

SSL 采用的基本的策略在两个通信程序之间提供一条在其间传递任意应用数据的安全通道。从理论上讲, SSL 连接非常类似“保密的” TCP 连接。协议栈中的 SSL 位置正好位于应用层的下面, TCP 的上面。SSL 假定其下层的数据包发送机制是可靠的。写入网络的数据将依顺序发送给另一端的程序,不会出现丢包或重复发送的情况。从理论上讲,有许多传输协议都能提供此种服务,但在实际应用中, SSL 几乎只是在 TCP 上运行,它不能在 UDP 或直接在 IP 上运行。

当在交换机上运行 web 功能时,当客户端通过浏览器访问我们的 Web 时,我们可以使用 SSL 功能,客户和交换机之间的通信通过 SSL 连接进行,这样很好地提高了访问的安全性。

首先我们在交换机上启动 SSL 功能,用户在客户端通过 https 访问交换机时,交换机就会和客户端进行 SSL 握手连接,形成安全的 SSL 连接通道,在此之后应用层协议所传送的数据都会被加密,从而保证通信的私密性。

使用 SSL 的时,首先进行 SSL 握手的过程,服务器端必须提供证书功能,目前我们使用的证书不是权威机构颁发的正规证书,而是 linux 下自己生成的证书,可能浏览器不能识别。考虑到我们使用 SSL 的环境,使用正规的证书没有必要,我们只要保证用户和交换机之间安全进行通信就可以了。目前我们不要求客户端证书验证功能。在握手的过程中还要协商通信中使用的密钥和加密算法,在交换应用数据的时候使用。

SSL 握手的基本过程:



6.8.2 SSL 配置任务序列

1. 启动/关闭 SSL 功能
2. 配置/删除 SSL 使用的端口号
3. 配置/删除 SSL 使用的加密套件
4. SSL 功能的维护与诊断

1. 启动/关闭 SSL 功能

命令	解释
全局配置模式	
ip http secure-server no ip http secure-server	启动/关闭 SSL 功能。

2. 配置/删除 SSL 使用的端口号

命令	解释
全局配置模式	
ip http secure-port <port-number> no ip http secure-port	配置 SSL 使用的端口号，no 命令删除 SSL 使用的端口号。

3. 配置/删除 SSL 加密套件

命令	解释
全局配置模式	

<pre>ip http secure-ciphersuite {des-cbc3-sha rc4-128-sha des-cbc-sha} no ip http secure-ciphersuite</pre>	配置/删除 SSL 使用的加密套件。

4. SSL 功能的维护与诊断

命令	解释
特权模式或者配置模式	
show ip http secure-server status	显示配置的 SSL 信息。
debug ssl no debug ssl	打开/关闭 SSL 功能的 DEBUG 显示。

6.8.3 SSL 典型案例

当在交换机上运行 web 功能时，当客户端通过浏览器访问我们的 Web 时，我们可以使用 SSL 功能，客户和交换机之间的通信通过 SSL 连接进行，这样很好地提高了访问的安全性。

首先我们在交换机上启动 SSL 功能，用户在客户端通过 https 访问交换机时，交换机就会和客户端进行 SSL 握手连接，形成安全的 SSL 连接通道，在此之后应用层协议所传送的数据都会被加密，从而保证通信的私密性。

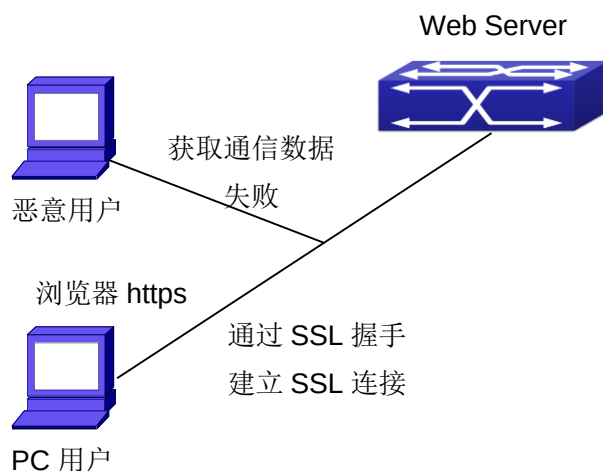


图 6-24 SSL 案例示意图

我们在交换机上进行如下配置：

```
Switch(config)#ip http secure-server
Switch(config)#ip http secure-port 1025
Switch(config)#ip http secure-ciphersuite rc4-128-sha
```

6.8.4 SSL 排错帮助

在配置、使用 SSL 功能时,可能会由于物理连接、配置错误等原因导致 SSL 功能异常。

因此,用户应注意以下要点:

- ☞ 首先应该保证物理连接的正确无误;
- ☞ 其次,保证接口和链路协议是 UP (使用 `show interface` 命令);
- ☞ 然后,确保开启了 SSL 功能(使用 `ip http secure-server` 命令);
- ☞ 如果配置了端口,不使用默认的端口号,注意输入网址的时候指明端口号;
- ☞ 如果先启动了 SSL 功能,再进行配置、取消端口号和加密套件操作时需要等到重启 SSL 功能后才生效;
- ☞ 使用 `des-cbc-sha` 加密套件时,必须 ie7.0 以上的浏览器才能支持;
- ☞ 如果尝试使用以上检查仍无法解决 SSL 的问题,那么请使用 `debug ssl` 等调试命令,然后将 3 分钟内的 DEBUG 信息拷贝下来,发送给本公司技术服务中心。

6.9 VLAN-ACL

6.9.1 VLAN-ACL 功能简介

用户通过对 VLAN 配置 ACL 策略,从而实现对 VLAN 内所有端口的访问控制,VLAN-ACL 使用户能够更加方便地管理网络。用户只需在 VLAN 下配置 ACL 策略,相应的 ACL 动作就能在 VLAN 的所有成员端口生效,而无需在每个成员端口上单独配置。

当 VLAN ACL 与 Port ACL 同时配置时,即需要满足 **deny** 优先,当报文同时匹配 VLAN ACL 和端口 ACL 时,只要有一个动作是 **drop**,则结果就是 **drop**。

Egress ACL 可以在出口和入口方向实现对报文的过滤,符合特定规则的报文可以允许通过或者禁止通过。ACL 可以支持 IP ACL, MAC ACL, MAC-IP ACL, IPv6 ACL。在 VLAN 入口方向上可以同时配置四种 ACL;在 VLAN 出口方向上由于硬件只有四个资源,IP 和 MAC ACL 各占一个资源,MAC-IP 和 IPv6 ACL 各占两个资源,当硬件资源占满后则无法再配置,即一个 VLAN 在出口方向上不能同时配置四种 ACL,同时配置三种 ACL 时也只能是 IP、MAC 和 MAC-IP 类型同时配置或 IP、MAC 和 IPv6 类型同时配置,同时配置两种 ACL 时可以任意选取,没有限制。一种类型的 ACL 在 VLAN 上只能配置一条。

6.9.2 VLAN-ACL 功能配置任务序列

1. 配置 IP 类型的 VLAN-ACL
2. 配置 MAC 类型的 VLAN-ACL
3. 配置 MAC-IP 类型的 VLAN-ACL
4. 配置 IPv6 类型的 VLAN-ACL
5. 显示 VLAN-ACL 的配置及统计信息
6. 清除 VLAN-ACL 的统计信息

1. 配置 IP 类型的 VLAN-ACL

命令	解释
全局配置模式	
vacl ip access-group {<1-299> WORD} {in out} [traffic-statistic] vlan WORD no vael ip access-group {<1-299> WORD}{in out} vlan WORD	配置或删除 IP 类型的 VLAN-ACL。

2. 配置 MAC 类型的 VLAN-ACL

命令	解释
全局配置模式	
vacl mac access-group {<700-1199> WORD} {in out} [traffic-statistic] vlan WORD no vael mac access-group {<700-1199> WORD} {in out} vlan WORD	配置或删除 MAC 类型的 VLAN-ACL。

3. 配置 MAC-IP 类型的 VLAN-ACL

命令	解释
全局配置模式	
vacl mac-ip access-group {<3100-3299> WORD} {in out} [traffic-statistic] vlan WORD no vael mac-ip access-group {<3100-3299> WORD} {in out} vlan WORD	配置或删除 MAC-IP 类型的 VLAN-ACL。

4. 配置 IPv6 类型的 VLAN-ACL

命令	解释
全局配置模式	
vacl ipv6 access-group (<500-699> WORD) {in out} (traffic-statistic) vlan WORD no ipv6 access-group {<500-699> WORD} {in out} vlan WORD	配置或删除 IPv6 类型的 VLAN-ACL。

5. 显示 VLAN-ACL 的配置及统计信息

命令	解释
特权配置模式	
show vacl [in out] vlan [<vlan-id>]	显示 VACL 的配置及统计信息。

6. 清除 VLAN-ACL 的统计信息

命令	解释
特权配置模式	
clear vacl [in out] statistic vlan [<vlan-id>]	清除 VACL 的统计信息。

6.9.3 VLAN-ACL 功能配置案例

某公司的网络配置如下，将不同部分划分到不同 VLAN，技术部为 Vlan1、财务部为 Vlan2，技术部门要求在每天的休息时间可以访问外网，而财务部出于安全性的考虑在任何时间段均不允许访问外部网络。则可以配置如下策略：

- ☞ 为技术部定制 ACL 策略 VACL_A，在休息时间可以访问外网，规则为 Permit，其他时间段均为 Deny，并应用策略到 Vlan1 上。
- ☞ 为财务部定制 ACL 策略 VACL_B，在任何时间均不允许访问外网，但可以不受限制的访问内网资源，并应用策略到 Vlan2 上。

组网环境如下：

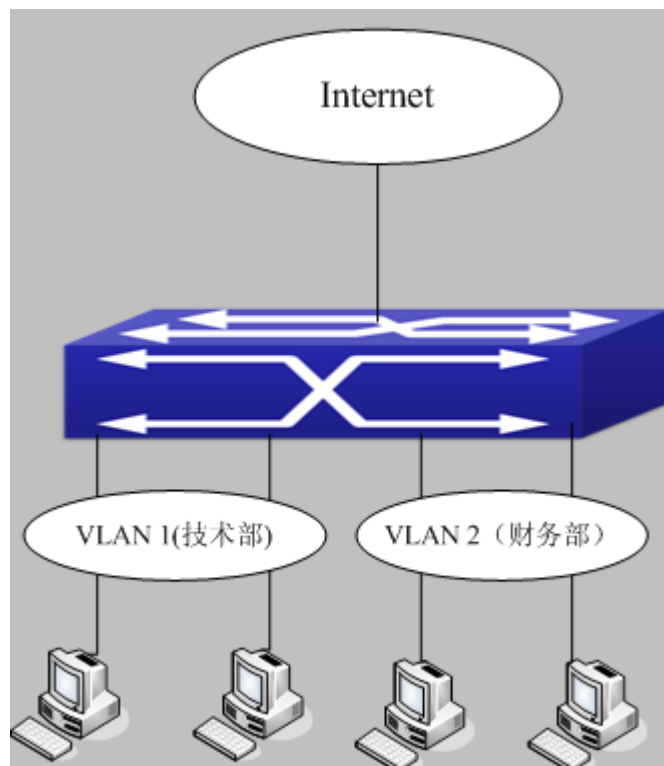


图 6-25 VLAN-ACL 配置案例

配置举例：

1) 先配置一个 timerange，有效时间为工作日的上班时间。

```
Switch(config)#time-range t1
```

```
Switch(config-time-range-t1)#periodic weekdays 9:00:00 to 12:00:00
```

```
Switch(config-time-range-t1)#periodic weekdays 13:00:00 to 18:00:00
```

2) 配置 IP 的扩展 acl_a，上班时间仅允许访问同一网段（如 192.168.0.255）内的主机。

```
Switch(config)# ip access-list extended vacl_a
```

```
Switch(config-ip-ext-nacl-vacl_a)# permit ip any-source 192.168.0.0 0.0.0.255 time-range  
t1
```

```
Switch(config-ip-ext-nacl-vacl_a)# deny ip any-source any-destination time-range t1
```

3) 配置 IP 的扩展 acl_b，任何时间仅允许访问同一网段（如 192.168.1.255）内的主机。

```
Switch(config)#ip access-list extended vacl_b
```

```
Switch(config-ip-ext-nacl-vacl_a)# permit ip any-source 192.168.1.0 0.0.0.255
```

```
Switch(config-ip-ext-nacl-vacl_a)# deny ip any-source any-destination
```

4) 将配置应用到 VLAN。

```
Switch(config)#vacl ip access-group vacl_a in vlan 1
```

```
Switch(config)#vacl ip access-group vacl_b in vlan 2
```

6.9.4 VLAN-ACL 排错帮助

- ☞ 当 VLAN ACL 与 Port ACL 同时配置时，即需要满足 deny 优先，当报文同时匹配 VLAN ACL 和端口 ACL 时，只要有一个动作是 drop，则结果就是 drop。
- ☞ 每一个不同类型的 ACL 在一个 VLAN 上仅能配置一个，如：基本的 IP ACL，在每个 VLAN 上仅能够应用一个。

6.10 PPPoE 中间代理

6.10.1 PPPoE Intermediate Agent 介绍

6.10.1.1 PPPoE 简介

PPPoE 协议，即 Point to Point Protocol over Ethernet，可以看出 PPPoE 协议是把 PPP 协议应用到 Ethernet 上的一种协议。PPP 协议是一种链路层协议，提供了一种点对点的链路层通信方式。PPP 通常是住宅主机拨号链路中所选择的协议，例如链路是电话线拨号。

把 PPP 协议应用到 Ethernet 上, 即 PPPoE 协议可以使 Ethernet 上的多个主机通过一个或多个简单的桥接设备连接到一个远端的接入集中器上, 如果这个远端的接入集中器是宽带接入服务器 (BAS), 便可以提供这些主机宽带接入和计费等功能。所以 PPPoE 协议经常被用作 Ethernet 上的宽带接入认证方式。

6.10.1.2 PPPoE IA 介绍

随着宽带接入技术的快速发展, 宽带接入网络发展的越来越大, 但是随着宽带接入网络的发展, 安全问题逐渐变为公众焦点。在当前接入网络环境下, 无论是客户还是接入设备和网络都面临着威胁, 尤其是那些来自用户端的威胁。传统以太网的用户不能被精确的识别, 追踪以及定位, 但是在一个开放的可管理的面向公众的网络, 用户身份识别和定位是最基本的特征和要求, 例如在提供用户账号登陆的应用时, 这种方式能够有效的防止用户账号被盗用, 而 PPPoE Intermediate Agent 正提供了这种功能。

PPPoE 协议的工作过程主要分为两个阶段, 即发现阶段和会话阶段。发现阶段的作用是得到远端服务器的 Mac 地址, 以便建立一个点对点的链接, 并与服务器建立一个会话标识 (SESSION ID), 在会话阶段使用这个 SESSION ID 进行通信。由于 PPPoE Intermediate Agent 只与发现阶段相关, 所以这里只对发现阶段做简要介绍。

PPPoE 协议发现阶段主要分为四步:

1. 主机 (client) 端发送 PADI 报文: PPPoE 发现阶段的第一步, 由客户端首先以广播地址为目的地址, 广播这个 PADI (PPPoE Active Discovery Initiation) 报文, 用以发现二层网络中的接入集中器, 注意此消息有可能发送给网络中的多个接入集中器。
2. 接入集中器 (server) 端回应 PADO 报文: PPPoE 发现阶段的第二步, 由服务器端根据接收到的 PADI 报文的源 Mac 地址, 向主机端回应 PADO (PPPoE Active Discovery Offer) 报文, 该报文中会携带接入集中器名和服务名等信息。
3. 主机 (client) 端发送 PADR 报文: PPPoE 发现阶段的第三步, 由客户端根据接收到的 PADO 报文, 选择一个接入集中器作为将要进行会话的接入集中器, 因为在第一步中的 PADI 消息可能发送给网络中的多个接入集中器, 所以可能收到多个 PADO 报文, 而从这多个接入集中器中进行选择的依据则是主机需要的服务等信息和 PADO 报文中所携带的服务等信息的匹配结果。当选择完接入集中器后便知道将要进行会话的另一端的 Mac 地址, 便向其发送 PADR (PPPoE Active Discovery Request) 报文, 用于告诉接入集中器有主机进行会话请求。
4. 接入集中器 (server) 端回应 PADS 报文: PPPoE 发现阶段的第四步, 由服务器端根据接收到的 PADR 报文, 知道有主机端有会话请求, 便创建一个 SESSION ID, 并把这个 SESSION ID 通过 PADS (PPPoE Active Discovery Session-confirmation) 报文发送给客户端。至此 PPPoE 发现阶段已经完成, 因为客户端和服务端都有了 SESSION ID 可以进入会话阶段了。

PPPoE 有一种特殊的报文, 即 PADT (PPPoE Active Discovery Terminate) 报文, 它和上述四种报文具有相同的以太网协议号, 即 0x8863, 所以它也可以看作是发现阶段的一种报文, 但是 PADT 报文可能在会话进行开始之后的任意时间内被发送, 主要是用来终止一个 PPPoE 会话的。它可以由主机或接入集中器发送。

PPPoE Intermediate Agent，它提供了一种用户身份识别和定位的功能。其原理是，在 PPPoE 发现阶段，把从客户端主机发出的 PADI 和 PADR 消息，在经过网络接入设备时，打上该设备的接入链路的标识，这样在接入集中器（服务器端，通常是 BAS）上便可以精确的识别该用户的身份以及定位该用户的位置了。

如果直连的接入设备是 LAN Switch，则加入的信息包括：接入交换机的 Mac，Slot ID，Port Index，Vlan ID 等。本功能主要是依据 DSL Forum Technical Report 101，即 Migration to Ethernet-based DSL aggregation 来实现。

6.10.1.2.1 PPPoE Intermediate Agent 协议交互过程

PPPoE Intermediate Agent 的协议交互过程和 PPPoE 的协议交互过程类似，只是在与 Host 端直连的交换机（Intermediate Agent）上对 PADI 和 PADR 报文加入了接入链路标识而已。具体交互过程如下图所示。

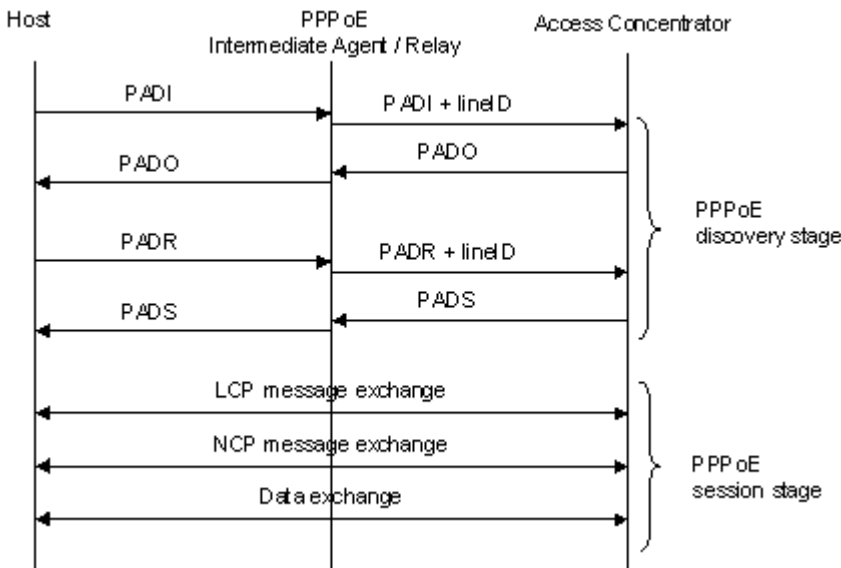


图 6-26 PPPoE IA 协议交互过程

6.10.1.2.2 PPPoE 报文格式

PPPoE 报文格式如下：

Ethernet II 帧结构

目的 MAC	源 MAC	类型域	PPPOE 数据	CRC 校验和

PPPOE 数据

版本	类型	代码	会话 ID	长度域	TLV 1		TLV N

TLV 结构

类型	长度	数据

其中各字段含义如下：

Ethernet II 帧的类型域(2byte)：协议规定 PPPoE 协议报文的类型域取值为 0x8863，仅包括 PPPoE 发现阶段的 5 种报文。会话阶段的报文类型域取值为 0x8864。

PPPoE 版本域(4bit)：指明当前 PPPoE 协议版本，规定当前版本必须设置为 0x1

PPPoE 类型域(4bit)：指明协议类型，规定当前版本必须设置为 0x1

PPPoE 代码域(1byte)：指明发送 PPPoE 的报文类型。0x09 代表 PADI 报文，0x07 代表 PADO 报文，0x19 代表 PADR 报文，0x65 代表 PADS 报文，0xa7 代表 PADT 报文。

PPPoE 会话 ID 域(2byte)：指明会话标识

PPPoE 长度域(2byte)：指明所有 TLV 长度和

TLV 类型域(2byte)：一个 TLV 结构代表一个 TAG，类型域则代表 TAG 类型，如下表所示

TLV 长度域(2byte)：指明该 TAG 的数据域的长度。

TLV 数据域(长度不定)：指明该 TAG 的传输的数据

标记类型	标记说明
0x0000	表示PPPOE报文数据域中一串标记的结束，为了保证版本的兼容性而保留，在有些报文中有应用。
0x0101	服务名，主要用来表明网络侧所能提供给用户的一些服务。
0x0102	访问集中器名，当用户侧接收到了AC的响应的PADO报文时，就可获从所携带的标记中获知访问集中器的名字，而且还可以据此来选择相应的访问集中器。
0x0103	主机唯一标识，类似于PPP数据报文中的标识域，主要是用来匹配发送和接收端的，因为对于广播式的网络中会同时存在很多个PPPOE的数据报文。
0x0104	AC-Cookies，主要被用来防止恶意性DOS攻击。
0x0105	销售商的标识符。
0x0110	中继会话ID，对于PPPOE的数据报文也同样可以像DHCP报文一样被中断到另外的AC上终结，这个字段则是用来维护另一个连接的。
0x0201	服务名错误，当请求的服务名不被对端所接受时，会在响应的报文中携带这个标记。
0x0202	访问集中器名出错。
0x0203	一般性错误。

图 6-27 PPPoE 中 TAG 取值类型

6.10.1.2.3 PPPoE Intermediate Agent vendor tag 结构

下图是 PPPoE IA 最主要的功能，即 PPPoE IA 所要添加的链路标识的结构。

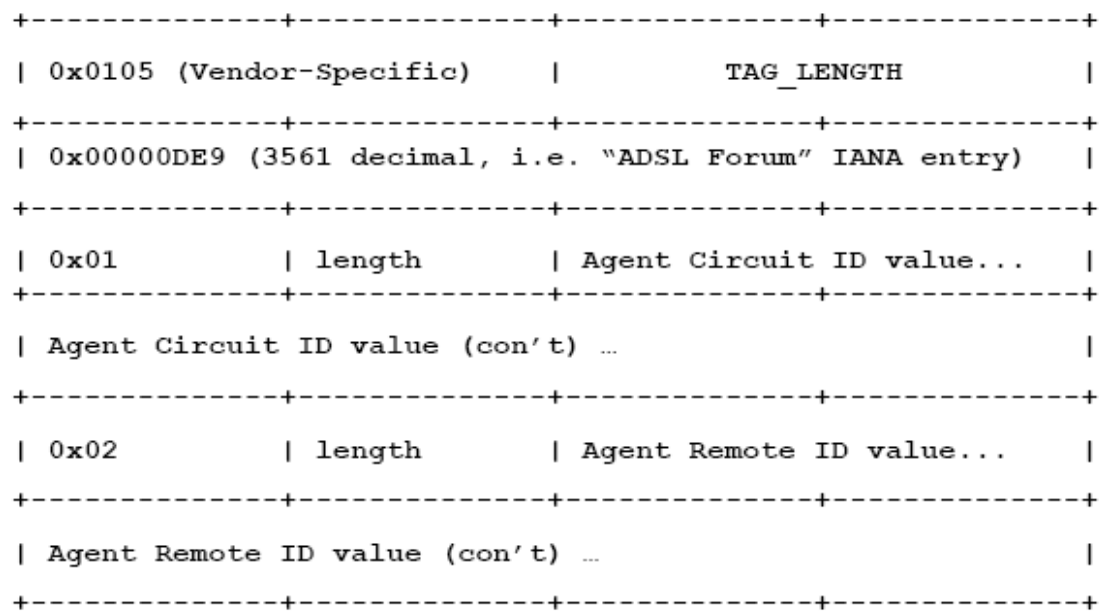


图 6-28 PPPoE IA - vendor tag（每行 4 个字节）

PPPoE IA 添加 tag 的 TLV 类型标识为 0x0105，TAG_LENGTH 是 vendor tag 的长度域； 0x00000DE9 为固定 4 个字节的“ADSL Forum” IANA entry； 0x01 是 Agent Circuit ID 的类型域，length 是其长度域，和 Agent circuit ID value 域； 0x02 是 Agent Remote ID 的类型域，length 是其长度域，和 Agent Remote ID value 域。

PPPoE IA 提供一个默认生成的 circuit ID value，默认生成的 circuit ID 如图所示，其主要包括 5 个字段，其中 ANI（Access Node Identifier）是用户可以配置的，其长度小于 47 个字节；如果没有配置则系统默认为接入交换机的 Mac,占 6 个字节，后面的字段用空格符分隔，“eth”占 3 个字节，和后面的字段用空格符分隔，Slot ID 占 2 个字节，用分隔符“/”分隔且占 1 个字节，Port Index 占 3 个字节，用分隔符“:”分隔且占 1 个字节，Vlan ID 占 4 个字节,这里的所有字段均用 ASCII 码表示。用户也可以根据需求,为每个端口单独配置 circuit id。

ANI (n byte)	Space (1byte)	eth (3 byte)	Space (1byte)	Slot ID (2byte)	/ (1byte)	Port Index (3byte)	: (1byte)	Vlan ID (4byte)

图 11-3 Agent Circuit ID value

PPPoE IA 默认生成的 remote ID value 是接入交换机的 Mac。对于 remote ID value 用户也可以灵活配置，其长度小于 63 个字节。

6.10.1.2.4 PPPoE Intermediate Agent 的信任端口

PPPoE 发现阶段发送有五种报文，其中 PADI 和 PADR 是由客户端向服务器发送的报文，而 PADO 和 PADS 是由服务器向客户端发送的报文，PADT 报文既可以由服务器发送

到客户端，也可以从客户端发送到服务器。

在 PPPoE IA 中，为了安全和减少通讯流量，把连接服务器方向的端口设置为信任端口，把连接客户端的端口设置为非信任端口。对于信任端口可以接收所有报文，对于非信任端口只能接收从客户端发往服务器的 PADI、PADR 和 PADT 报文。为了保证客户端的运行正确，必须把可以连接到服务器的端口设置成信任端口，并且每个接入设备中至少有一个信任端口。

由于 PPPoE IA vendor tag 不能存在于从服务器发往客户端的 PPPoE 报文中，所以这些报文中如果存在 PPPoE IA vendor tag，我们可以选择剥离这些 vendor tag 后转发出去。剥离功能必须在信任端口上配置，非信任端口上也可以开启 vendor tag 剥离功能，但是不起作用。

6.10.2 PPPoE Intermediate Agent 配置任务序列

1. 配置全局开启 PPPoE Intermediate Agent
2. 配置端口开启 PPPoE Intermediate Agent

命令	解释
全局配置模式	
pppoe intermediate-agent no pppoe intermediate-agent	开启全局 PPPoE intermediate agent 功能。
pppoe intermediate-agent type tr-101 circuit-id access-node-id <string> no pppoe intermediate-agent type tr-101 circuit-id access-node-id	配置添加的 vendor tag 中 circuit ID 的 access node id 域的值。
pppoe intermediate-agent type tr-101 circuit-id identifier-string <string> option {sp sv pv spv} delimiter <WORD> [delimiter <WORD>] no pppoe intermediate-agent type tr-101 circuit-id identifier-string option delimiter	配置添加的 vendor tag 标识中的 circuit-id。
pppoe intermediate-agent type self-defined circuit-id {vlan port id (switch-id (mac hostname) remote-mac) string WORD} no pppoe intermediate-agent type self-defined circuit-id	配置自定义的 circuit-id 形式。

pppoe intermediate-agent type self-defined remoteid {mac vlan-mac [hostname string WORD} no pppoe intermediate-agent type self-defined remote-id	配置自定义的 remote-id 形式。
pppoe intermediate-agent delimiter <WORD> no pppoe intermediate-agent delimiter	配置 circuit-id 和 remote-id 各个字段间的分隔符。
pppoe intermediate-agent format (circuit-id remote-id) (hex ascii) no pppoe intermediate-agent format (circuit-id remote-id)	配置 circuit-id 和 remote-id 的表示形式。
端口配置模式	
pppoe intermediate-agent no pppoe intermediate-agent	开启端口 PPPoE intermediate agent 功能。
pppoe intermediate-agent vendor-tag strip no pppoe intermediate-agent vendor-tag strip	设置端口的 vendor tag 剥离功能。
pppoe intermediate-agent trust no pppoe intermediate-agent trust	设置端口为信任端口。
pppoe intermediate-agent circuit-id <string> no pppoe intermediate-agent circuit-id	设置端口的 circuit-id。
pppoe intermediate-agent remote-id <string> no pppoe intermediate-agent remote-id	设置端口的 remote-id。

6.10.3 PPPoE Intermediate Agent 典型应用

PPPoE IA 的典型应用场景如下：

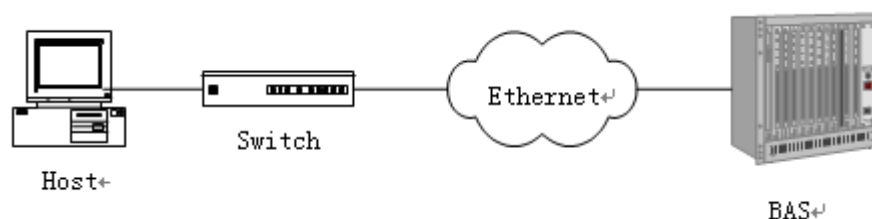


图 6-29 PPPoE IA 典型应用场景

其中 Host 主机和 BAS 服务器端都运行 PPPoE 协议，它们通过二层以太网相连，在与 Host 端直连的交换机开启 PPPoE Intermediate Agent 功能。

典型配置（1）如下：

第一步：在交换机(Switch)上开启全局 PPPoE IA 功能，交换机的 Mac 为 0a0b0c0d0e0f。

Switch (config)#pppoe intermediate-agent

第二步：配置连接服务器的端口 ethernet1/1 为 trust 端口，并且配置 vendor tag 剥离功能。

Switch(config-if-ethernet1/1)#pppoe intermediate-agent trust

Switch(config-if-ethernet1/1)#pppoe intermediate-agent vendor-tag strip

第三步：在 vlan 1 的端口 ethernet1/2 和 vlan 1234 的端口 ethernet1/3 上开启端口 PPPoE IA 功能。

Switch(config-if-ethernet1/2)#pppoe intermediate-agent

Switch(config-if-ethernet1/3)#pppoe intermediate-agent

第四步：配置 pppoe intermediate-agent access-node-id 为 abcd。

Switch (config)#pppoe intermediate-agent type tr-101 circuit-id access-node-id abcd

第五步：在端口 ethernet1/3 上配置 circuit id 为 aaaa，remote id 为 xyz。

Switch(config-if-ethernet1/3)#pppoe intermediate-agent circuit-id aaaa

Switch(config-if-ethernet1/3)#pppoe intermediate-agent remote-id xyz

对于端口 ethernet1/2 的 PPPoE 报文添加的 vendor tag 标识中 circuit-id value 为"abcd eth 01/002:0001"，remote-id value 为"0a0b0c0d0e0f"。

对于端口 ethernet1/3 的 PPPoE 报文添加的 vendor tag 标识中 circuit-id value 为"aaaa"，remote-id value 为"xyz"。

典型配置（2）如下：

第一步：在交换机(Switch)上开启全局 PPPoE IA 功能。交换机的 Mac 为 0a0b0c0d0e0f。

Switch(config)# pppoe intermediate-agent

第二步：配置连接服务器的端口 ethernet1/1 为 trust 端口，并且配置 vendor tag 剥离功能。

Switch(config-if-ethernet1/1)#pppoe intermediate-agent trust

Switch(config-if-ethernet1/1)#pppoe intermediate-agent vendor-tag strip

第三步：在 vlan 1 的端口 ethernet1/2 和 vlan 1234 的端口 ethernet1/3 上开启端口 PPPoE IA 功能。

Switch(config-if-ethernet1/2)#pppoe intermediate-agent

Switch(config-if-ethernet1/3)#pppoe intermediate-agent

第四步：配置 pppoe intermediate-agent access-node-id 为 abcd

Switch(config)#pppoe intermediate-agent type tr-101 circuit-id access-node-id abcd

第五步：配置 pppoe intermediate-agent identifier-string 为"efgh"，组合模式为 spv，Slot ID

和 Port ID 的分隔符为"#", Port ID 和 Vlan ID 的分隔符为"/"。

```
Switch(config)#pppoe intermediate-agent type tr-101 circuit-id identifier-string efgh option  
spv delimiter # delimiter /
```

第六步：在端口 ethernet1/2 上配置 circuit-id value 为 bbbb。

```
Switch (config-if-ethernet1/2)#pppoe intermediate-agent circuit-id bbbb
```

第七步：在端口 ethernet1/3 上配置 remote id 为 xyz。

```
Switch (config-if-ethernet1/3)#pppoe intermediate-agent remote-id xyz
```

对于端口 ethernet1/2 的 PPPoE 报文添加的 vendor tag 标识中 circuit-id value 为"bbbb", remote-id value 为"0a0b0c0d0e0f"。

对于端口 ethernet1/3 的 PPPoE 报文添加的 vendor tag 标识中 circuit-id value 为"efgh eth 01#003/1234", remote-id value 为"xyz"。

6.10.4 PPPoE Intermediate Agent 排错帮助

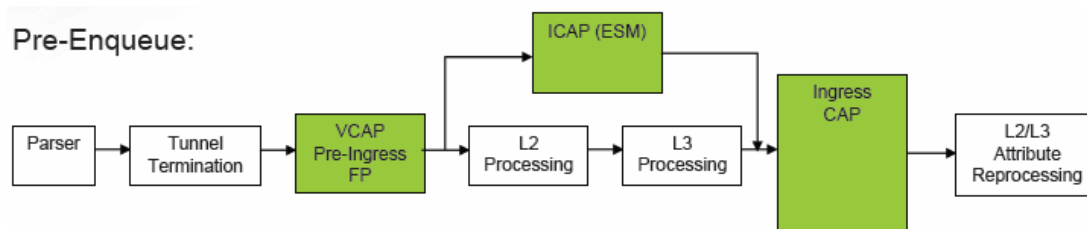
- ☞ 如果要在交换机端口上运行 pppoe intermediate-agent, 首先必须在全局打开 pppoe intermediate-agent 开关。在没有打开全局 pppoe intermediate-agent 开关之前, 打开端口的 pppoe intermediate-agent 开关是无效的。
- ☞ 必须至少配置一个信任端口, 并且该端口能连接到服务器。
- ☞ Vendor tag strip 功能必须在信任端口上配置。
- ☞ Circuit-id 的覆盖优先级顺序为: 优先级最高的是端口配置 pppoe intermediate-agent circuit-id, 优先级次高为 pppoe intermediate-agent identifier-string option delimiter, 优先级最低为 pppoe intermediate-agent access-node-id。

6.11 EIFP

6.11.1 EIFP 介绍

6.11.1.1 外扩 TCAM 介绍

BCM56620 支持外扩的 TCAM, 它作为 IFP 的第 17 个 Slice, 与其它 slice 的优先级关系是通过虚拟 slice 位图来确定的, 其大小为 36Mbit。它的位置与 IFP 的内部 slice 不一样, 它与 L2、L3 匹配 (lookup) 平行, 在 IFP 的内部 slice 匹配之前, 所以 L2、L3 匹配的结果不在 EIFP 的 key 中。IFP 的外部 slice 由 ESM 块管理, 其大小和划分由 ESM 的配置决定。其在转发流程中的位置如下图所示:



图中 ESM 为外扩 TCAM 部分。像 IFP 的内部 slice 一样，外扩 TCAM 只能产生一个结果，产生的结果与 IFP 内部 slice 的结果由决策引擎裁决。

EIFP 可以在入口方向实现对报文的过滤，可以实现 ACL、QoS 等。目前 EIFP 只能使用 350MHz，支持 ip acl, mac acl, ipv6 acl 和相应的 QoS。目前仅 5662x 芯片支持 EIFP，即 kylin 卡支持 EIFP。

6.11.1.2 外扩 ACL 介绍

External Ingress acl 可以根据指定的匹配规则在入口方向决定如何处理报文。External Ingress acl 可以绑定到端口上或者 VLAN 接口上。

External Ingress acl 采用 EIFP 实现，EIFP 相当于 1 个 slice，其有 32 个 block（我们使用的是 NL56617 类型的 TCAM），每个 block 16K 个 entrys，每个 entry 72bit，一个 block 只对应一种模板。对于 144bit 的模板，一个 block 可以放 8K 个规则，对于 288bit 的模板，可以放 4K 个规则。整个 EIFP 只能得出一个匹配表项。

EIFP 根据哪一模板生成 key，由三个因素决定：Tcam_type、数据包类型和端口的配置寄存器的值。根据目前的 EIFP 模板类型（目前只支持 350MHz，只有 5 种模板），可以支持 ip acl、mac acl、ipv6 acl 三种。这三种 acl 使用的 acl 号为：ip<11000-12999>，mac<14000-15999>，ipv6<18000-19999>，由 ACL 规则号决定使用哪个模板。目前每种报文使用的模板类型是固定的，对于 IPv4 报文，其使用 L3 IPv4 模板（包括 144bit 和 288bit），非 IP 二层报文使用 Mac 模板（包括 144bit 和 288bit），IPv6 报文使用 IPv6 模板。

三种 ACL 中，ip acl 和 mac acl 又分为两个模板，其 key 的长度分别为 144bit 和 288bit。由于每个端口都对应一个配置寄存器，该寄存器决定了从该端口进入的数据包按照哪种模板生成 key，考虑到下发某条 VACL 时，还要设置该 vlan 对应的端口的寄存器，使端口按照 VACL 的模板生成 key，在实际使用时，默认情况下 IP acl 和 mac acl 使用 288bit 宽的模板，在特殊情况下（比如评测时）使用 144bit 的模板，由于目前 ipv6 只有一种 144bit 宽的模板，所以其一直使用 144bit 的模板。

EIACL 不支持端口位图，所以如果一个 ACL 需要应用于多个端口时，需要为每个端口下发一次规则。

6.11.1.3 外扩 QoS 介绍

EIFP 使用内部 Slice 的 meter，可以产生三色报文，而且 EIFP 中的 Policy 与内部 FP 中的 Policy 基本一样，所以只要可以下到 EIFP 上的 ACL，都可以对匹配该 ACL 的报文使用 QoS。

EIFP 虽然可以产生三色报文，但每个 entry 只有一个 counter，如何使用该 counter 可以通过寄存器来设定。Counter 的使用可以通过在 policy map 的 match 时设定模式。

6.11.2 EIFP 配置

6.11.2.1 EACL 配置序列

1. 配置 access-list
 - (1) 配置数字 IP 访问列表
 - (2) 配置 MAC 访问列表
 - (3) 配置数字 IPV6 访问列表
2. 配置包过滤功能
 - (1) 全局打开包过滤功能
3. 配置时间范围功能
 - (1) 创建时间范围名称
 - (2) 配置周期性时段
 - (3) 配置绝对性时段
4. 将 access-list 绑定到特定端口或 vlan 的 in 方向
5. 显示端口或 VLAN ACL 的配置及统计信息
6. 清空指定接口或 VLAN 的包过滤统计信息

1. 配置 access-list
 - (1) 配置数字 IP 访问列表

命令	解释
全局配置模式	
access-list {<11000-12999>} {deny permit} tcp {<sipAddr> <sipMASK> } any-source {host-source <sipAddr>}} [s-port {<port> range <sPortMin> <sPortMax>}] {<dipAddr> <dipMASK> } any-destination {host-destination <dipAddr>}} [d-port {<port> range <dPortMin> <dPortMax>}] [ack+ fin+ psh+ rst+ urg+ syn] [precedence <prec>] [tos <tos>] [time-range <time-range-name>]	创建一条 TCP 数字 IP 访问列表。

access-list {<11000-12999>} {deny permit} udp {{<sipAddr> <sipMASK>} any-source {host-source <sipAddr>}} [s-port {<port> range <sPortMin> <sPortMax>}] {{<dipAddr> <dipMASK>} any-destination {host-destination <dipAddr>}} [d-port {<port> range <dPortMin> <dPortMax>}] [precedence <prec>] [tos <tos>] [time-range <time-range-name>]	创建一条 UDP 数字 IP 访问列表。
access-list {<11000-12999>} {deny permit} icmp {{<sipAddr> <sipMASK>} any-source {host-source <sipAddr>}} {{<dipAddr> <dipMASK>} any-destination {host-destination <dipAddr>}} [<icmp-type> [<icmp-code>]] [precedence <prec>] [tos <tos>] [time-range <time-range-name>]	创建一条 ICMP 数字 IP 访问列表。
access-list {<11000-12999>} {deny permit} igmp {{<sipAddr> <sipMASK>} any-source {host-source <sipAddr>}} {{<dipAddr> <dipMASK>} any-destination {host-destination <dipAddr>}} [<igmp-type>] [precedence <prec>] [tos <tos>] [time-range <time-range-name>]	创建一条 IGMP 数字 IP 访问列表。
access-list {<11000-12999>} {deny permit} { eigrp gre igrp ipinip ip ospf <protocol-num>} {{<sipAddr> <sipMASK>} any-source {host-source <sipAddr>}} {{<dipAddr> <dipMASK>} any-destination {host-destination <dipAddr>}} [precedence <prec>] [tos <tos>] [time-range <time-range-name>]	创建一条其他特定协议或所有 IP 协议的数字 IP 访问列表。
no access-list <11000-12999>	删除一条数字 IP 访问列表。

(2) 配置数字 MAC 访问列表

命令	解释
全局配置模式	

access-list {<14000-15999>} {deny permit} {any-source-mac {host-source-mac <source-mac>} {<source-mac> <source-mask>}} {any-destination-mac {host-destination-mac <host-dmac>} {<dmac> <dmac-mask>}} {untagged-eth2 tagged-eth2 untagged-802-3 tagged-802-3} [ethertype <ethertype> <ethertypemask>]	创建一条数字 MAC 访问列表。
no access-list {<num>}	删除一条数字 MAC 访问列表。

(3) 配置数字 IPv6 访问列表

命令	解释
全局配置模式	
ipv6 access-list {<18000-19999>} {deny permit} {any-source <sIPv6Prefix/sPrefixlen> {host-source <source-lower64> } {any-destination {host-destination- <host-lower64>} {<dIPv6Prefix/dPrefixlen>}} [time-range <time-range-name>	创建一条数字 IPv6 访问列表。
no ipv6 access-list {<num>}	删除一条数字 IPv6 访问列表。

2. 配置包过滤功能

(1) 打开/关闭全局包过滤功能

命令	解释
全局配置模式	
firewall enable	全局打开包过滤功能。
firewall disable	全局关闭包过滤功能。

3. 配置时间范围功能

(1) 创建时间范围名称

命令	解释
全局配置模式	
time-range <time_range_name>	创建一个名为 time_range_name 的时间范围名。
no time-range <time_range_name>	停止名为 time_range_name 的时间范围功能。

(2) 配置周期性时段

命令	解释
时间范围模式	
absolute-periodic {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <start_time> to {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <end_time>	配置一周内的各种不同要求的时间范围，每周都循环这个时间。
periodic {{Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+ Sunday} daily weekdays weekend} <start_time> to <end_time>	
no absolute-periodic {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <start_time> to {Monday Tuesday Wednesday Thursday Friday Saturday Sunday} <end_time>	取消一周内的时间范围配置。
no periodic {{Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+ Sunday} daily weekdays weekend} <start_time> to <end_time>	

(3) 配置绝对性时段

命令	解释
全局配置模式	
absolute start <start_time> <start_data> [end <end_time> <end_data>]	创建一个绝对时间范围。
no absolute start <start_time> <start_data> [end <end_time> <end_data>]	取消一个绝对时间范围功能。

4. 将 access-list 绑定到端口或 VLAN 的入口方向

(1) 将 access-list 绑定到端口入方向

命令	解释
端口配置模式	

ip access-group {<11000-12999>} in [traffic-statistic] no ip access-group {<11000-12999>} in	在端口的入口方向上应用一条 ip access-list; no 操作为删除绑定在端口上的 access-list。
mac access-group {<14000-15999>} in [traffic-statistic] no mac access-group {<14000-15999>} in	在端口的入口方向上应用一条 mac access-list; no 操作为删除绑定在端口上的 access-list。
ipv6 access-group {<18000-19999>} in [traffic-statistic] no ipv6 access-group {<18000-19999>} in	在端口的入口方向上应用一条 ipv6 access-list; no 操作为删除绑定在端口上的 access-list。

(2) 将 access-list 绑定到 VLAN 入方向

命令	解释
全局配置模式	
vacl ip access-group {<11000-12999>} in [traffic-statistic] vlan WORD no vacl ip access-group {<11000-12999>} in vlan WORD	在 VLAN 的入口方向上应用一条 IP access-list; no 操作为删除绑定在 VLAN 上的 access-list。
vacl mac access-group {<14000-15999>} in [traffic-statistic] vlan WORD no vacl mac access-group {<14000-15999>} in vlan WORD	在 VLAN 的入口方向上应用一条 MAC access-list; no 操作为删除绑定在 VLAN 上的 access-list。
vacl ipv6 access-group (<18000-19999>) in (traffic-statistic) vlan WORD no vacl ipv6 access-group {<18000-19999>} in vlan WORD	在 VLAN 的入口方向上应用一条 IPv6 access-list; no 操作为删除绑定在 VLAN 上的 access-list。

5. 显示端口或 VLAN ACL 的配置及统计信息

命令	解释
特权和配置模式	
show access-group in (interface ({ethernet IFNAME} IFNAME))	显示端口上 ACL 绑定情况及统计信息。
show vacl in vlan <vlan-id>	显示 VLAN 上 ACL 绑定情况及统计信息。

6. 清空指定接口或 VLAN 的包过滤统计信息

命令	解释
特权模式	
clear access-group in statistic (interface (IFNAME {ethernet IFNAME}))	清空端口上 ACL 统计信息。
clear vacl in statistic vlan <vlan-id>	清空 VLAN 上 ACL 统计信息。

6.11.2.2 EQoS 配置序列

1. 配置分类表
2. 配置策略表
3. 将 policy 应用到特定端口或 VLAN
4. 清除端口或 VLAN 策略表的计数统计
5. 显示 QoS 配置信息

1. 配置分类表

命令	解释
全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立一个 class-map（分类表），并进入 class-map 模式；本命令的 no 操作为删除指定的 class-map。
match {access-group <acl-index> ip dscp <dscp-list> ip precedence <ip-precedence-list> ipv6 access-group <acl-index> vlan <vlan-list> cos <cos-list>} no match {access-group ip dscp ip precedence ipv6 access-group vlan cos}	设置分类表中的匹配标准，可以根据 ACL、CoS、VLAN ID、IPV4 Precedent、DSCP 等对数据流进行分类；本命令的 no 操作为删除指定的匹配标准。

2. 配置策略表

命令	解释
全局配置模式	
class <class-map-name> [insert-before <class-map-name>] no class <class-map-name>	建立一个 policy-map（策略表）之后，可以将其对应于一个 class（策略分类表），并进入策略分类表模式，之后就可

	以对不同的数据流采取不同的策略或者分配一个新的 DSCP 值；本命令的 no 操作为删除指定策略分类表。
set {ip dscp <new-dscp> ip precedence <new-precedence> internal priority <new-inp> drop precedence <new-dp> ip [default] nexthop [vrf <vrf>] <ip-address> ipv6 nexthop [vrf <vrf>] <ipv6-address> cos <new-cos>} no set {ip dscp ip precedence internal priority drop precedence ip nexthop ipv6 nexthop cos}	为分类后的流量分配一个新的 DSCP、CoS 和 IP Precedence 值等；本命令的 no 操作为取消分配新的值。
单速单桶模式： policy <bit_per_second> <normal_burst_bytes> ({conform-action ACTION exceed-action ACTION}) no policy <bit_per_second> <normal_burst_bytes> ({conform-action ACTION} {exceed-action ACTION}) 双桶模式： policy <bit_per_second> <normal_burst_bytes> (pir <peak_rate_bps>) <maximum_burst_bytes> ({conform-action ACTION exceed-action ACTION violate-action ACTION}) no policy <bit_per_second> <normal_burst_bytes> (pir <peak_rate_bps>) <maximum_burst_bytes> ({conform-action ACTION exceed-action ACTION violate-action ACTION}) ACTION: drop set-dscp-transmit <dscp_value> set-prec-transmit <ip_precedence_value> set-drop-precedence <dp_value> transmit set-internal-priority <inp_value> set-cos-transmit	为分类后的流量配置一个策略，支持三色 color 的非聚合 Policy 命令，根据配置参数解析出令牌桶的工作模式，是单速单桶、单速双桶还是双速双桶，并且给不同颜色报文设置相应动作。no 命令删除模式配置。单桶模式只有特定的交换机支持。

<cos_value>	
accounting no accounting	为分类后的流量设置统计功能，使能 counter 统计，只统计绿色包。

3. 将 QoS 应用到端口或 VLAN

命令	解释
端口配置模式	
mls qos trust {cos dscp} no mls qos trust {cos dscp}	配置交换机端口信任状态；本命令的 no 操作为禁止交换机端口的当前信任状态。
mls qos cos {<default-cos>} no mls qos cos	配置交换机端口的缺省 CoS 值；本命令的 no 操作为恢复缺省情况。
service-policy input <policy-map-name> no service-policy input <policy-map-name>	在端口入方向上应用一个策略表；本命令的 no 操作为删除应用到交换机端口的某个指定策略表。
全局配置模式	
service-policy input <policy-map-name> vlan <vlan-list> no service-policy input <policy-map-name> vlan <vlan-list>	在交换机 VLAN 接口上应用一个策略表；本命令的 no 操作为删除应用到交换机 VLAN 接口的某个指定策略表。

4. 清除特定端口或者 VLAN 的 Accounting 数据

命令	解释
特权配置模式	
clear mls qos statistics [interface <interface-name> vlan <vlan-id>]	清除特定端口或者 vlan Policy Map 的 Accounting 数据，如果不带参数，清除所有的 Policy Map 的 Accounting 数据。

5. 显示 QoS 的配置信息

命令	解释
特权配置模式	

show mls qos maps [cos-intp cos-dp dscp-dscp dscp-intp dscp-dp intp-cos intp-dscp]	显示 QoS 全局映射相关配置内容。
show class-map [<class-map-name>]	显示 QoS 的分类表信息。
show policy-map [<policy-map-name>]	显示 QoS 的策略表信息。
show mls qos {interface [<interface-id>] [policy queuing] vlan <vlan-id>}	显示 QoS 在交换机端口上的配置信息。

6.11.3 EIFP 举例

举例 1:

用户有如下配置需求：在端口 2/10 上，每天 12: 0: 0 到 20: 0: 0 期间不允许如下 TCP 报文通过：源 IP 为 10.1.1.10，源端口范围 1~90，目的端口范围为 3~8，并且 ack 被置位，优先级为 7，tos 值为 4。

配置更改：

1. 创建相应的 time-range
2. 创建相应的 ACL
3. 配置包过滤功能
4. 绑定 ACL 到端口

配置步骤如下：

```
switch(config)#time-range a
switch(config-time-range-a)# periodic daily 12:0:0 to 20:0:0
switch(config-time-range-a)#exit
switch(config)# access-list 11000 deny tcp 10.1.1.10 0.0.0.0 s-port range 1 90
any-destination d-port range 3 8 ack precedence 7 tos 4 time-range a
switch(config)#firewall enable
switch(config)#interface ethernet 2/10
switch(config-if-ethernet2/10)#ip access-group 11000 in
switch(config-if-ethernet2/10)#exit
switch(config)#exit
```

配置结果：

```
switch#show firewall
Firewall Status: Enable.
```

```
switch#show access-lists
access-list 11000(used 1 time(s)) 1 rule(s)
rule ID 1: deny tcp host-source 10.1.1.10 s-port range 1 90 any-destination d-port range 3
8 ack precedence 7 tos 4 time-range a (active)
```

```
switch# show access-group interface ethernet 2/10
interface name:Ethernet2/10
IP Ingress access-list used is 11000, traffic-statistics Disable.
```

举例 2:

用户有如下配置需求：交换机的 1/10 端口不允许转发源 MAC 地址是 00-12-11-23-XX-XX 的 802.3 的数据报文。

配置更改：

1. 创建相应的 MAC ACL
2. 配置包过滤功能
3. 绑定 ACL 到端口

配置步骤如下：

```
Switch(config)#access-list 14000 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac untagged-802-3
Switch(config)#access-list 14000 deny 00-12-11-23-00-00 00-00-00-00-ff-ff
any-destination-mac tagged-802-3
Switch(config)#firewall enable
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#mac access-group 14000 in
Switch(Config-If-Ethernet1/10)#exit
Switch(config)#exit
```

配置结果：

```
Switch#show firewall
Firewall Status: Enable.
```

```
Switch #show access-lists
access-list 14000(used 1 time(s)) 2 rule(s)
access-list 14000 deny 00-12-11-23-00-00 00-00-00-00-ff-ff any-destination-mac
untagged-802-3
      access-list 14000 deny 00-12-11-23-00-00 00-00-00-00-ff-ff any-destination-mac
tagged-802-3
```

Switch #show access-group interface ethernet 1/10

interface name:Ethernet1/10

MAC Ingress access-list used is 14000,traffic-statistics Disable.

案例 3:

用户有如下配置需求：交换机的端口连接的网段是 IPv6，并且 IPv6 的地址为 2003: 1: 1: 1:: 0/48 网段，管理员不希望除了 2003: 1: 1: 1:: 0/64 网段用户访问外网。

配置更改：

1. 创建相应的 ACL
2. 配置包过滤功能
3. 绑定 ACL 到端口

配置步骤如下：

Switch(config)#ipv6 enable

Switch(config)#ipv6 access-list 18000 permit 2003:1:1:1::0/64 any-destination

Switch(config)#ipv6 access-list 18000 deny 2003:1:1:1::0/48 any-destination

Switch(config)#firewall enable

Switch(config)#interface ethernet 1/10

Switch(Config-If-Ethernet1/10)#ipv6 access-group 18000 in

Switch(Config-If-Ethernet1/10)#exit

Switch(config)#exit

配置结果：

Switch#show firewall

Firewall Status: Enable.

Switch#show ipv6 access-lists

Ipv6 access-list 18000(used 1 time(s)) 2 rule(s)

ipv6 access-list 18000 permit 2003:1:1:1::0/48 any-source

ipv6 access-list 18000 deny 2003:1:1:1::0/64 any-source

Switch #show access-group interface ethernet 1/10

interface name:Ethernet1/10

IPv6 Ingress access-list used is 18000, traffic-statistics Disable.

案例 4

用户有以下配置需求：交换机的 1, 2, 5, 7 端口属于 Vlan100，管理员不希望 IP 地址为 192.168.0.1 的设备接入到这几个端口。

配置更改：

1. 创建相应的 ACL
2. 配置包过滤功能
3. 绑定 ACL 到端口

配置步骤如下：

```
Switch (config)#firewall enable
Switch (config)#vlan 100
Switch (Config-Vlan100)#switchport interface ethernet 1/1;2;5;7
Switch (Config-Vlan100)#exit
Switch (config)#access-list 11000 deny ip 192.168.0.1 0.0.0.0 any-destination
Switch (config)#vacl ip access-group 11000 in vlan 100
```

配置结果：

```
Switch (config)# show vacl vlan
VLAN 100:
IP Ingress access-list used is 11000, traffic-statistics Disable.
```

案例 5

在端口 ethernet1/2 上，将属于网段 192.168.1.0 内的报文带宽限制为 10M 比特/秒，突发值设为 4M 字节，超过带宽的该网段内的报文一律丢弃。

配置步骤如下：

```
Switch#config
Switch(config)#access-list 11000 permit ip 192.168.1.0 0.0.0.255 any-destination
Switch(config)#class-map c1
Switch(Config-ClassMap-c1)#match access-group 11000
Switch(Config-ClassMap)# exit
Switch(config)#interface ethernet 1/2
Switch(Config-If-Ethernet1/2)#service-policy input p1
```

配置结果：

网段 192.168.1.0 内的报文带宽在端口 ethernet 1/2 上被限制为 10M 比特/秒，突发值为 4M 字节，超过带宽的该网段内的报文一律丢弃。

6.11.4 EIFP 排错帮助

- ☞ 确认板卡支持 EIFP，目前仅 5662x 芯片支持 EIFP，即 kylin 卡支持 EIFP。
- ☞ 对于 IP 和 MAC ACL，288bit 和 144bit 的模板不能同时存在，所以这两个命令同时配置时，只有最后配置的生效。当配置 288bit 宽的模板时，IP 和 MAC 对应的 144bit 的表项数被清零，反之亦然。
- ☞ MAC ACL 只对非 IP 报文的二层报文生效，对 IP 报文不生效（即使它匹配了此 ACL 的所有信息）。
- ☞ 对 ACL 中的表项的检查是自上而下的，只要匹配一条表项，对此 ACL 的检查就马上结束。对于不同类型的 ACL 可以在端口或 VLAN 上各绑定一条，对于同类型的 ACL 在端口或 VLAN 上只能绑定一条。
- ☞ 物理端口加入/移出 VLAN 时（Trunk 口除外），该 VLAN 中配置的 ACL 将自动绑定/移除该物理端口；如果目的 VLAN 中配置的 ACL 和该端口原有的 ACL 配置冲突，将导致端口移动失败。
- ☞ VLAN 中无物理端口时（Trunk 口除外），将删除 ACL 在 VLAN 中的配置，以后如果再有物理端口移入 VLAN，将无 ACL 应用于该物理端口。
- ☞ 端口模式转换：access->trunk，将取消通过 VLAN 接口配置模式绑定到该物理接口的 ACL；trunk->access，将 Vlan 1 接口配置的 ACL 绑定到该物理接口，如果绑定失败，端口模式转换将执行失败。
- ☞ EQoS 统计每次只能统计一种颜色的报文，可通过在 policy map 的 match 时设定模式。
- ☞ 每个 VLAN 上面只能配置一个 Policy Map。
- ☞ 系统将对 Policy Map 的每个 Class 中的每个规则增加一个 VLAN ID 匹配属性，如果原有规则中有匹配 VLAN ID 的属性，并且 VLAN ID 和绑定的 VLAN ID 不一致，这个 Policy Map 不能被绑定。

6.12 Captive Portal 认证

6.12.1 Captive Portal 认证配置

6.12.1.1 Captive Portal 认证介绍

认证功能是为管理和控制使用网络资源的用户的一种方法。认证功能将客户端的认证信息按照一定的原则存储在认证服务器中，当用户需要使用网络资源时，captive portal 功能将用户的网络请求重定向到认证服务器上，这时需要用户提供被允许的用户名和密码等信息，由认证服务器对用户提供的信息进行判断，以决定是否允许该用户使用网络资源。交换机在认证功能中主要起着传递用户和认证服务器之间通信的作用。通过在交换机上进行配置使得客户端可以与认证服务器进行连接和沟通，并且在认证服务器对客户端信息判断完成后将认证结果和相应处理推向用户。认证功能是建立在重定向功能基础之上的。

重定向功能是一种将原请求重新连接到设定好的地点后再进行后续操作的功能。该功能

是在交换机接收到客户端的请求后，将客户端的请求转到一个已经设定好的地址上，之后的操作由客户端和重定向之后的地址进行操作，以此完成某些特定的功能和操作。该操作可以实现达到管理和监控用户的目的。客户端被重定向 portal 认证界面，需要用户填写用户名和密码，只有当该用户名和密码通过认证后才允许用户使用网络资源。Portal 认证方式可以实现对不同类型用户的不同控制策略。

多 portal server 功能是为不同的 CP configuration 配置不同的外置 portal server 的一种方法。绑定在不同 CP configuration 下的端口配置了不同 portal server 之后就可以各自通过自身绑定的 portal server 来推出重定向页面。最多可以配置 10 台外置 portal 服务器。每个 cp configuration 可以绑定一台 portal 服务器。

6.12.1.2 Captive Portal 认证配置

认证功能配置任务序列如下：

1. 打开或关闭 captive portal 认证功能
2. 配置 captive portal 重定向功能
 - 1) 设置或删除 portal 服务器名称
 - 2) 配置用户例程
 - 3) 配置重定向地址
 - 4) 配置重定向 url 头部
 - 5) 设置或删除 radius 服务器名称
 - 6) 配置绑定或者解除 portal 服务器
 - 7) 配置 url 携带 ac-name 参数
 - 8) 配置 url 携带 ssid 参数
 - 9) 配置 url 携带 nas-ip 参数
3. 配置 AAA 功能
 - 1) 打开使能或禁止 AAA 功能
 - 2) 配置 RADIUS 认证服务器组名称
4. 配置 RADIUS 认证服务器
 - 1) 配置 RADIUS 服务器密钥
 - 2) 配置 RADIUS 认证服务器地址
5. 将 portal 规则绑定到端口下

1. 打开或关闭 captive portal 认证功能

命令	解释
Captive portal 配置模式	
enable disable	全局打开或关闭 captive portal 功能。

26. 2. 配置 captive portal 重定向功能

命令	解释
Captive Portal 配置模式	
external portal-server server-name <name> {ipv4 ipv6} <ipaddr> [port <1-65535>] no external portal-server {ipv4 ipv6}server-name <name>	配置或删除外置 portal 服务器。
nas-ip4 <A.B.C.D>	配置 nas-ip 地址
configuration <cp-id> no configuration <cp-id>	配置与删除不同类型用户的 portal 例程。该例程可以配置 10 种不同的例程。
Captive Portal Instance 配置模式	
redirect url-head <word> no redirect url-head	配置重定向 url 头部分，包括传输协议、主机名、端口和 path 等几个部分。本命令的 no 形式为删除重定向 url 头部分的配置。
radius-auth-server <server-name> no radius-auth-server	设置或删除 radius 认证服务器名称。
portal-server {ipv4 ipv6} <name> no portal-server {ipv4 ipv6}	绑定或解除绑定 portal 服务器名称。
enable disable	打开或关闭某个 portal 例程。
ac-name <word> no ac-name	配置重定向 url 中的参数 acname 的值。本命令的 no 形式为删除重定向 url 中的参数 acname 值的配置。
redirect attribute ssid name <word> no redirect attribute ssid name	配置在重定向 url 中携带的 ssid 参数名称。本命令的 no 形式为恢复重定向 url 中携带的 ssid 参数名称为默认值。
redirect attribute nas-ip enable no redirect attribute nas-ip enable	使能重定向 url 中携带 nas-ip 参数功能。本命令的 no 形式为关闭重定向 url 中携带 nas-ip 参数的功能。
redirect attribute nas-ip name <word> no redirect attribute nas-ip name	配置在重定向 url 中携带的 nas-ip 参数的名称。本命令的 no 形式为恢复重定向 url 中携带的 nas-ip 参数的名称为默认值。

3. 配置 AAA 功能

命令	解释
全局配置模式	
aaa enable no aaa enable	使能或禁止 AAA 功能
aaa group server radius <word> no aaa group server radius <word>	设置或删除设置 AAA 功能的 RADIUS 名称。

4. 配置 RADIUS 认证服务器

命令	解释
全局配置模式	
radius-server key <word> no radius-server key	设置或删除 RADIUS 服务器密钥。
radius-server authentication host <A.B.C.D> no radius-server authentication host <A.B.C.D>	配置或删除 RADIUS 认证服务器地址。

5. 开启/关闭端口 portal 功能

命令	解释
config 配置模式	
vlan-pool <1-255> <WORD> no vlan-pool <1-255>	配置或者删除地址池（可选）。
端口模式	
portal enable configuration <id> [vlan-pool WORD] no portal enable [vlan-pool WORD]	开启端口 portal 认证功能，指定端口绑定的实例号，并且可以指定哪些 VLAN 使能 portal，如果不指定 vlan，则默认所有 vlan 使能 portal。本命令的 no 形式为关闭端口 portal 认证功能。（必选）

6.12.1.3 Captive Portal 认证举例

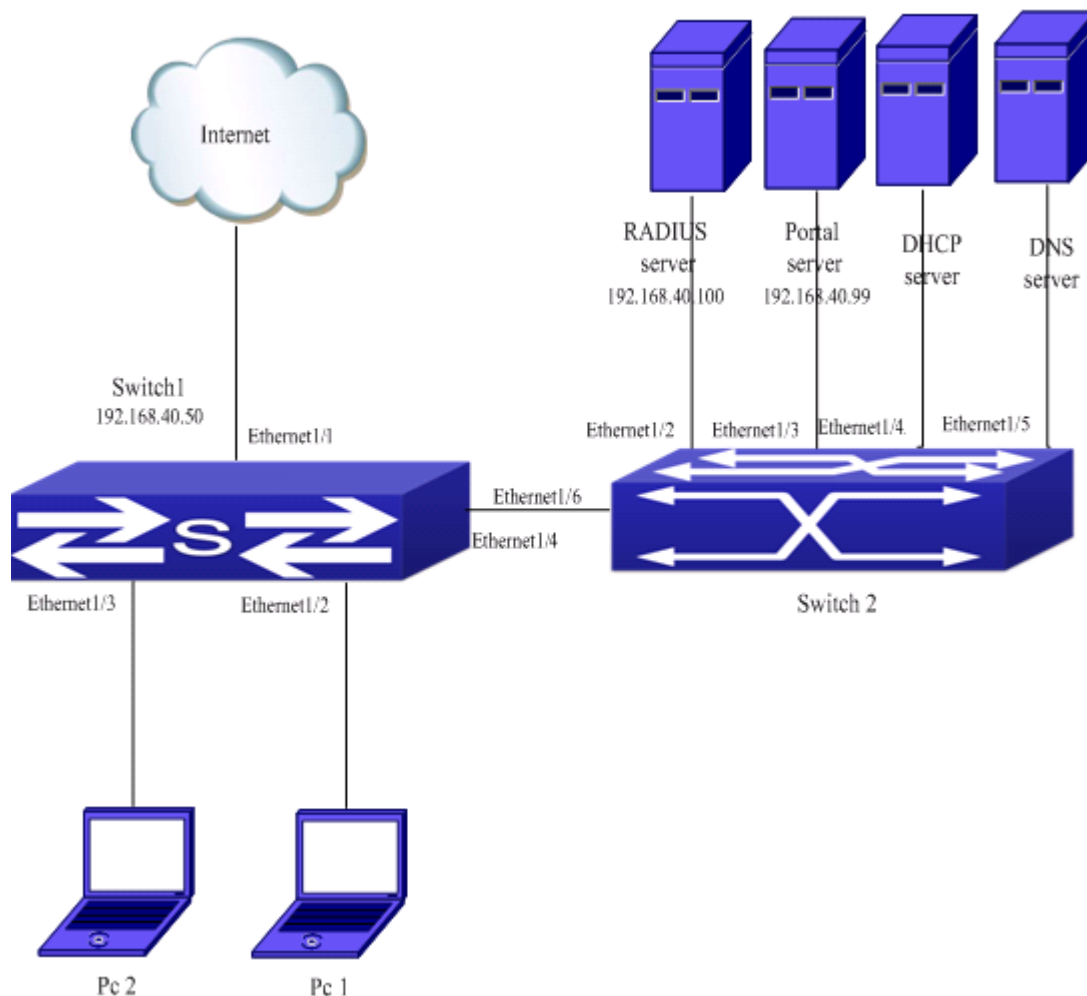


图 6-30 认证功能配置

如图所示，pc1 为终端用户，上面有 http 浏览器，但没有任何 802.1x 认证客户端，pc1 想要通过 portal 认证上网。

交换机是 Switch1 为接入设备，上面配置了计费服务器的地址为 RADIUS 服务器的 IP 和端口，打开了计费功能。其中 Ethernet1/2 和 pc1 相连，端口打开了 portal 认证功能，配置了重定向的地址为 portal 服务器的 IP 和端口，因此端口 Ethernet1/2 禁止所有的流量，只放行 dhcp/dns/arp 报文通过。

交换机 Switch2 为汇聚交换机，Ethernet1/2 与后台 radius 服务器互通，Ethernet1/3 与 portal 服务器互通。Radius 服务器的地址为 192.168.40.100，portal 服务器的地址为 192.168.40.99。Ethernet1/4 连接的是 DHCP 服务器，Ethernet1/5 连接的是 DNS 服务器。Ethernet1/6 为 trunk 口，与 Switch 1 的 trunk 口 Ethernet1/4 相连。

配置 radius 服务器：

```
switch (config)#radius-server key 0 test
```

```
switch (config)#aaa group server radius radius_aaa_1
switch (config-sg-radius)# server 192.168.40.100
```

portal 全局开启认证功能相关的配置如下：

```
Switch(config)#interface vlan 1
Switch(config-if-vlan1)#ip address 192.168.40.50 255.255.255.0
Switch(config)#free-resource 1 destination ipv4 192.168.40.99/32
Switch(config)# vlan-pool 1 2-10
```

portal实例下Portal功能、配置Portal Server服务器等信息：

```
Switch (config)#captive-portal
Switch (config-cp)#enable
Switch(config-cp)# nas-ipv4 192.168.40.50
Switch(config-cp)# external portal-server server-name abc ipv4 192.168.40.99
Switch (config-cp)# configuration 1
Switch (config-cp-instance)#name helix4
Switch (config-cp-instance)#radius-auth-server abc99
Switch (config-cp-instance)# redirect attribute nas-ip enable
Switch (config-cp-instance)#redirect attribute nas-ip name kk
Switch (config-cp-instance)#ac-name helix4
Switch (config-cp-instance)#redirect url-head http://192.168.40.99/a70.htm
Switch (config-cp-instance) # portal-server ipv4 abc
```

端口上开启 portal 功能

```
Switch (config)# interface ethernet1/2
Switch (config-if-ethernet1/2)#portal enable configuration 1 vlan-pool 1
```

6.12.1.4 Captive Portal 认证排错帮助

当在使用重定向功能和 Captive Portal 功能过程中遇到问题，请检查是否是如下原因：

- ☞ 是否启动了 captive portal 功能以及是否打开了 portal 的 configuration 开关，这两项都应当打开，否则 captive portal 功能不能正常使用，客户端也不能重定向到指定页面。
- ☞ AAA 模块的认证服务器名称与 captive portal 配置的认证服务器名称相同。
- ☞ pc 和交换机相连的端口上是否打开了 portal 认证功能。

6.12.2 计费功能配置

6.12.2.1 计费功能介绍

计费功能是为对网络中的用户使用网络资源进行监控和计费的功能。客户端在未通过 captive portal 认证功能之前是无法访问网络资源的，只有通过了 portal 认证才能访问网络资源，可以定义用户的会话时长以控制用户使用网络资源的时间和流量等信息。

6.12.2.2 计费功能配置

计费功能配置任务序列如下：

1. 配置 RADIUS 计费服务器
 - 1) 设置或删除计费服务器地址
2. 配置 AAA 计费功能
 - 1) 打开或关闭计费服务功能
3. 配置 captive portal 计费功能
 - 1) 阻塞或解除阻塞 portal 功能
 - 2) 设置或删除 captive portal 配置名称
 - 3) 打开或关闭 captive portal 计费功能
 - 4) 配置或删除 captive portal 计费服务器名称
 - 5) 配置或删除 captive portal 会话时长

1. 配置 RADIUS 计费服务器

命令	解释
全局配置模式	
radius-server accounting host <A.B.C.D> no radius-server accounting host <A.B.C.D>	设置或删除计费服务器地址。

2. 配置 AAA 计费功能

命令	解释
全局配置模式	
aaa-accounting enable no aaa-accounting	打开或关闭计费服务功能。

3. 配置 captive portal 计费功能

命令	解释
Captive portal 配置模式	
block no block	阻塞或解除阻塞 portal 功能

name <word> no name	设置或删除 captive portal 配置名称。
radius accounting no aaa-accounting	打开或关闭 captive portal 计费服务功能。
radius-acct-server <word> no radius-acct-server	配置或删除 captive portal 计费服务器名称。
session-timeout <0-86400> session-timeout	配置或删除 captive portal 会话时长。

6.12.2.3 计费功能举例

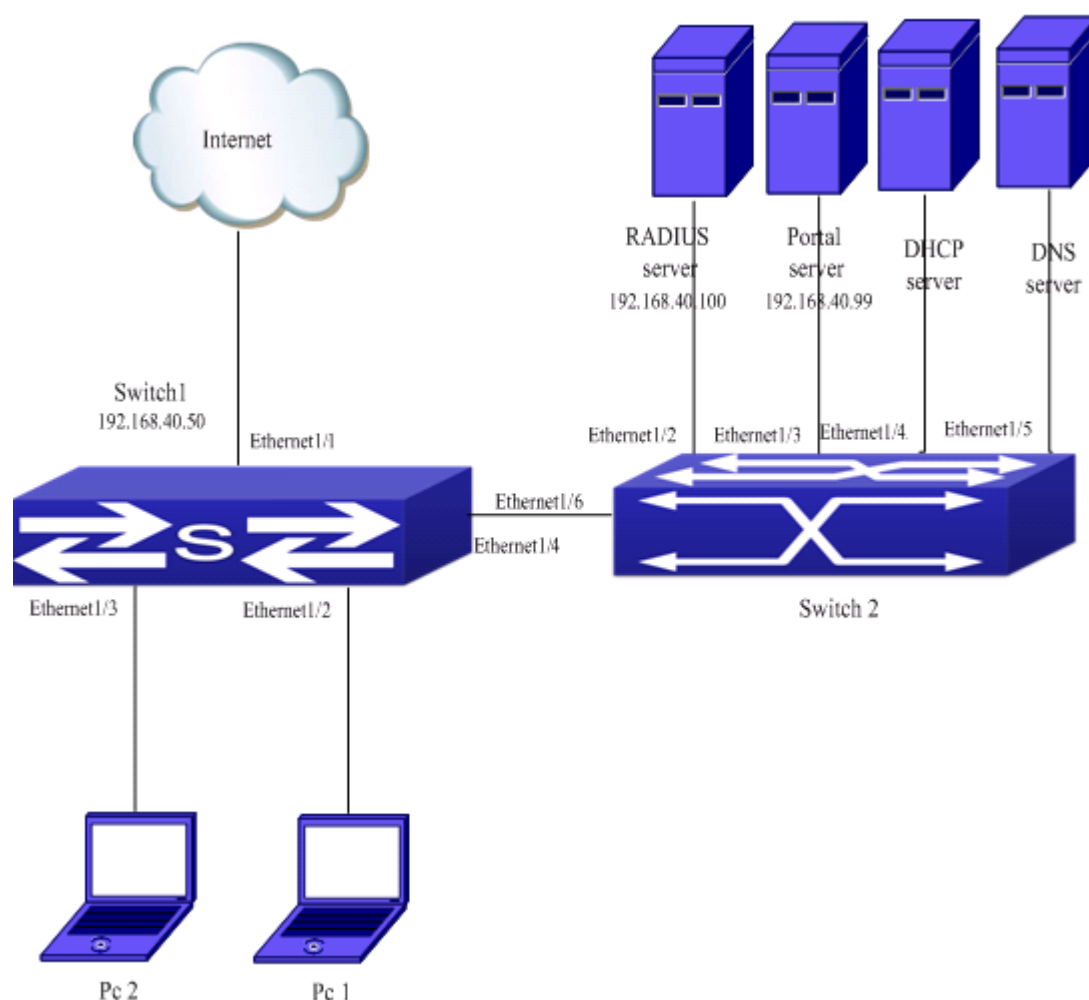


图 6-31 计费功能配置图

1. 在 switch1 上配置 AAA 计费功能

Switch1 的 AAA 配置：

```

switch 1(config)# aaa enable
switch 1(config)# aaa-accounting enable
switch 1(config)# radius-server accounting host 192.168.40.100
switch1 (config)#radius-server key 0 test
switch1 (config)#aaa group server radius abc99
switch (config-sg-radius)# server 192.168.40.100

```

2. 在 switch1 上配置 captive portal 计费功能

```

Switch 1(config)#captive-portal
Switch 1(config-cp)#enable

Switch1 (config-cp)# configuration 1
Switch1 (config-cp-instance)#radius accounting
Switch1 (config-cp-instance)# radius-acct-server abc99

```

6.12.2.4 计费功能排错帮助

当在使用计费功能过程中遇到问题，请检查是否是如下原因：

- ☞ 是否启动了 captive portal 功能以及是否打开了 portal 的 configuration 开关，这两项都应当 enable，否则 captive portal 功能不能正常使用。
- ☞ AAA 模块的认证服务器名称与 captive portal 配置的认证服务器名称相同。

6.12.3 Free-resource 功能配置

6.12.3.1 Free-resource 功能介绍

Free-resource 功能是 captive portal 模块中实现访问免费资源规则的方法，通过配置 free-resource 规则，可以让某些特定的客户端不通过 portal 认证就可以直接访问特定的网络资源。

6.12.3.2 Free-resource 功能配置

1. 配置 free-resource 规则

命令	解释
config 配置模式	
free-resource destination {ipv4 A.B.C.D/M ipv6 X:X:X:X/M} [no] free-resource destination {ipv4 A.B.C.D/M ipv6 X:X:X:X/M}	配置或删除 free-resource 规则。

6.12.3.3 Free-resource 功能举例

案例：

搭建环境如下图所示，IP 为客户端 client1 所在的地址段，目的 IP 为客户端想要访问的资源的地址段，指定 RADIUS Server 1 为认证服务器，客户端 client1 和 clint2 可以访问该免费资源 3.1.1.0/24，不会被重定向到认证服务器上，直接就可以成功访问。

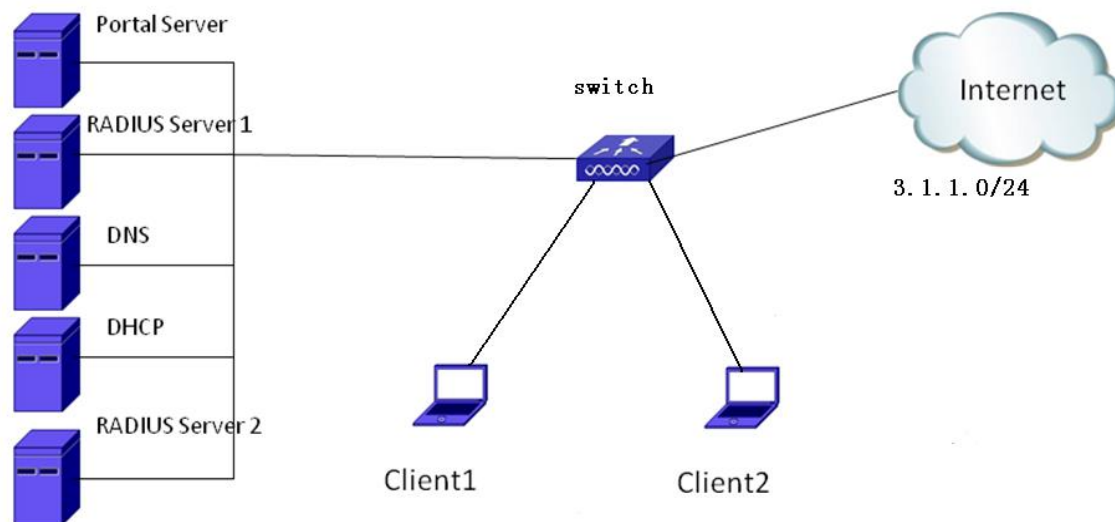


图 6-32 多 portal server 功能配置

配置免费资源步骤：

```
Switch1 (config)# free-resource destination ipv4 3.1.1.0/24
```

6.12.3.4 Free-resource 功能排错帮助

当在使用重定向功能过程中遇到问题，请检查是否是如下原因：

- ☞ 是否正确启动了 captive portal 功能以及是否打开了 portal 的 configuration 开关，这两项都应当打开，否则多 portal server 功能不能正常使用，客户端也不能重定向到指定页面。
- ☞ Client 和交换机互连的端口是否开启了 portal 规则。

6.12.4 认证白名单功能配置

6.12.4.1 认证白名单功能介绍

认证白名单功能是为对网络中某些特殊的用户使用的，管理员可以设置一些用户使其连接到网络之后不需要通过认证即可使用网络资源，但是管理员需要获得该类用户的 mac 地址才能使用该功能，同时具有该权限的用户所使用的网络资源不需要不给计费，因此该类用户的权限属于高级用户类型。

6.12.4.2 认证白名单功能配置

1. 配置具有认证白名单功能权限的用户的 mac

命令	解释
config 配置模式	
free mac < MACADD><MACMASK> no free mac < MACADD><MACMASK>	设置或删除可以免认证的 mac 地址。

6.12.4.3 认证白名单功能举例

如图所示，client1 和 client2 为终端用户，和交换机相连的端口开启了 portal 认证，但是这两个用户为高级用户，不需要认证，就可以正常访问资源。

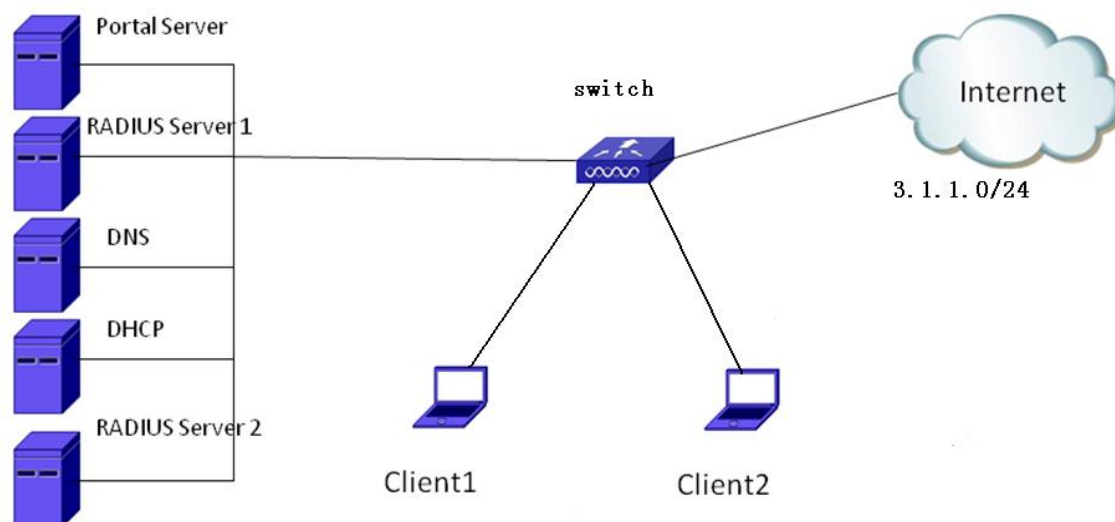


图 6-33 认证白名单功能配置图

配置步骤：

为 pc1 和 pc2 配置免 portal 认证白功能

```
Switch1 (config)#free-mac 68-74-7f-29-76-04 ff-ff-ff-ff-ff
```

```
Switch1 (config)#free-mac 00-03-0f-11-11-11 ff-ff-ff-ff-ff
```

6.12.4.4 认证白名单功能排错帮助

当在使用认证白名单过程中遇到问题，请检查是否是如下原因：

- ☞ 配置的 free mac 是否和 client 的 mac 相匹配；
- ☞ Client 和交换机互连的端口是否开启了 portal 规则。

6.12.5 认证成功页面自动推送（暂不支持）

6.12.5.1 认证成功后页面自动推送介绍

认证成功后页面自动推送功能就是在用户通过认证后,能够重新打开用户所需要访问的网页。根据实际情况可以配置用户认证成功后自动推送认证前访问页面,或者自动推送指定的网页。

6.12.5.2 认证成功后页面自动推送配置

认证成功后页面自动推送功能配置序列:

1. 打开或关闭 captive portal 认证功能
2. 配置认证成功后页面自动推送功能

1. 打开或关闭 captive portal 认证功能

命令	解释
Captive portal 配置模式	
enable disable	全局打开或关闭交换机的 captive portal 功能。

2. 配置认证成功后页面自动推送功能

命令	解释
Captive portal instance 配置模式	
redirect attribute url-after-login enable no redirect attribute url-after-login enable	使能重定向 url 中携带认证成功后推送 url 地址的功能。本命令的 no 形式为关闭重定向 url 中携带认证成功后推送 url 地址的功能。
redirect attribute url-after-login name <name> no redirect attribute url-after-login name	配置在重定向 url 中携带认证成功后推送 url 地址的属性值名称。本命令的 no 形式为恢复重定向 url 中携带认证成功后推送 url 地址的属性值名称为默认值。
redirect attribute url-after-login encode {plain-text base64}	配置对重定向 url 中携带的认证成功后推送 url 地址的编码方式。
redirect attribute url-after-login value <url-value> no redirect attribute url-after-login value	配置用户认证成功后弹出的指定页面的 url 地址。本命令的 no 形式为删除用户认证成功后弹出的指定页面的 url 地址。

6.12.5.3 认证成功后页面自动推送举例

案例:

在 configuration 1 中配置认证成功后页面自动推送功能，认证成功后自动推送页面 <http://www.test.com>。

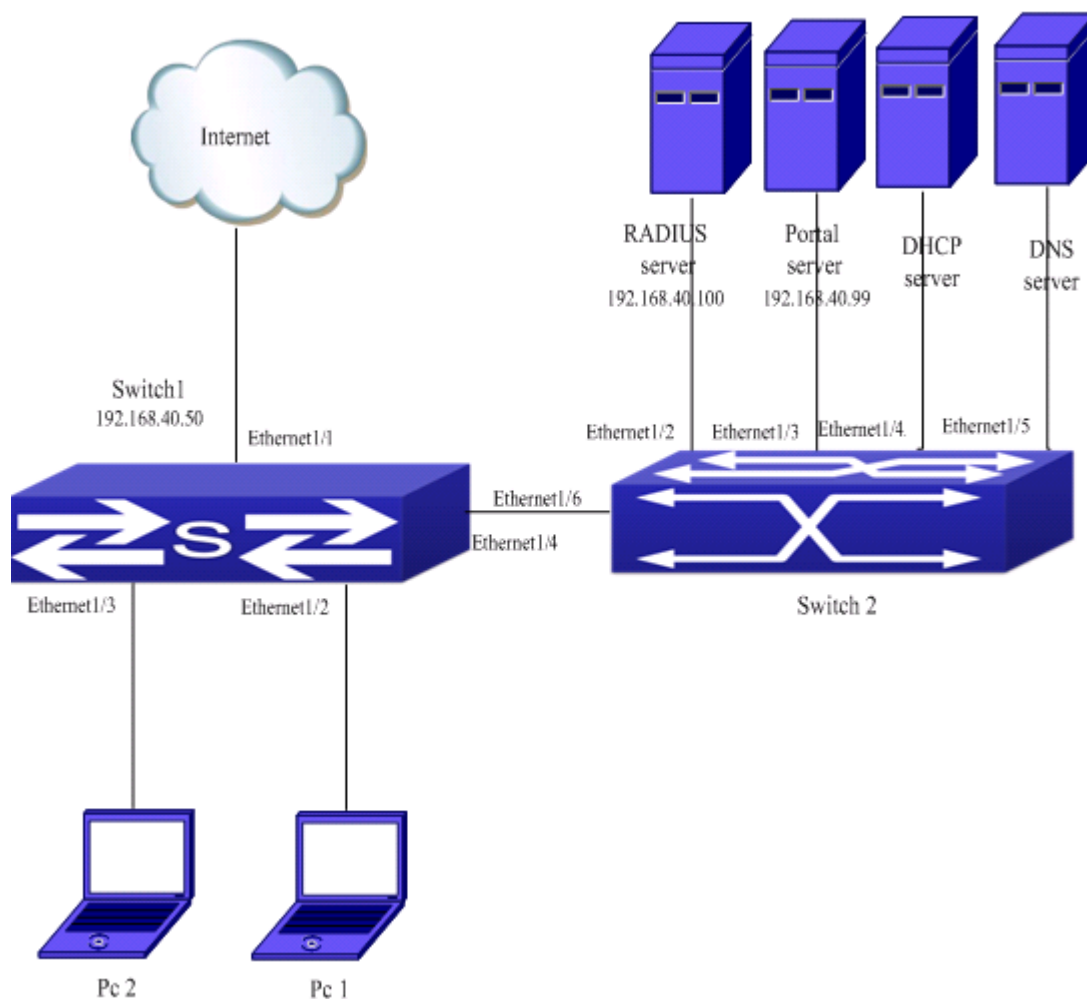


图 6-34 认证成功后页面自动推送功能配置

配置步骤：

为 siwitch1 配置 portal 服务器信息

```
switch1 (config-cp)#enable
```

```
switch1 (config-cp)#configuration 1
```

```
switch1 (config-cp-instance)#enable
```

```
switch1 (config-cp-instance)# redirect attribute url-after-login enable
```

```
switch1 (config-cp-instance)# redirect attribute url-after-login encode plain-text
```

```
switch1 (config-cp-instance)# redirect attribute url-after-login name ad
```

```
switch1 (config-cp-instance)# redirect attribute url-after-login value http://www.test.com
```

6.12.5.4 认证成功后页面自动推送排错帮助

当在使用认证成功后页面自动推送功能过程中遇到问题，请检查是否是如下原因：

- ☞ 是否正确配置了 captive portal 的认证功能。认证成功后页面自动推送功能需要 captive portal 功能正常的情况下才能生效。
- ☞ 若配置了命令 `redirect attribute url-after-login value`，则认证通过后页面自动推送配置的页面地址，若没有配置该命令则推送的页面为用户认证前访问的页面。
- ☞ 客户端认证前访问的页面或命令指定的推送页面是否存在，若不存在页面推送后也无法访问。

6.12.6 http-redirect-filter

6.12.6.1 http-redirect-filter 介绍

http-redirect-filter 为 portal 认证的 HTTP 重定向指定 IP 或域名，只有匹配了这个 IP 或域名的 HTTP 报文才做重定向处理。

此功能与 Captive Portal 功能结合使用，即在开启了 Captive Portal 接入认证的情况下，实现对用户的访问进行过滤。

6.12.6.2 http-redirect-filter 配置

http-redirect-filter 功能配置序列：

1. 配置 http-redirect-filter 规则
2. 配置 http-redirect-filter 规则到 cp instance 中

1. 配置 http-redirect-filter 规则

命令	解释
captive portal 模式	
http-redirect-filter <1-32> (ip A.B.C.D domain WORD) no http-redirect-filter (<1-32> all)	配置 http-redirect-filter 规则。本命令的 no 形式为删除规则。

2. 配置 http-redirect-filter 规则到 cp instance 中

命令	解释
captive portal 模式	
http-redirect-filter <1-32> no http-redirect-filter (<1-32> all)	绑定 http-redirect-filter 规则到 captive portal configuration 下。本命令的 no 形式为解除 captive portal configuration 下 http-redirect-filter 的绑定。

6.12.6.3 http-redirect-filter 举例

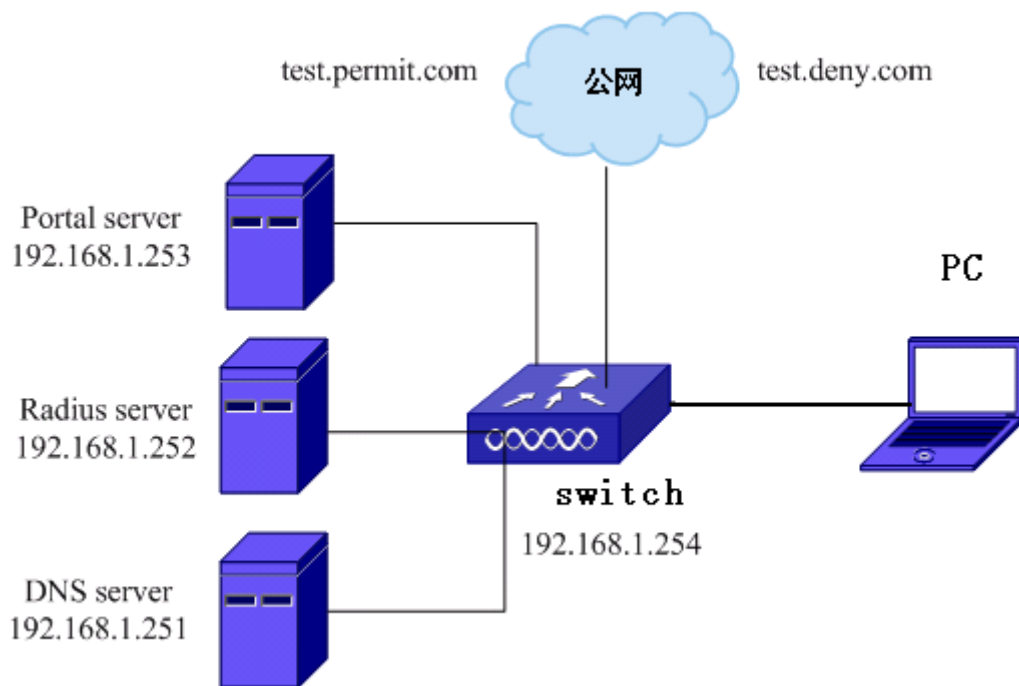


图 6-35 http-redirect-filter 功能典型案例

如上图所示的拓扑，client 在通过 portal 认证前想要访问公网地址“test.permit.com”需要配置 url 白名单，client 认证通过后需要禁止访问公网地址“test.deny.com”需要配置 url 黑名单。

按照以下步骤进行配置：

1. 全局模式下配置 RADIUS server 相关的认证密钥，认证服务器，计费服务器，aaa 模式等信息：

```
switch (config)#radius-server key 0 test
switch (config)#radius-server authentication host 192.168.1.252
switch (config)#radius-server accounting host 192.168.1.252
switch (config)#aaa-accounting enable
switch (config)#aaa enable
switch (config)#aaa group server radius radius_aaa_1
switch config-sg-radius)# server 192.168.1.252
switch(config)#interface vlan 192
switch(config-if-vlan1)#ip address 192.16.1.254 255.255.255.0
switch(config)#free-resource 1 destination ipv4 192.168.1.252/32
```

2. portal 实例下 Portal 功能、配置 Portal Server 服务器等信息：

```
Switch (config)#captive-portal
Switch (config-cp)#enable
Switch(config-cp)# nas-ipv4 192.168.1.254
```



```
Switch(config-cp)# external portal-server server-name abc ipv4 192.168.1.253
Switch (config-cp)# configuration 1
Switch (config-cp-instance)#enable
Switch (config-cp-instance)#name helix4
Switch (config-cp-instance)#radius accounting
Switch (config-cp-instance)#radius-acct-server abc99
Switch (config-cp-instance)#radius-auth-server abc99
Switch (config-cp-instance)#redirect attribute nas-ip enable
Switch (config-cp-instance)#ac-name helix4
Switch (config-cp-instance)#redirect url-head http://192.168.1.253/a70.htm
Switch (config-cp-instance)#portal-server ipv4 abc
```

3. 配置 http-redirect-filter 规则:

```
Switch (config)#captive-portal
Switch (config-cp)# http-redirect-filter 1 domain test.permit.com
Switch (config-cp)#configuration 1
Switch (config-cp-instance)# http-redirect-filter 1
```

4. 端口上开启 portal 功能:

```
Switch (config)# interface ethernet1/2
Switch (config-if-ethernet1/2)#portal enable configuration 1
```

通过配置，可以看出在 client 认证通过前只有访问“test.permit.com”，才能正常重定向认证，访问其它地址，则不能重定向。

6.12.6.4 http-redirect-filter 排错帮助

在使用中，如果发现 http-redirect-filter 功能无法正常使用，可以采用下面的检查步骤：

- ☞ 查看配置的匹配的规则内容是否与访问的域名匹配；
- ☞ DNS server 的配置是否正确，能否正确解析出配置的域名；
- ☞ captive-portal configuration 下 http-redirect-filter 命令是否配置。

6.12.7 Portal 无感知

6.12.7.1 Portal 无感知介绍

MAC 认证具备“一次认证，多次使用”用户体验。如果开通了 MAC 快速认证，用户首次登陆 Portal 页面成功认证后，后续用户就可以用任意应用上网。

为了实现大量用户的 MAC 快速认证，必须使用外置的服务器来保存 MAC 绑定信息，

并且能够动态的添加，而不是手动添加，这一新的实现方案被称为 MAC 快速认证方案，由于用户再次接入网络时不需要手动输入用户名和密码进行认证，因此也称为 Portal 无感知认证方案。

6.12.7.2 Portal 无感知配置

1. 开启/关闭 mac 快速认证功能

命令	解释
captive portal config 模式	
fast-mac-auth no fast-mac-auth	开启或关闭 mac 快速认证功能。

6.12.7.3 Portal 无感知举例

搭建环境如下图所示，其中包括以下几部分：

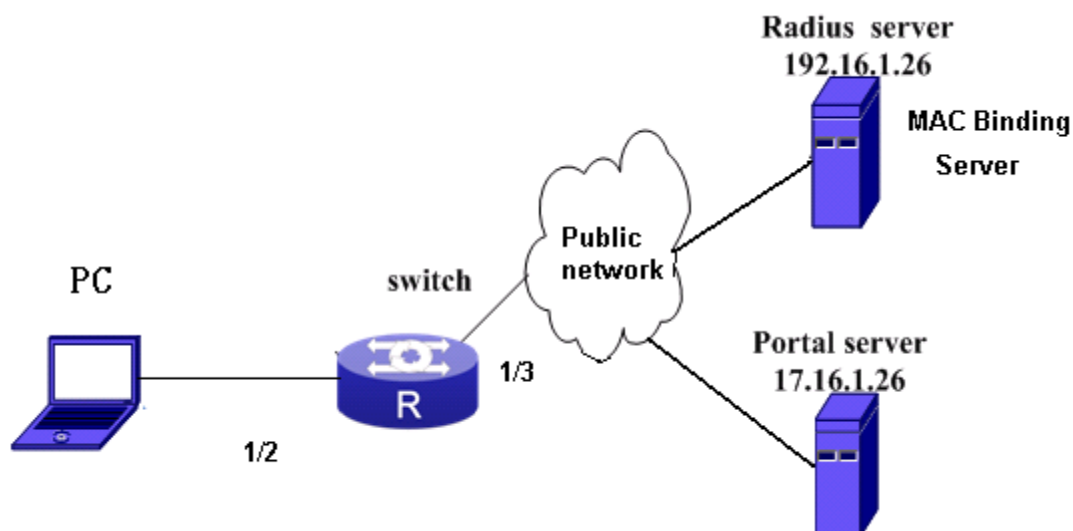
- 1、PC，用户通过交换机接入网络。
- 2、公共网，此部分可以没有，也可以是其他的交换机设备。
- 3、服务器，其中包括：

MAC 绑定服务器，用于保存曾经认证通过的终端 MAC 地址；

Radius 服务器，用于对用户进行 portal 认证和计费；

Portal 服务器，用于对用户进行 portal 认证时使用；

MAC 绑定服务器、Radius 服务器、portal 服务器可以是一台设备，MAC 绑定服务器是在 Radius 服务器上的扩展。



按照以下步骤进行配置：

1. 具体的配置如下：全局模式下配置 RADIUS server 相关的认证密钥，认证服务器，计费服务器，aaa 模式等信息：

```
switch (config)#radius-server key 0 test
switch (config)#radius-server authentication host 192.16.1.26
switch (config)#radius-server accounting host 192.16.1.26
switch (config)#aaa-accounting enable
switch (config)#aaa enable
switch (config)#aaa group server radius radius_aaa_1
switch config-sg-radius)# server 192.16.1.26
switch(config)#interface vlan 192
switch(config-if-vlan1)#ip address 192.16.1.50 255.255.255.0
switch(config)#free-resource 1 destination ipv4 192.16.1.26/32
```

2. portal 实例下 Portal 功能、配置 Portal Server 服务器等信息：

```
Switch (config)#captive-portal
Switch (config-cp)#enable
Switch(config-cp)# nas-ipv4 192.168.1.50
Switch(config-cp)# external portal-server server-name abc ipv4 172.16.1.26
Switch (config-cp)# configuration 1
Switch (config-cp-instance)#enable
Switch (config-cp-instance)#name helix4
Switch (config-cp-instance)#radius accounting
Switch (config-cp-instance)#radius-acct-server abc99
Switch (config-cp-instance)#radius-auth-server abc99
Switch (config-cp-instance)#redirect attribute nas-ip enable
Switch (config-cp-instance)#ac-name helix4
Switch (config-cp-instance)#redirect url-head http://172.16.1.26/a70.htm
Switch (config-cp-instance) # portal-server ipv4 abc
```

3. 配置 portal 无感知：

```
Switch (config-cp)#enable
Switch (config-cp)#fast-mac-auth
```

4. 端口上开启 portal 功能：

```
Switch (config)# interface ethernet1/2
Switch (config-if-ethernet1/2)#portal enable configuration 1
```

用户第一次访问网络需要 portal 正常认证，之后即可以 portal 无感知认证。

6.12.7.4 Portal 无感知排错帮助

使用 Portal 无感知功能过程中如遇到问题，请检查是否是以下原因导致：

- ☞ 查看是否已开启 captive-portal 功能；
- ☞ 查看是否已开启 mac 快速认证功能；
- ☞ 配置完成 mac 快速认证功能后如果未生效查看是否向交换机下发了 app 表项。

6.12.8 Portal 逃生

在目前的 Portal 实际组网应用中，存在这样的一个风险，当接入设备与 Portal 服务器的通信中断，会造成新用户无法上线，已经在线的 Portal 用户无法正常下线，接入设备和 Portal 服务器信息不一致引起的计费错误。这些现象会给网络运营商和使用者带来很大的不便。

针对上面的问题，portal 逃生功能提供了一个很好的解决方式，在 Portal 服务器或者 radius 服务器无法正常工作的前提下，让已经在线用户仍然可以正常使用网络，新用户仍然可以上线接入网络，因此 Portal 逃生又分 portal server 逃生和 radius server 逃生。

6.12.8.1 Portal Server 逃生功能

6.12.8.1.1 Portal Server 逃生介绍

Portal server 逃生功能的原理：交换机周期性探测 Portal 服务器，若探测成功则将服务器状态置为 UP，若探测失败 N 次（N 可配置）则将不可达则变成 Down 状态（逃生状态），取消网络的认证限制，允许 Portal 用户不需经过认证即可访问网络资源，并发送状态改变的 Trap 信息和日志。当探测到服务器可达时，恢复服务器状态为 Up 状态（认证状态），并重新开启网络认证限制，不允许未认证的用户访问网络，并发送状态恢复的日志和 Trap。

交换机探测 Portal 服务器状态的具体实现方式是探测 TCP 连接：交换机定期向 Portal 服务器的 portal 服务端口（默认 2000，可配置）发起 TCP 连接。若连接成功建立则表示此服务器的 portal 服务已开启，就认为一次探测成功且服务器可达（状态为 Up）；若连接失败则认为一次探测失败。

探测周期及最大探测失败次数：通过命令行可以配置每次探测之间的时间间隔，此间隔称为探测周期。最大探测失败次数是判断服务器不可达前所必须经历的连续探测失败次数。一次探测失败并不一定认为服务器不可达，而需要查看连续探测失败的次数是否达到配置的最大探测失败次数，若达到此值则认为服务器不可达，否则只是累加，当某次探测成功后，此值会自动清零。探测周期及最大探测失败次数都可以通过命令进行配置。

服务器从可达状态变为不可达状态时触发如下三个操作，管理员可以通过配置选择：

- 发送 Trap：向网管服务器发送 Trap 信息。Trap 信息中记录了 Portal 服务器名以及该服务器状态改变前后的状态等信息。
- 发送日志：向日志服务器发送日志信息。日志信息中记录了 Portal 服务器名以及该服务器状态改变前后的状态等信息。
- Permit-all：也称为 Portal 逃生，表示在 Portal 服务器不可达（down）时，暂时取

消 Portal 认证，允许所有 Portal 用户访问网络资源。

服务器从不可达状态变为可达状态时触发如下三个操作，其中发送 Trap 和发送日志可通过配置选择，关闭 Portal 逃生是系统强制执行的：

- 发送 Trap：向网管服务器发送 Trap 信息。Trap 信息中记录了 Portal 服务器名以及该服务器状态改变前后的状态等信息。
- 发送日志：向日志服务器发送日志信息。日志信息中记录了 Portal 服务器名以及该服务器状态改变前后的状态等信息。
- 关闭 Portal 逃生：Portal 服务器变为可达（up）时，恢复该 VAP 的 Portal 认证功能，新上线的用户必须通过 Portal 认证后才能访问网络资源。

注意：Portal 逃生目前只能实现新旧用户访问网络不受影响，至于用户无法正常下线，目前提供其他方式进行处理，如若 portal server 恢复 UP，接入设备将用户强制下线，保证用户下线处理正常。

6.12.8.1.2 Portal Server 逃生基本配置

Portal 逃生功能的基本配置序列：

1. 打开 Portal 逃生功能，并配置探测周期和最大失败次数。
2. 显示 Portal 服务器的当前连接状态。

1. 打开 Portal 逃生功能，并配置探测周期和最大失败次数

命令	解释
Captive Portal 全局配置模式	
portal-server-detect server-name <name> [interval <interval>] [retry <retries>][action {log permit-all trap }] no portal-server-detect server-name <name>	开启 Portal 逃生功能，并配置（可选）探测周期、最大失败次数和服务器状态发生变化时的操作。该命令的 NO 命令关闭 Portal 逃生功能。

2. 显示 Portal 服务器的当前连接状态

命令	解释
特权模式	
show captive-portal ext-portal-server server-name <name> status	显示 Portal 逃生服务器的状态。

6.12.8.1.3 Portal Server 逃生配置举例

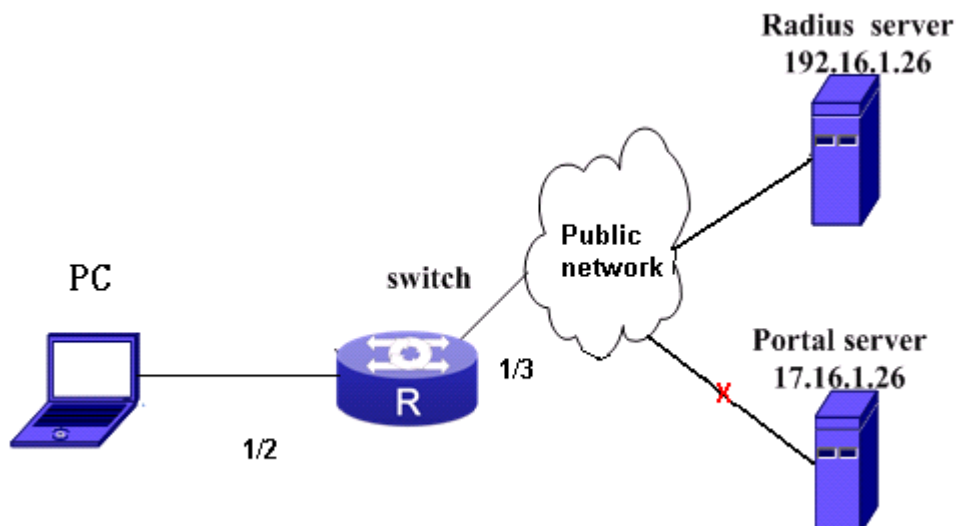


图 6-36 Portal Server 逃生功能典型案例

如上图所示的拓扑，在 Portal server 正常可达的情况下，客户端上线时可以进行正常的 Portal 认证后接入网络。当 portal server 由于本身宕机或者和交换机间的链路断开时，如果交换机没有开启 Portal 逃生功能，客户端将无法认证上线。如果交换机开启 Portal 逃生功能，交换机就能探测到 Portal server 已经不可用，就会启动 Portal 逃生，客户端不需要认证就可以访问网络。如果客户端在 Portal server 断开前就已经通过认证，也不会受到影响，仍然可以访问网络。

具体的配置如下：

27. 全局模式下配置 RADIUS server 相关的认证密钥，认证服务器，计费服务器，aaa 模式等信息：

```
switch (config)#radius-server key 0 test
switch (config)#radius-server authentication host 192.16.1.26
switch (config)#radius-server accounting host 192.16.1.26
switch (config)#aaa-accounting enable
switch (config)#aaa enable
switch (config)#aaa group server radius radius_aaa_1
switch (config-sg-radius)# server 192.16.1.26
switch (config)#interface vlan 192
switch (config-if-vlan1)#ip address 192.16.1.50 255.255.255.0
switch (config)#free-resource 1 destination ipv4 192.16.1.26/32
```

28. portal 实例下 Portal 功能、配置 Portal Server 服务器等信息：

```
Switch (config)#captive-portal
Switch (config-cp)#enable
```

```
Switch(config-cp)# nas-ipv4 192.168.1.50
Switch(config-cp)# external portal-server server-name abc ipv4 172.16.1.26
Switch (config-cp)# configuration 1
Switch (config-cp-instance)#enable
Switch (config-cp-instance)#name helix4
Switch (config-cp-instance)#radius accounting
Switch (config-cp-instance)#radius-acct-server abc99
Switch (config-cp-instance)#radius-auth-server abc99
Switch (config-cp-instance)#redirect attribute nas-ip enable
Switch (config-cp-instance)#ac-name helix4
Switch (config-cp-instance)#redirect url-head http://172.16.1.26/a70.htm
Switch (config-cp-instance) # portal-server ipv4 abc
```

29. 配置 portal 逃生:

```
Switch (config-cp)#enable
Switch (config-cp)# portal-server-detect server-name abc interval 600 retry 2 action log
permit-all trap
```

30. 端口上开启 portal 功能:

```
Switch (config)# interface ethernet1/2
Switch (config-if-ethernet1/2)#portal enable configuration 1
```

以上过程，Portal 服务器 abc 绑定到 CP instance，配置了对该服务器的探测功能，每次探测间隔时间为 600 秒，若连续二次探测均失败，则发送服务器不可达的 Trap 信息和日志信息，并打开 Portal 逃生功能，允许未认证用户访问网络。

6.12.8.1.4 Portal Server 逃生排错帮助

在使用中，如果发现 Portal 逃生功能不能生效，可以采用下面的检查步骤：

- ☞ show captive-portal ext-portal-server server-name <name> status 检查 Portal server 的 Detect Mode 是否为 enable，如果没有 enable，说明没有开启该 Portal server 的逃生功能，需要开启。
- ☞ 如果 Portal 逃生功能已经开启，检查 Detect Operational Status 的状态是否为 down，只有在服务器状态为 down 时，逃生功能才会启动。
- ☞ 如果 Portal 服务器的状态已经为 down，逃生功能仍然不生效，可能是设备出现了问题，需要联系售后工程师处理。

6.12.8.2 Radius Server 逃生功能

6.12.8.2.1 Radius Server 逃生介绍

打开 radius server 逃生功能以后，当所有的 radius 认证服务器不可达时，逃生功能生效，此时 portal 认证客户端的流量放行。当探测到某个认证服务器重新可达时，删除对逃生客户端先前下发的流量放行规则表项。

6.12.8.2.2 Radius Server 逃生基本配置

Radius server 逃生功能的基本配置序列：

1. 打开 radius server 逃生功能
2. 配置对 radius server 进行可达性探测的周期

1. 打开 radius 逃生功能

命令	解释
全局配置模式	
radius-server escape {enable disable }	开启 radius server 逃生功能。

2. 配置对 radius server 进行可达性探测的周期

命令	解释
全局配置模式	
radius-server escape detection-interval {default <second>}	配置对 radius server 进行可达性探测的周期，默认为 180s。

6.12.8.2.3 Radius Server 逃生配置举例

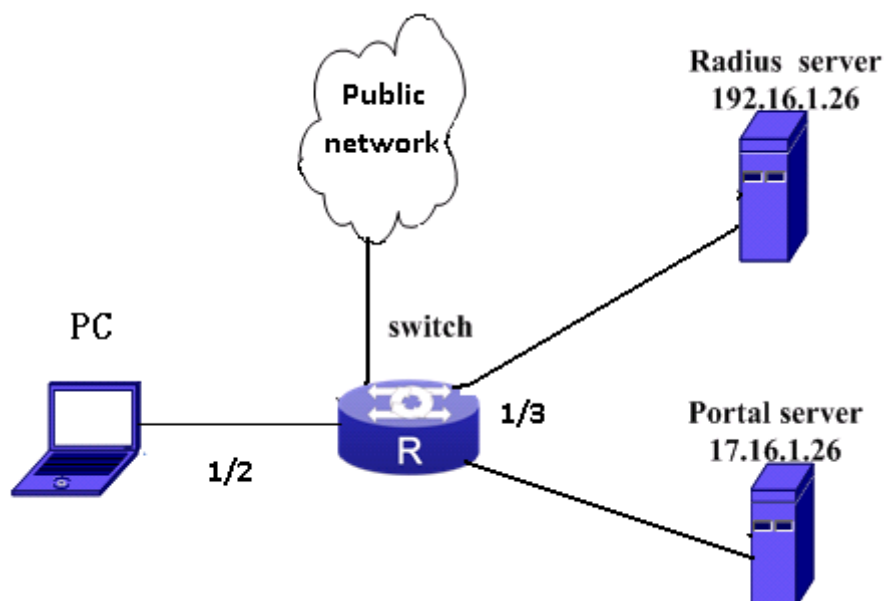


图 6-37 Radius server 逃生功能典型案例

具体的配置如下：

- 1 全局模式下配置 RADIUS server 相关的认证密钥，认证服务器，计费服务器，aaa 模式等信息：

```
switch (config)#radius-server key 0 test
switch (config)#radius-server authentication host 192.16.1.26
switch (config)#radius-server accounting host 192.16.1.26
switch (config)#aaa-accounting enable
switch (config)#aaa enable
switch (config)#aaa group server radius radius_aaa_1
switch config-sg-radius)# server 192.16.1.26
switch(config)#interface vlan 192
switch(config-if-vlan1)#ip address 192.16.1.50 255.255.255.0
switch(config)#free-resource 1 destination ipv4 192.16.1.26/32
```

- 2 portal 实例下 Portal 功能、配置 Portal Server 服务器等信息：

```
Switch (config)#captive-portal
Switch (config-cp)#enable
Switch(config-cp)# nas-ipv4 192.168.1.50
Switch(config-cp)# external portal-server server-name abc ipv4 172.16.1.26
Switch (config-cp)# configuration 1
Switch (config-cp-instance)#enable
```

```
Switch (config-cp-instance)#name helix4
Switch (config-cp-instance)#radius accounting
Switch (config-cp-instance)#radius-acct-server abc99
Switch (config-cp-instance)#radius-auth-server abc99
Switch (config-cp-instance)#redirect attribute nas-ip enable
Switch (config-cp-instance)#ac-name helix4
Switch (config-cp-instance)#redirect url-head http://172.16.1.26/a70.htm
Switch (config-cp-instance) # portal-server ipv4 abc
```

3 配置 radius server 逃生:

```
Switch (config)# radius-server escape enable
Switch (config)# radius-server escape detection-interval 120
```

4 端口上开启 portal 功能:

```
Switch (config)# interface ethernet1/2
Switch (config-if-ethernet1/2)#portal enable configuration 1
```

以上过程每隔 120s 则对 radius server 进行周期性的可达性探测。

6.12.8.2.4 Radius Server 逃生排错帮助

在使用中, 如果发现 radius server 逃生功能不生效, 可以采用下面的检查步骤:

- ☞ 检查是否开启了 radius server 逃生功能;
- ☞ 如果开启了该功能, 通过 show aaa config 获得 Retransmit 和 Time Out 这两项参数值, 然后检查配置的 detection-interval 是否大于 $(\text{Retransmit}+1)*\text{Time Out}$ 所获得的值, 如果小于, 则逃生失败;
- ☞ 如果 show aaa config 显示 Is Server live by radius escape function =0 说明逃生中, 但是功能不生效, 则需要联系售后工程师处理。

6.13 QoS

6.13.1 QoS

6.13.1.1 QoS 介绍

QoS (Quality of Service, 服务品质保证) 是指一个网络能够利用各种各样的技术向选定的网络通信提供更好的服务的能力。QoS 是服务品质保证, 提供稳定、可预测的数据传

送服务，来满足使用程序的要求，QoS 不能产生新的带宽，而是根据应用的需求以及网络管理的设置来有效的管理网络带宽。

6.13.1.1.1 QoS 术语

QoS: Quality of Service 服务品质保证，提供稳定、可预测的数据传送服务，来满足使用程序的要求，QoS 不能产生新的带宽，而是根据使用程序的需求以及网络管理的设置来有效的管理网络带宽。

QoS 功能域: QoS Domain 支持 QoS 的设备组成的一个能够提供服务品质保证的网络拓扑，定义为 QoS 功能域。

CoS: Class of Service 服务级别，L2 802.1Q 帧携带的分类信息，在帧头的 Tag 字段中占 3bits，称为用户优先级，范围为 0~7。

Layer 2 802.1Q/P Frame

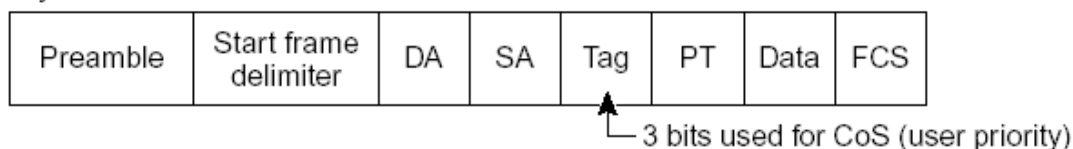


图 6-38 CoS 优先级

ToS: Type of Service 服务类型，L3 IPv4 包头携带的一个字节的字段，标记 IP 包的服务类型，ToS 字段内可以是 IP Precedence 值，也可以是 DSCP 值。

Layer 3 IPv4 Packet

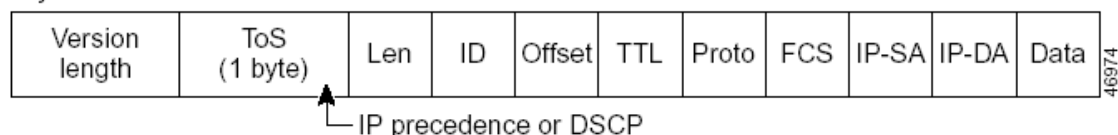


图 6-39 ToS 优先级

IP Precedence: IP 优先级，L3 IP 包头携带的分类信息，共占 3bits，范围为 0~7。

DSCP: Differentiated Services Code Point 差别化业务编码点，L3 IP 包头携带的分类信息，共占 6bits，范围为 0~63，向下兼容 IP Precedence。

MPLS TC(EXP):



MPLS 报文中的一个字段，用来表明 MPLS 报文的服务等级，一共三个 bit，范围为 0~7。

Internal Priority: 交换芯片内部优先级设置，支持范围和芯片相关，后续文档中简写为 Int-Prio 或者 IntP。

Drop Precedence: 丢弃优先级，在交换芯片处理报文的时候，丢弃优先级大的报文将会

被优先丢弃,三色算法取值范围为 0-2,双色算法取值范围 0-1,后续文档中简写为 Drop-Prec 或者 DP。

分类: QoS 的入口动作,根据数据包携带的分类信息和 ACLs,将数据包流量进行分类。

监管: QoS 的入口动作,制定监管策略,对分类后的数据包进行监管。

重写: QoS 的入口动作,根据制定的监管策略,对数据包进行通过、降级、丢包等操作。

排程: QoS 的出口动作,配置八条出口队列的 WRR (Weighted Round Robin) 权重。

In-Profile: 对于在 QoS 监管策略规定范围(带宽或突发值)内的流量,我们称它是 In- Profile 的。

Out-of-Profile: 对于超出 QoS 监管策略规定范围(带宽或突发值)的流量,我们称它是 Out-of-Profile 的。

6.13.1.1.2 QoS 实现

对于 L3 交换机软件 QoS 的实现,首先应该给出一个参考模型,而这个模型要求是通用的,成熟的。QoS 并不能产生新的带宽,但是它可以将现有的带宽资源做一个最佳的调整和配置,完整的应用 QoS,可以对网络的数据传输做到完全的控管。下面将对 QoS 做一个尽量严格的描述:

IP 协议的数据传送规范,只针对发送端和接收端的地址和服务等作出规定,并且利用 OSI 四层以上的,如 TCP 协议等来确认数据包的传送正确无误,但是,IP 协议没有提供和保护传输数据包的带宽,而是以尽力而为(Best Effort)的方式来提供带宽服务。此种方式对于 Mail 以及 FTP 等服务尚可以接受,但是对于越来越多的多媒体业务数据和电子商务数据的传输,则无法满足这些应用需要的带宽和低延迟要求。

基于差别化服务的 QoS 在网络的入口指定每个数据包的优先级别,这些分类信息被携带在 L3 的 IP 包头或者 L2 的 802.1Q 帧头中。QoS 对于相同优先级别的数据包提供相同服务,而对不同级别的数据包则提供不同的操作。支持 QoS 的交换机或路由器可以根据数据包的分类信息,为其提供不同的带宽资源,并且可以根据配置的监管策略来重写这些数据包携带的分类信息甚至在带宽紧张的时候丢弃某些低级别的数据包。

如果在一个网络中,每一跳的设备都支持基于差别化服务的 QoS,那么就可以构建一个端到端的 QoS 解决方案。QoS 的配置是很灵活的,可以很复杂,也可以很简单,这一切都依赖于实际的网络拓扑和网络设备,以及对网络出入流量的分析。

6.13.1.1.3 QoS 基本模型

QoS 的基本模型分为四个部分: Classification (分类)、Policing (监管)、Mark (重写)和 Scheduling (排程),其中分类、监管和重写是顺序执行的 QoS 的入口行为,整形和排程是 QoS 的出口行为。

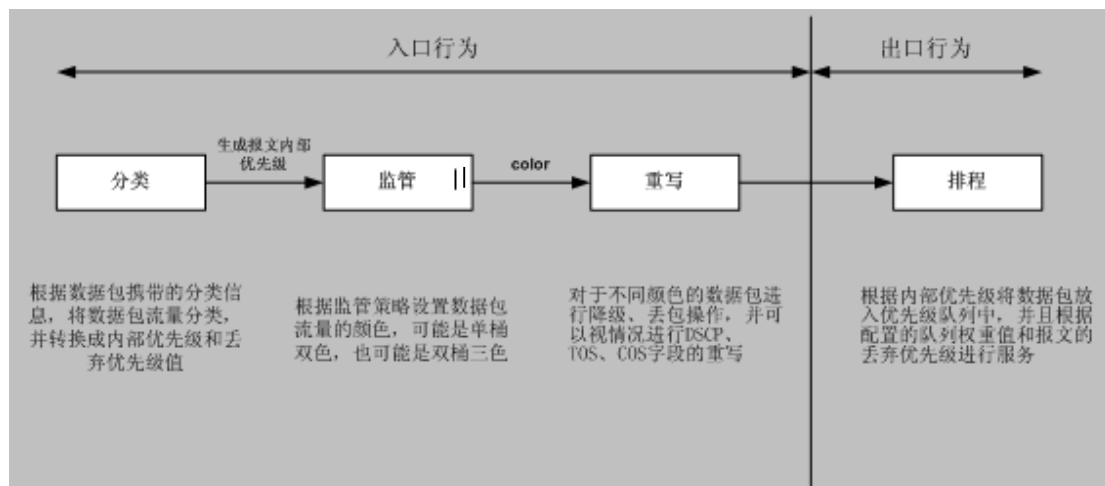


图 6-40 QoS 基本模型

分类：通过检查数据包的分类信息，来区别不同的流量，并且根据数据包的分类信息生成内部优先级和丢弃优先级。对于不同类型的数据包和不同的交换机配置，分类有不同的处理方式，由下面的流程图详细说明：

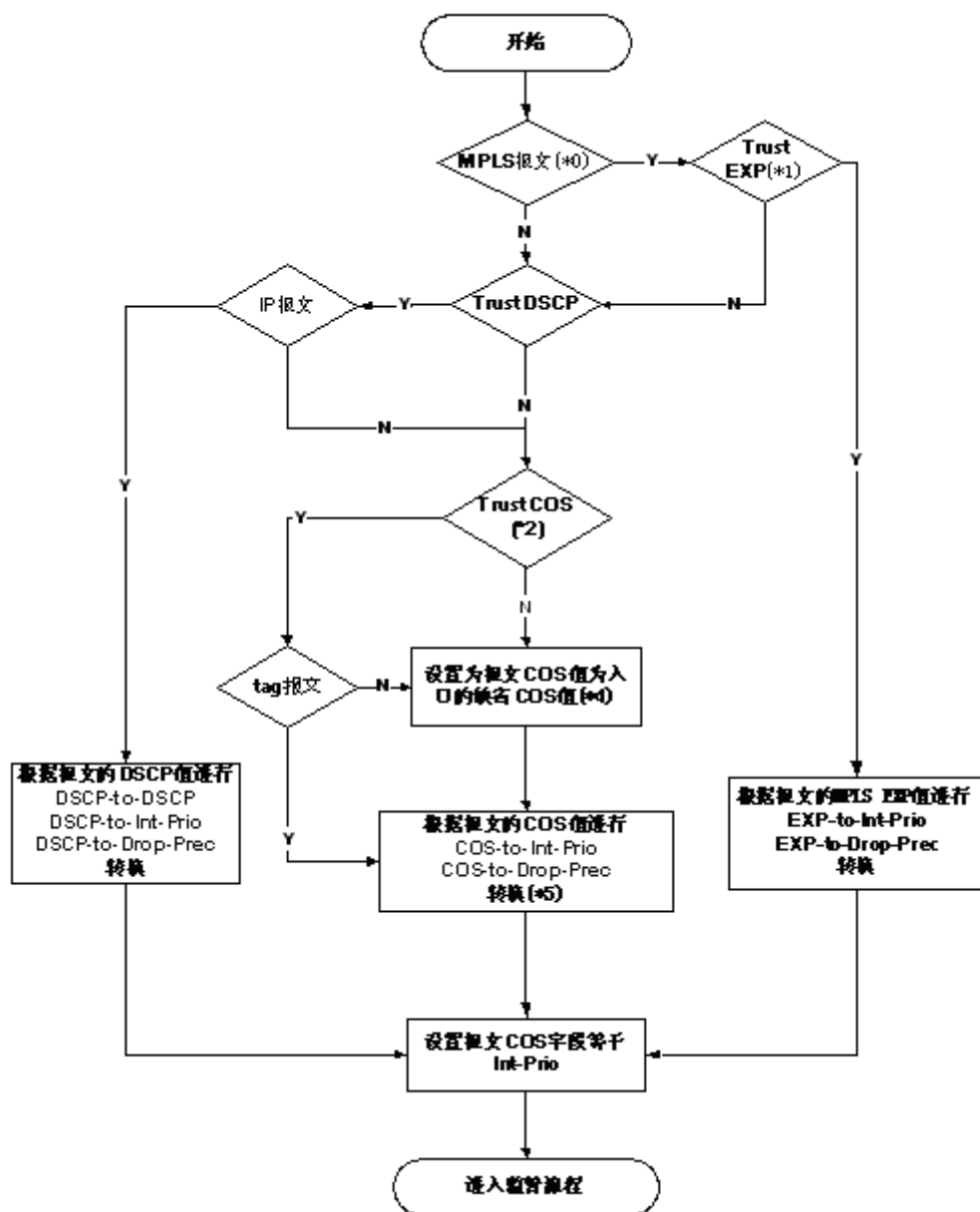


图 6-41 分类流程

监管和重写：在入口的流量经过分类，并且每个数据包都被分配了一个内部优先级和丢弃优先级值以后，就可以进行监管和重写。

监管行为可以基于流配置不同的监管策略，为分类后的流量分配带宽，带宽的分配策略可以是双桶双色，也可以是双桶三色。流量会被分配不同的颜色（color），对不同颜色的流量可以选择丢包、通过两种处理方式；对于通过处理方式的数据包，可以附加重写动作。重写行为即用一个新的优先级较低的 DSCP 值来代替数据包原有的级别较高的 DSCP 值。监管和重写的具体行为，由下面的流程图详细说明：

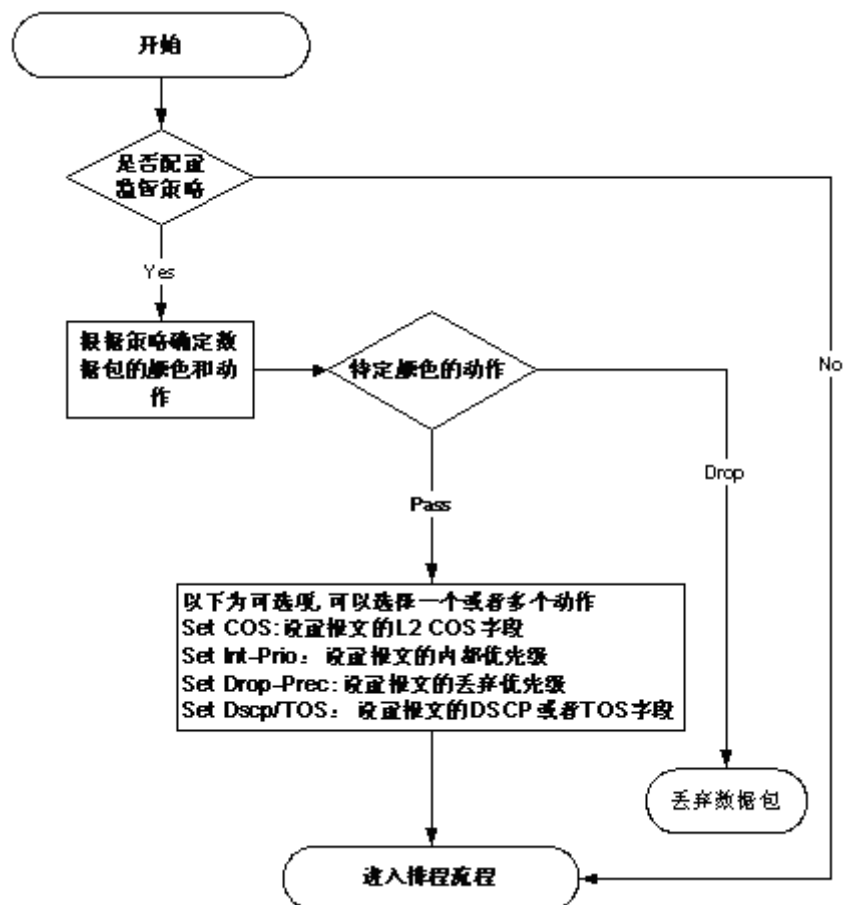


图 6-42 监管和重写流程

整形和排程：出口的数据包会都具有内部优先级值和丢弃优先级。整形行为根据数据包的内部优先级值将其分配到不同的优先级队列中，而排程行为根据配置的优先级队列权重和丢弃优先级来进行数据包转发服务。整形和排程的具体行为，由下面的流程图详细说明：

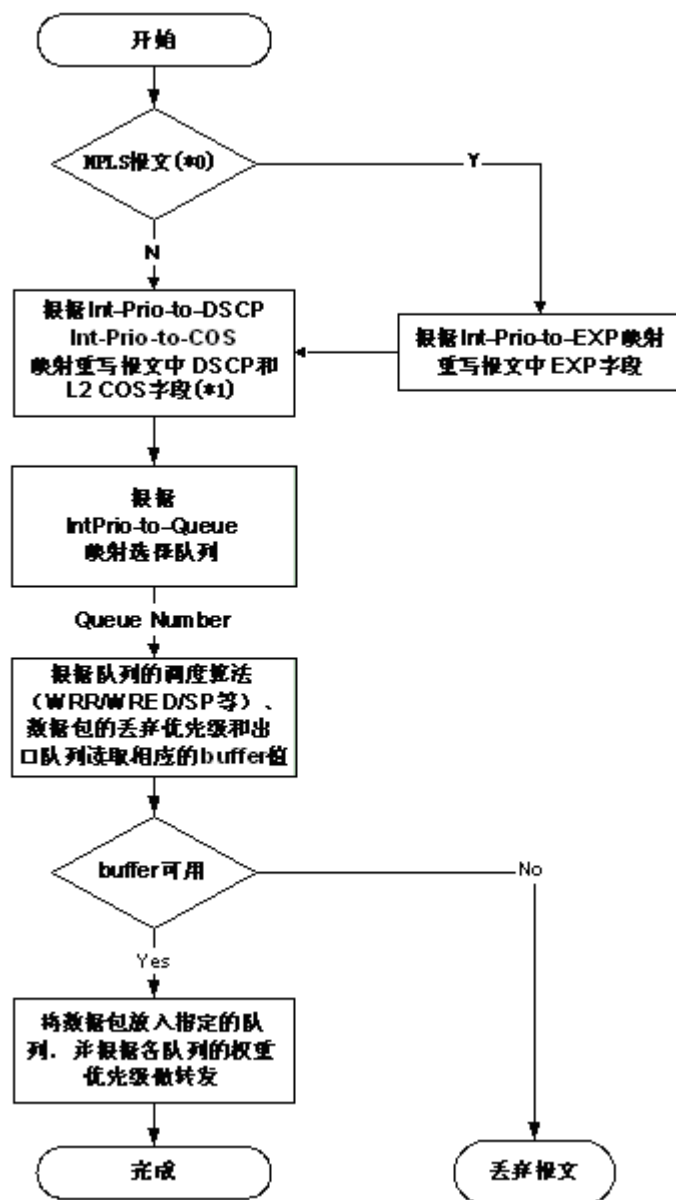


图 6-43 整形和排程流程

6.13.1.2 QoS 配置

QoS 配置任务序列如下：

配置分类表（classmap）

建立一个分类规则，可以按照 ACL、CoS、VLAN ID、IPV4 Precedent、DSCP、IPV6 FL 来分类。之后，对于不同类别的数据流采取不同的策略。

配置策略表（policymap）

对数据流进行分类之后，就可以建立一个策略表，之后就可以将其对应一个先前建立的 class-map，进入策略分类表模式，然后就可以对于不同的数据流采取不同的策略，如带宽

限制、优先级降低、分配新的 DSCP 值等等。也可以定义一个集合策略，这个策略可以在同一个策略表内部被多个策略分类表使用。

将 QoS 应用到端口或 VLAN

配置端口的信任模式，或者绑定策略。策略只有绑定到具体的端口，才在此端口生效。

策略也可以绑定到具体的 VLAN 上。

建议不要在 VLAN 和属于该 VLAN 的端口上同时使用 policy map。

配置端口队列调度算法

设定端口队列调度算法，如 sp, wrr, wdrp 等。

设定端口队列丢弃算法，如 wred, tail 等。

配置 QoS 映射关系

配置 cos 到 intp 或 dp, dscp 到 dscp 或 intp 或 dp, exp 到 intp 或 dp, intp 到 dscp 或 cos 或 exp 的映射关系。

1. 配置分类表 (classmap)

命令	解释
全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立一个 class-map (分类表)，并进入 class-map 模式；本命令的 no 操作为删除指定的 class-map。
match {access-group <acl-index-or-name> ip dscp <dscp-list> ip precedence <ip-precedence-list> ipv6 access-group <acl-index-or-name> ipv6 dscp <dscp-list> ipv6 flowlabel <flowlabel-list> vlan <vlan-list> cos <cos-list> exp <exp-list>} no match {access-group ip dscp ip precedence ipv6 access-group ipv6 dscp ipv6 flowlabel vlan cos exp}	设置分类表中的匹配标准，可以根据 ACL、CoS、VLAN ID、IPV4 Precedent、DSCP、IPV6 FL 优先级等对数据流进行分类；本命令的 no 操作为删除指定的匹配标准。

2. 配置策略表 (policymap)

命令	解释
全局配置模式	
policy-map <policy-map-name> no policy-map <policy-map-name>	建立一个 policy-map (策略表)，并进入 policy-map (策略表) 模式；本命令

	令的 no 操作为删除指定的 policy-map。
class <class-map-name> [insert-before <class-map-name>] no class <class-map-name>	建立一个 policy-map（策略表）之后，可以将其对应于一个 class（策略分类表），并进入策略分类表模式，之后就可以对不同的数据流采取不同的策略或者分配一个新的 DSCP 值；本命令的 no 操作为删除指定策略分类表。
set {ip dscp <new-dscp> ip precedence <new-precedence> internal priority <new-inp> drop precedence <new-dp> cos <new-cos>} no set {ip dscp ip precedence internal priority drop precedence cos }	为分类后的流量分配一个新的 DSCP、CoS 和 IP Precedence 值等；本命令的 no 操作为取消分配新的值。
单桶模式： policy <bits_per_second> <normal_burst_bytes> ({conform-action ACTION exceed-action ACTION}) 双桶模式： policy <bits_per_second> <normal_burst_bytes> [pir <peak_rate_bps>] <maximum_burst_bytes> [{conform-action ACTION exceed-action ACTION violate-action ACTION }] ACTION 定义： drop transmit set-dscp-transmit <dscp_value> set-prec-transmit <ip_precedence_value> set-cos-transmit <cos_value> set-internal-priority <inp_value> set-Drop-Precedence <dp_value> no policy	为分类后的流量配置一个策略，支持三色color的非聚合Policy命令，根据配置参数解析出令牌桶的工作模式，是单速单桶、单速双桶还是双速双桶，并且给不同颜色报文设置相应动作。no命令删除模式配置。单桶模式只有特定的交换机支持。
policy aggregate <aggregate-policy-name> no policy aggregate <aggregate-policy-name>	为分类后的流量应用一个聚合策略；本命令的 no 为删除指定的聚合策略。

accounting no accounting	为分类后的流量设置统计功能。 在策略分类表配置模式下将功能开启后,会对策略分类表中的流量增加统计功能,单桶模式,报文经过 policy 时仅会有绿色和红色两种颜色,打印统计信息时 in-profile 表示绿色, out-profile 表示红色;双桶模式下,会有三种颜色的报文, in-profile 表示绿色, out-profile 表示红色和黄色。
策略分类表配置模式	
drop no drop transmit no transmit	对分类后的流量可以选择丢弃、通过动作;本命令的 no 操作为取消分配的动作。

3. 将 QoS 应用到端口或 VLAN 接口

命令	解释
接口配置模式	
mls qos trust {cos dscp} no mls qos trust {cos dscp}	配置交换机端口信任状态;本命令的 no 操作为禁止交换机端口的当前信任状态。
mls qos cos {<default-cos> } no mls qos cos	配置交换机端口的缺省 CoS 值;本命令的 no 操作为恢复缺省情况。
service-policy input <policy-map-name> no service-policy input {<policy-map-name>}	在端口上应用一个策略表;本命令的 no 操作为删除应用到交换机端口的某个指定策略表。目前在出口不支持出口策略表或删除该端口入方向应用的所有策略表。
全局配置模式	
service-policy input <policy-map-name> vlan <vlan-list> no service-policy input {<policy-map-name>} vlan <vlan-list>	在交换机 VLAN 接口上应用一个策略表;本命令的 no 操作为删除应用到交换机 VLAN 接口的某个指定策略表或删除该 vlan 接口入方向应用的所有策略表。

4. 配置端口队列调度算法和权重

命令	解释
端口配置模式	
mls qos queue algorithm {sp wrr wdr} no mls qos queue algorithm	端口配置队列调度算法。端口缺省队列调度算法为 wrr。
mls qos queue wrr weight <weight0..weight7> no mls qos queue wrr weight	基于端口配置队列权重。缺省队列权重为 1 2 3 4 5 6 7 8。
mls qos queue wdr weight <weight0..weight7> no mls qos queue wdr weight	基于端口配置队列权重。缺省队列权重为 10 20 40 80 160 320 640 1280。
mls qos queue <queue-id> bandwidth <minimum-bandwidth> <maximum-bandwidth> no mls qos queue <queue-id> bandwidth	基于端口配置带宽保证值。

5. 配置端口队列丢弃算法

命令	解释
全局配置模式	
mls qos queue drop-algorithm {wred tail} no mls qos queue drop-algorithm	端口配置队列丢弃算法。端口缺省队列丢弃算法为 tail，尾部丢弃算法。
mls qos queue <queue-id> wred dp<dp> drop-startpoint <drop-startpoint> drop-endpoint <drop-endpoint> max-drop-rate <drop-rate> no mls qos queue <queue-id> wred [dp <dp>]	配置端口 wred 算法参数。

6. 配置 QoS 映射关系

命令	解释
全局配置模式	
mls qos map (cos-intp <intp1...intp8> cos-dp <dp1...dp8> dscp-dscp <in-dscp list> to <out-dscp> dscp-intp <in-dscp	设置 QoS 的优先级映射，no 命令恢复缺省映射值。

<pre>list> to <intp> dscp-dp <in-dscp list> to <dp>) no mls qos map (cos-intp cos-dp dscp-dscp dscp-intp dscp-dp) mls qos map (intp-dscp <dscp1..dscp8> intp-cos <cos1..cos8>) no mls qos map (intp-dscp intp-cos)</pre>	

7. 清除特定端口或者 VLAN 的 Accounting 数据

命令	解释
特权配置模式	
<pre>clear mls qos statistics [in out] [interface <interface-name> vlan <vlan-id>]</pre>	清除特定端口或者 vlan Policy Map 的 Accounting 数据，如果不带参数，清除所有的 Policy Map 的 Accounting 数据。

8. 显示 QoS 的配置信息

命令	解释
特权配置模式	
<pre>show mls qos maps [cos-intp cos-dp dscp-dscp dscp-intp dscp-dp intp-cos intp-dscp]</pre>	显示 QoS 全局映射相关配置内容。
<pre>show class-map [<class-map-name>]</pre>	显示 QoS 的分类表信息。
<pre>show policy-map [<policy-map-name>]</pre>	显示 QoS 的策略表信息。
<pre>show mls qos {interface [<interface-id>] [policy queuing] vlan <vlan-id>}</pre>	显示 QoS 在交换机端口上的配置信息。
<pre>show mls qos interface <interface-id> wred [queue <queue-id>] [dp<dp>]</pre>	显示 QoS 在交换机端口上的 WRED 参数配置信息。
<pre>show mls qos queue statistics <interface <name>></pre>	显示接口的队列统计信息。

6.13.1.3 QoS 举例

案例 1:

启动 QoS 功能，将端口 ethernet 1/1 出口队列的权重改为 1:1:2:2:4:4:8:8，队列丢弃算法设置成 wred，然后再配置成信任 cos 模式，但是不改变报文里的 DSCP 值，并且将此端口的

默认 cos 值设为 5。

配置步骤如下：

```
Switch#config
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch(Config-If-Ethernet1/1)#mls qos queue wrr weight 1 1 2 2 4 4 8 8
```

```
Switch(Config-If-Ethernet1/1)#mls qos queue drop-algorithm wred
```

```
Switch(Config-If-Ethernet1/1)#mls qos trust cos
```

```
Switch(Config-If-Ethernet1/1)#mls qos cos 5
```

配置结果：

全局启动了 QoS 功能，端口 ethernet 1/1 出口带宽的比例依次为 1:1:2:2:4:4:8:8，使用 wred 丢弃算法，将有效改善 TCP 连接饿死现象。当从 ethernet 1/1 进来的报文带 cos 值时，根据 cos 值到出口队列的映射关系，cos 值 0—7 分别对应出口队列 1, 2, 3, 4, 5, 6, 7, 8，放到不同的优先队列，如果进来的报文不带 cos 值，则将其 cos 值设为 5，根据对应关系，放入优先级队列 6。所有经过的报文不改变其携带的 DSCP 值。

案例 2：

在端口 ethernet1/2 上，将属于网段 192.168.1.0 内的报文带宽限制为 10M 比特/秒，突发值设为 4M 字节，超过带宽的该网段内的报文一律丢弃。

配置步骤如下：

```
Switch#config
```

```
Switch(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

```
Switch(config)#class-map c1
```

```
Switch(Config-ClassMap-c1)#match access-group 1
```

```
Switch(Config-ClassMap)# exit
```

```
Switch(config)#policy-map p1
```

```
Switch(Config-PolicyMap-p1)#class c1
```

```
Switch(Config-PolicyMap-p1-Class-c1)#policy 10000 4000 exceed-action drop
```

```
Switch(Config-PolicyMap-p1-Class-c1)#exit
```

```
Switch(Config-PolicyMap-p1)#exit
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(Config-If-Ethernet1/2)#service-policy input p1
```

配置结果：

先设置一个 ACL：1，匹配网段 192.168.1.0。全局启动 QoS 功能，创建一个 class-map：c1，在 class-map 中匹配 ACL1，再创建一个 policy-map：p1，在 P1 中引用 c1，设置对

应的 policy 来限制带宽和峰值。然后在端口 ethernet 1/2 应用此 policy-map。设置后，网段 192.168.1.0 内的报文带宽在端口 ethernet 1/2 上被限制为 10M 比特/秒，突发值为 4M 字节，超过带宽的该网段内的报文一律丢弃。

案例 3:

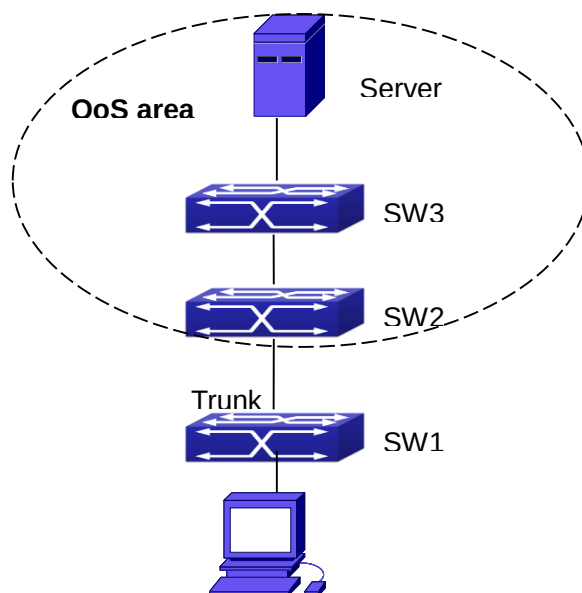


图 6-44 QoS 典型拓扑

如图，虚线框内组成一个 QoS 域，SW1 将不同的流量分类，分配不同的 IP 优先级，如在端口 ethernet 1/1 上将属于网段 192.168.1.0 内的报文的 IP 优先级设为 5，与 SW2 相连的端口为 trunk 口。在 SW2 上，将与 SW1 相连的端口 ethernet 1/1，设置信任 cos。这样在 QoS 域的内部，不同优先级的报文就走不同的队列，分配不同的带宽。

配置步骤如下：

交换机 1 上的 QoS 配置：

```
Switch#config
Switch(config)#access-list 1 permit 192.168.1.0 0.0.0.255
Switch(config)#class-map c1
Switch(Config-ClassMap-c1)#match access-group 1
Switch(Config-ClassMap-c1)# exit
Switch(config)#policy-map p1
Switch(Config-PolicyMap-p1)#class c1
Switch(Config-PolicyMap-p1-Class-c1)#set ip precedence 5
Switch(Config-PolicyMap-p1-Class-c1)#exit
Switch(Config-PolicyMap-p1)#exit
```

```
Switch(config)#interface ethernet 1/1
Switch(Config-If-Ethernet1/1)#service-policy input p1
```

交换机 2 上的 QoS 配置:

```
Switch#config
Switch(config)#interface ethernet 1/1
Switch(Config-If-Ethernet1/1)#mls qos trust cos
```

6.13.1.4 QoS 排错帮助

- ☞ trust COS 和 exp 配置可以和其他的 trust 一起生效，也可以和 Policy Map 一起使用。
- ☞ trust dscp 配置可以和其他的 trust 一起生效，也可以和 Policy Map 一起使用。本配置对 IPv4 和 IPv6 报文都生效。
- ☞ Trust EXP、Trust DSCP 和 Trust COS 可以同时配置，他们之间的优先级如下：EXP>DSCP>COS
- ☞ 如果配置了动态 vlan（mac vlan/voice vlan/ip subnet vlan/protocol vlan），报文的 COS 值等于动态 vlan 的 COS 值。
- ☞ 目前策略表只支持绑定到入口，对出口不支持。
- ☞ 目前建议不要在 vlan 和属于该 vlan 的端口上同时使用 policy map。

6.13.2 PBR

6.13.2.1 PBR 介绍

PBR（Policy-Based Routing，策略路由）是根据 IP 包的源地址、目的地址、IP 优先级、TOS 值、IP 协议、源端口号、目的端口号等信息来策略性的指定数据包转发的下一跳的一种方法。

6.13.2.2 PBR 配置

1. 配置一个 class-map（分类表）
2. 设置分类表中的匹配标准
3. 配置一个 policy-map（策略表）
4. 配置一个策略表对应一个 class（分类表）
5. 配置下一跳 IPv4 地址
6. 配置端口捆绑策略表
7. 配置 VLAN 捆绑策略表

1. 配置一个 class-map（分类表）

命令	解释

全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立或删除一个 class-map（分类表）。

2. 设置分类表中的匹配标准

命令	解释
class-map（分类表）配置模式	
match ip {access-group <acl-index-or-name>} no match ip {access-group }	设置分类表中的匹配标准。

3. 配置一个 policy-map（策略表）

命令	解释
全局配置模式	
policy-map <policy-map-name> no policy-map <policy-map-name>	建立或删除一个 policy-map（策略表）。

4. 配置一个策略表对应一个 class（分类表）

命令	解释
策略表配置模式	
class <class-map-name> no class <class-map-name>	对应一个 class（分类表），并进入策略分类表模式。

5. 配置下一跳 IPv4 地址

命令	解释
策略分类表配置模式	
set ipv4 [default] nexthop [vrf <vrf>] <nexthop-ip> no set ipv4 nexthop	为分类后的流量设置下一跳 IP；本命令的 no 操作为取消分配新的值。

6. 配置端口捆绑策略表

命令	解释
端口配置模式	

service-policy {input <policy-map-name> output <policy-map-name>} no service-policy {input <policy-map-name> output <policy-map-name>}	在具体端口上应用策略表；每个端口在每个方向上只能应用一个策略表。目前在出口不支持出口策略表。

7.配置 VLAN 捆绑策略表

命令	解释
全局配置模式	
service-policy input <policy-map-name> vlan <vlan-list> no service-policy input <policy-map-name> vlan <vlan-list >	在具体 VLAN 上绑定策略表。本命令的 no 操作为删除应用到交换机 VLAN 接口的某个指定策略表。

6.13.2.3 PBR 举例

案例：

在端口 ethernet 1/1 上，设置源 IP 为 192.168.1.0/24 网段内的报文进行策略路由，其下一跳为 218.31.1.119。同时，该网络的内网 IP 是在 192.168.0.0/16 范围。为了保证内网正常通信，对于源 IP 为 192.168.1.0/24，目的 IP 为内网 IP 192.168.0.0/16 范围内的报文不进行策略路由，本设备 192.168.1.0/24 段的接口地址为 192.168.1.1。

配置步骤如下：

```
Switch#config
Switch(config)#ip access-list extended a1
Switch(Config-IP-Ext-Nacl-a1)# permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.255.255
Switch(Config-IP-Ext-Nacl-a1)#exit
Switch(config)#ip access-list extended a2
Switch(Config-IP-Ext-Nacl-a1)# permit ip 192.168.1.0 0.0.0.255 any-destination
Switch(Config-IP-Ext-Nacl-a1)#exit
Switch(config)#mls qos
Switch(config)#class-map c1
Switch(Config-ClassMap-c1)#match access-group a1
Switch(Config-ClassMap-c1)# exit
Switch(config)#class-map c2
Switch(Config-ClassMap-c2)#match access-group a2
```

```
Switch(Config-ClassMap-c2)# exit
Switch(config)#policy-map p1
Switch(Config-PolicyMap-p1)#class c1
Switch(Config-PolicyMap-p1-Class-c1)#set ip nexthop 192.168.1.1
Switch(Config-PolicyMap-p1-Class-c1)#exit
Switch(Config-PolicyMap-p1)#class c2
Switch(Config-PolicyMap-p1-Class-c2)#set ip nexthop 218.31.1.119
Switch(Config-PolicyMap-p1-Class-c2)#exit
Switch(config)#interface ethernet 1/1
Switch(Config-If-Ethernet1/1)#service-policy input p1
```

配置结果：

先设置两个包含两个项的 ACL a1、a2，a1 匹配源 IP 网段 192.168.1.0/24 和目的地 IP 网段 192.168.0.0/16，a2 匹配源 IP 网段 192.168.1.0/24。全局启动 QoS 功能，创建两个 class-map：c1 和 c2，在 class-map c1 中匹配 ACL a1，class-map c2 中匹配 ACL a2。再创建一个 policy-map：p1，在 P1 中引用 c1，设置本设备 192.168.1.0/24 段的接口地址为 192.168.1.1，设置下一跳 IP 为 218.31.1.119。然后在端口 ethernet 1/1 应用此 policy-map。设置后，端口 ethernet 1/1 上对于源 IP 为网段 192.168.1.0/24 的报文，除目的 IP 为网段 192.168.0.0/16 的报文进行正常的 L3 转发之外，其余的报文经 218.31.1.119 转发。

6.13.3 IPv6 PBR

6.13.3.1 PBR（基于策略的路由）功能介绍

基于策略的路由为网络管理者提供了比传统路由协议对报文的转发和存储更强的控制能力，传统上，路由器用从路由协议派生出来的路由表，根据目的地址进行报文的转发，基于策略的路由比传统路由强，使用更灵活，它使网络管理者不仅能够根据目的地址而且能够根据报文的大小，应用或 IP 源地址来选择转发路径。策略可以定义为通过多路由器的负载均衡或根据总流量在各线上进行转发的服务质量（QOS）。

PBR（Policy-Based Routing，策略路由）是根据 IP 包的源地址、目的地址、IP 优先级、TOS 值、IP 协议、源端口号、目的端口号等信息来策略性的指定数据包转发的下一跳的一种方法。

6.13.3.2 PBR 配置任务序列

1. 配置一个 class-map（分类表）
2. 设置分类表中的匹配标准

3. 配置一个 policy-map（策略表）
4. 配置一个策略表对应一个 class（分类表）
5. 配置下一跳 IPv6 地址
6. 配置端口捆绑策略表
7. 配置 VLAN 捆绑策略表

1. 配置一个 class-map（分类表）

命令	解释
全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立或删除一个 class-map（分类表）。

2. 设置分类表中的匹配标准

命令	解释
class-map（分类表）配置模式	
match ipv6 {access-group <acl-index-or-name>} no match ipv6 {access-group }	设置分类表中的匹配标准。

3. 配置一个 policy-map（策略表）

命令	解释
全局配置模式	
policy-map <policy-map-name> no policy-map <policy-map-name>	建立或删除一个 policy-map（策略表）。

4. 配置一个策略表对应一个 class（分类表）

命令	解释
策略表配置模式	
class <class-map-name> no class <class-map-name>	对应一个 class（分类表），并进入策略分类表模式。

5. 配置下一跳 IPv6 地址

命令	解释
策略分类表配置模式	
set ipv6 [default] nexthop [vrf <vrf>] <nexthop-ip> no set ipv6 nexthop	为分类后的流量设置下一跳 IP；本命令的 no 操作为取消分配新的值。

6. 配置端口捆绑策略表

命令	解释
端口配置模式	
service-policy {input <policy-map-name> output <policy-map-name>} no service-policy {input <policy-map-name> output <policy-map-name>}	在具体端口上应用策略表；每个端口在每个方向上只能应用一个策略表。目前在出口不支持出口策略表。

7. 配置 VLAN 捆绑策略表

命令	解释
全局配置模式	
service-policy input <policy-map-name> vlan <vlan-list> no service-policy input <policy-map-name> vlan <vlan-list>	在具体 VLAN 上绑定策略表。本命令的 no 操作为删除应用到交换机 VLAN 接口的某个指定策略表。

6.13.3.3 PBR 举例

案例：

在端口 ethernet 1/1 上，本设备默认网关地址为 3000::1，设置源 IP 为 2000::/64 网段内的报文进行策略路由，其下一跳地址为 3100::2。

配置步骤如下：

```
Switch#config
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ipv6 address 2000::1/64
Switch(Config-if-Vlan1)#ipv6 neighbor 2000::2 00-00-00-00-00-01 interface Ethernet 1/1
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ipv6 address 3000::1/64
Switch(Config-if-Vlan2)#ipv6 neighbor 3000::2 00-00-00-00-00-02 interface Ethernet 1/2
Switch(config)#interface vlan 3
Switch(Config-if-Vlan3)#ipv6 address 3100::1/64
Switch(Config-if-Vlan3)#ipv6 neighbor 3100::2 00-00-00-00-00-03 interface Ethernet 1/5
Switch(config)# ipv6 access-list extended b1
```

```

Switch(Config-IPv6-Ext-Nacl-b1)# permit 2000::/64 any-destination
Switch(Config-IPv6-Ext-Nacl-b1)#exit
Switch(config)#class-map c1
Switch(config-ClassMap)#match ipv6 access-group b1
Switch(config-ClassMap)#exit
Switch(config)#policy-map p1
Switch(config-PolicyMap)#class c1
Switch(config-Policy-Class)#set ipv6 nexthop 3100::2
Switch(config--Policy-Class)#exit
Switch(config-PolicyMap)#exit
Switch(config)#interface ethernet 1/1
Switch(Config-Ethernet1/1)#service-policy input p1

```

配置结果：

先设置一个包含一个项的 ACL b1，匹配源 IP 网段 2000::/64（允许）。全局启动 QoS 功能，创建一个 class-map：c1，在 class-map 中匹配 ACL b1，再创建一个 policy-map：p1，在 p1 中引用 c1，设置下一跳 IP 为 3100::2。然后在端口 ethernet 1/1 应用此 policy-map。设置后，端口 ethernet 1/1 上对于源 IP 为网段 2000::/64 的报文经 3100::2 转发。

6.13.3.4 PBR 排错帮助

- ☞ 目前策略表只支持绑定到入口，对出口不支持。
- ☞ 受硬件资源限制，如果策略过于复杂，配置不上，会提示用户相关信息。

6.14 MPLS QoS

6.14.1 MPLS QoS 介绍

MPLS（MultiProtocol Lable Switch）多协议标签交换中的 exp 字段提供了对 QoS 的支持，通过该字段可以为网络通信提供更好的服务。

6.14.1.1 MPLS QoS 术语

CoS：Class of Service 服务级别，L2 802.1Q 帧携带的分类信息，在帧头的 Tag 字段中占 3bits，称为用户优先级，范围为 0~7。

DA&SA(12)	0x8100	CoS	CFI	vlan	Type(2)	DATA



图 6-45 CoS 优先级

DSCP: Differentiated Services Code Point 差别化业务编码点, L3 IP 包头携带的分类信息, 共占 6bits, 范围为 0~63, 向下兼容 IP Precedence。



图 6-46 MPLS EXP 优先级

MPLS 报文中的一个字段, 用来表明 MPLS 报文的等级, 一共三个 bit, 范围为 0—7。内部 DSCP: 交换机内部优先级设置, 用来区分交换机内部数据报文的优先级, 取值范围为 0—63。

In-Profile: 对于在 QoS 监管策略规定范围(带宽或突发值)内的流量, 我们称它是 In- Profile 的。

Out-of-Profile: 对于超出 QoS 监管策略规定范围(带宽或突发值)的流量, 我们称它是 Out-of-Profile 的。

6.14.1.2 MPLS QoS 实现

对于 L3 交换机软件 QoS 的实现, 首先应该给出一个参考模型, 而这个模型要求是通用的, 成熟的。QoS 并不能产生新的带宽, 但是它可以将现有的带宽资源做一个最佳的调整 and 配置, 完整的应用 QoS, 可以对网络的数据传输做到完全的控管。

基于差别化服务的 MPLS QoS 在网络的入口指定每个数据包的优先级别, 这些分类信息被携带在标签中的 exp 字段。MPLS QoS 对于相同优先级别的数据包提供相同服务, 而对不同级别的数据包则提供不同的操作。支持 MPLS QoS 的交换机或路由器可以根据数据包的分类信息, 为其提供不同的带宽资源, 并且可以根据配置的监管策略来重写这些数据包携带的分类信息甚至在带宽紧张的时候丢弃某些低级别的数据包。

6.14.2 MPLS QoS 配置

MPLS QoS 配置任务序列如下:

1. 配置分类表(classmap)

建立一个分类规则, 可以按照 exp。之后, 对于不同类别的数据流采取不同的策略。

2. 将 MPLS QoS 应用到端口

配置端口的信任模式为 exp, 或者绑定策略。策略只有绑定到具体的端口, 才在此端口生效。

3. 配置 MPLS QoS 映射关系

配置 exp 到内部优先级, 丢弃优先级和内部优先级到 exp 的映射关系。

4. 显示 MPLS QoS 映射关系

1. 配置分类表(classmap)按照 e x p 分类

命令	解释
全局配置模式	
match exp <exp-list> no match exp	设置分类表中的匹配标准；本命令的 no 操作为删除指定的匹配标准。

2. 配置 trust exp

命令	解释
端口配置模式	
mls qos trust exp no mls qos trust	配置交换机端口为信任 exp 状态；本命令的 no 操作为禁止交换机端口的信任状态。本交换机不支持此命令。

3. 配置 MPLS QoS 映射关系

命令	解释
全局配置模式	
mls qos map {exp-intp <intp1..intp8> exp-dp <dp1..dp8>} no mls qos map {exp-intp exp-dp} mls qos map intp-exp <exp1..exp8> no mls qos map intp-exp	设置 exp 到内部优先级，exp 到丢弃优先级和内部优先级到 exp 的映射。

4. 显示 MPLS QoS 映射关系

命令	解释
特权配置模式	
show mls qos maps [exp-intp exp-dp[intp-exp]]	显示 MPLS QoS 的映射配置信息。

6.14.3 MPLS QoS 举例

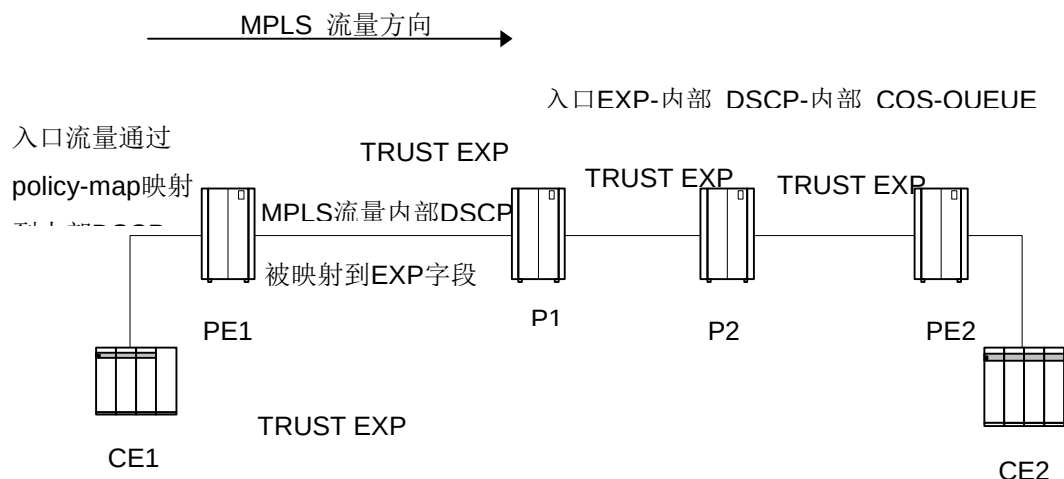


图 6-47 MPLS QoS 举例

按照 diff-serv QoS 模型，边缘交换机进行流量分类，中间交换机按照分类的结果进行转发。如上图所示，边缘 PE 采用 policy map 进行流量分类，并且将分类结果放入 MPLS 数据报文，后续的 P 和 PE trust EXP 进行流量转发。

配置举例

假设普通数据流量通过 vlan10 进入 PE1，voip 流量通过 vlan100 进入 PE1。流量从 Ethernet1/1 进入，从 Ethernet1/2 发出。

PE1: 分类

```
Switch#config
Switch(config)#class-map voip
Switch(Config-ClassMap-voip)#match vlan 100
Switch(Config-ClassMap-voip)# exit
Switch(config)#class-map data
Switch(Config-ClassMap-data)#match vlan 10
Switch(Config-ClassMap-data)# exit
Switch(config)#policy-map p1
Switch(Config-Policy Map-p1)#class voip
Switch(Config-Policy Map-p1-Class-c1)#set Internal-Priority 1
Switch(Config-Policy Map-p1-Class-c1)#exit
Switch(Config-Policy Map-p1)#class data
Switch(Config-Policy Map-p1-Class-c1)#set Internal-Priority 0
Switch(Config-Policy Map-p1-Class-c1)#exit
Switch(Config-Policy Map-p1)#exit
Switch(config)#interface ethernet 1/1
Switch(Config-If-Ethernet1/1)#service-policy input p1
Switch(config)#interface ethernet 1/2
```

```
Switch(Config-If-Ethernet1/2)# mls qos queue wrr weight 1 2 3 4 5 6 7 8
```

data流量的内部优先级为0，根据缺省Int-Prio-TO-QUEUE，从queue 0出去，同时根据缺省的Int-Prio-TO-(EXP, DSCP, COS)映射，数据报文的EXP、DSCP和COS字段被设置为0。

voip流量的内部优先级为1，根据缺省Int-Prio-TO-QUEUE，从queue 1出去，同时根据缺省的Int-Prio-TO-(EXP, DSCP, COS)映射，数据报文的EXP、DSCP和COS字段被设置为1。

P1、P2、PE2：按照分类结果转发，流量都从 Ethernet1/1 进入，从 Ethernet1/2 发出。

```
Switch#config
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch Config-If-Ethernet1/1)#
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(Config-If-Ethernet1/2)# mls qos queue wrr weight 1 2 3 4 5 6 7 8
```

data 流量的 EXP 为 0，根据缺省 EXP-TO-(Int-Prio, Drop-Prec)和 Int-Prio-TO-QUEUE 设置，从 queue 0 出去，同时根据缺省的 Int-Prio-TO-(EXP, DSCP, COS)映射，数据报文的 EXP、DSCP 和 COS 字段被设置为 0。

voip 流量的 EXP 为 1，根据缺省 EXP-TO-(Int-Prio, Drop-Prec)和 Int-Prio-TO-QUEUE 设置，从 queue 0 出去，同时根据缺省的 Int-Prio-TO-(EXP, DSCP, COS)映射，数据报文的 EXP、DSCP 和 COS 字段被设置为 1。

6.14.4 MPLS QoS 排错帮助

- ☞ 交换机端口必须使能 MPLS 功能，如果没使能，则 MPLS QoS 功能无效。
- ☞ MPLS 报文通过使能了 MPLS QoS 的端口后，报文的 cos 值被修改为 0，dscp 值不变。

6.15 基于流的重定向

6.15.1 基于流的重定向介绍

基于流的重定向功能是指交换机把端口接收到的、符合特定条件（ACL 指定）的数据帧转发到另一个指定的端口；其中符合同一个特定的条件称为一类流，数据帧的入端口称为重定向源端口，指定的出端口称为重定向目的端口。通常基于流的重定向应用在两个方面：1.在重定向目的端口处连接一个协议分析仪（如 Sniffer）或者 RMON 监测仪，可以监视和管理网络，并且能诊断网络故障；2.为特定类型的数据帧指定转发策略。

交换机只能为同一个重定向源端口内的同一类流指定一个唯一的重定向目的端口，为同一个重定向源端口内的不同流指定一个不同的重定向目的端口；同一类流也可以应用到不同的源端口。

6.15.2 基于流的重定向配置任务序列

1. 基于流的重定向配置
2. 查看当前的基于流的重定向配置

1.基于流的重定向配置

命令	解释
物理接口配置模式	
access-group <aclname> redirect to interface [ethernet <IFNAME> <IFNAME>] no access-group <aclname> redirect	为端口指定基于流的重定向； 本命令的 no 操作为删除基于流的重定向

2.查看当前的基于流的重定向配置

命令	解释
全局配置模式/特权用户配置模式	
show flow-based-redirect {interface [ethernet <IFNAME> <IFNAME>]}	显示当前系统/端口中配置的基于流的重定向信息

6.15.3 基于流的重定向举例

案例：

用户有如下的配置需求：将 1 端口收到的源 IP 为 192.168.1.111 的帧重定向到 6 端口，即从 1 端口收到的源 IP 为 192.168.1.111 的帧通过 6 端口转发出去。

配置修改：

- 1: 配置一条 ACL，匹配条件：源 IP 为 192.168.1.111；
- 2: 将基于该流的重定向应用到端口 1。

配置步骤如下：

```
Switch(config)#access-list 1 permit host 192.168.1.111
```

```
Switch(config)#interface ethernet 1/1
```

```
Switch(Config-If-Ethernet1/1)# access-group 1 redirect to interface ethernet 1/6
```

6.15.4 基于流的重定向排错帮助

当配置基于流的重定向功能出现问题时，请检查是否是如下原因：

- ☞ 流的类型（ACL）只能为数字标准 IP 访问列表、数字扩展 IP 访问列表、命名标准 IP 访问列表、命名扩展 IP 访问列表、数字标准 IPv6 访问列表和命名标准 IPv6 访问列表；
- ☞ ACL 中不能配置 **Timerange** 和 **Portrange** 参数，ACL 类型必须为 Permit。
- ☞ 基于流的重定向功能，其重定向目的端口必须为千兆口。

6.16 Egress QoS

6.16.1 Egress QoS 介绍

在传统的 IP 网络中，所有的报文都被无区别的等同对待，每个网络设备对所有的报文均采用先入先出的策略进行处理，它尽最大的努力（Best-Effort）将报文送到目的地，但对报文传送的可靠性、传送延迟等性能不提供任何保证。网络发展日新月异，随着 IP 网络上新应用的不断出现，对 IP 网络的服务质量也提出了新的要求。例如 VoIP 和视频等时延敏感业务对报文的传输时延提出了较高要求。如果报文传送延时太长，将是用户所不能接受的（相对而言，E-Mail 和 FTP 业务对时间延迟并不敏感）。为了支持具有不同服务需求的语音、视频以及数据等业务，要求网络能够区分出不同的通信，进而为之提供相应的服务。传统 IP 网络的尽力服务不可能识别和区分出网络中的各种通信类别，而具备通信类别的区分能力正是为不同的通信提供差异化服务的前提，所以说传统网络的尽力服务模式已不能满足应用的需要。QoS 技术的出现便致力于解决这个问题。

Egress Policymap 即出口 QoS 策略，Egress PolicyMap 可以在出口方向对报文进行 QoS 控制，通过利用各种各样的技术为选定的网络通信提供更好的服务的能力。Egress PolicyMap 包括 class-map、policy-map 两大部分，class-map 用于选定操作的数据包，policy-map 用于设定使用何种操作。目前仅部分设备支持 Egress QoS。

6.16.1.1 Egress QoS 术语

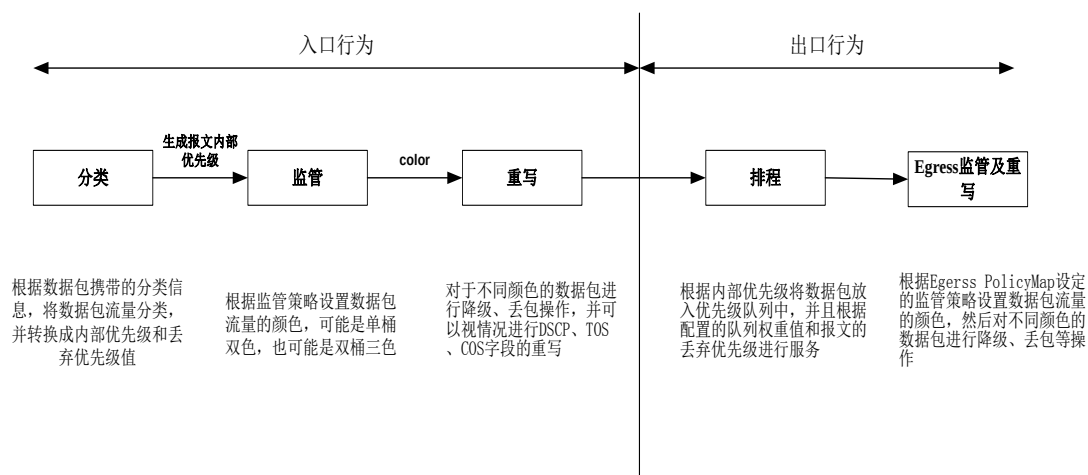
Egress QoS: 在端口出方向实现 QoS 功能。

Inner_vid: 双 TAG 的时候，靠近网络层头的 TAG 所带的 vlan id。

Outer_vid: 双 TAG 的时候，靠近网络链路层包头的 TAG 所带的 vlan id。一个 TAG 就默认为 outer tag。

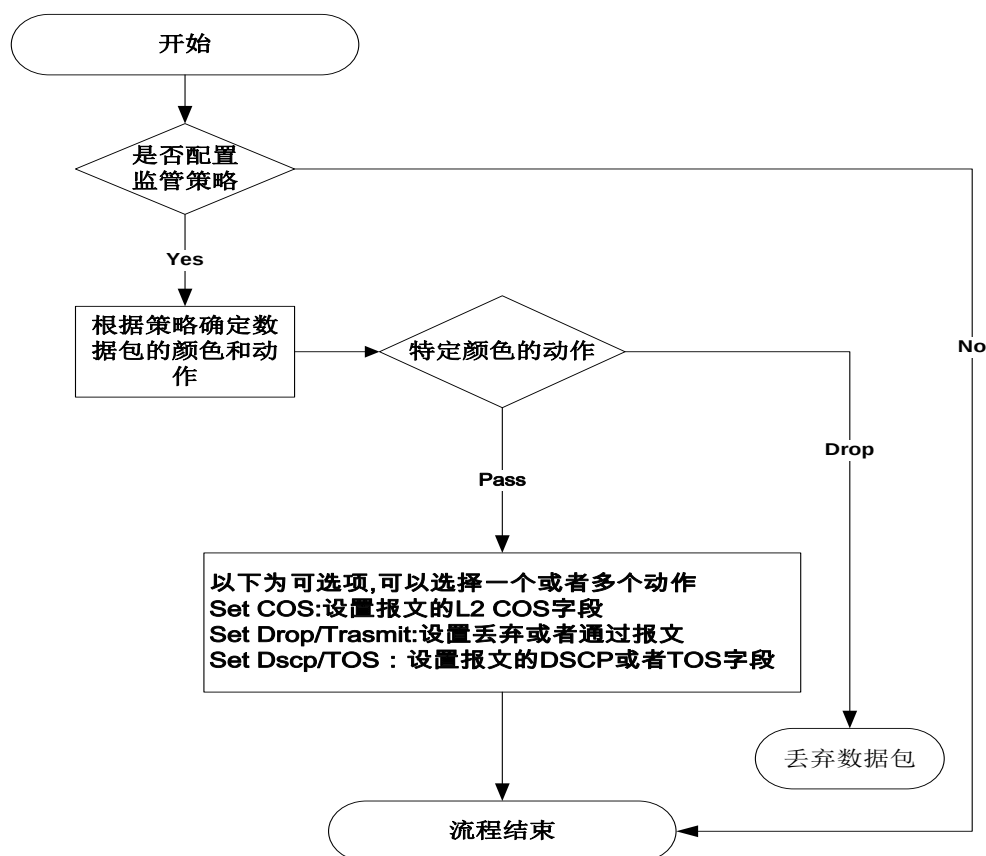
Outer_tpid: 网络链路层包头的协议类型，表示 outer tag 的类型。

6.16.1.2 Egress QoS 基本模型



Egress 监管及重写根据上游报文的特点（包括：CoS、DSCP 等域值），在报文出口之前，对报文进行最后一次 QoS 修改。

监管行为可以基于流配置不同的监管策略，为分类后的流量分配带宽，带宽的分配策略可以是双桶双色，也可以是双桶三色。流量会被分配不同的颜色（color），对不同颜色的流量可以选择丢包、通过两种处理方式；对于通过处理方式的数据包，可以附加重写动作。下面的流程图详细说明 Egress QoS 流程：



6.16.2 Egress QoS 配置

Egress QoS 配置任务序列如下：

配置分类表（classmap）

建立一个分类规则，可以按照 ACL、CoS、VLAN ID、IPv4 Precedent、DSCP、IPv6 DSCP 来分类。之后，对于不同类别的数据流采取不同的策略。

配置策略表（policymap）

对数据流进行分类之后，就可以建立一个策略表，之后就可以将其对应一个先前建立的 class-map，进入策略分类表模式，然后就可以对于不同的数据流采取不同的策略，如带宽限制、分配新的 DSCP 值等等。

将 Egress QoS 应用到端口或 VLAN

配置端口的信任模式，或者绑定策略。策略只有绑定到具体的端口，才在此端口生效。策略也可以绑定到具体的 VLAN 上。

1. 配置分类表（class-map）

命令	解释
全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立一个 class-map（分类表），并进入 class-map 模式；本命令的 no 操作为删除指定的 class-map。
match {access-group <acl-index-or-name> ip dscp <dscp-list> ip precedence <ip-precedence-list> ipv6 dscp <dscp-list> vlan <vlan-list> cos <cos-list> ipv6 access-group <acl-index-or-name>} no match {access-group ip dscp ip precedence ipv6 dscp vlan cos ipv6 access-group}	设置分类表中的匹配标准，可以根据 ACL、CoS、VLAN ID、IPv4 Precedence、DSCP、IPv6 DSCP 优先级等对数据流进行分类；本命令的 no 操作为删除指定的匹配标准。

2. 配置策略表（policy-map）

命令	解释
全局配置模式	
policy-map <policy-map-name> no policy-map <policy-map-name>	建立一个 policy-map（策略表），并进入 policy-map（策略表）模式；本命令的 no 操作为删除指定的 policy-map。
class <class-map-name> [insert-before <class-map-name>]	建立一个 policy-map（策略表）之后，可以将其对应于一个 class（策略分类

no class <class-map-name>	表), 并进入策略分类表模式, 之后就可以对不同的数据流采取不同的策略或者分配一个新的 DSCP 值; 本命令的 no 操作为删除指定策略分类表。
set {ip dscp <new-dscp> ip precedence <new-precedence> cos <new-cos> c-vid <new-c-vid> s-vid <new-s-vid> s-tpid <new-s-tpid>} no set {ip dscp ip precedence cos c-vid s-vid s-tpid}	为分类后的流量分配一个新的 DSCP、CoS 和 IP Precedence 值等; 本命令的 no 操作为取消分配新的值。
单桶模式: policy <bits_per_second> <normal_burst_bytes> ({conform-action ACTION} exceed-action ACTION}) 双桶模式: policy <bits_per_second> <normal_burst_bytes> [pir <peak_rate_bps>] <maximum_burst_bytes> [{conform-action ACTION exceed-action ACTION violate-action ACTION}] ACTION 定义: drop transmit set-dscp-transmit <dscp_value> set-cos-transmit <cos_value> no policy	为分类后的流量配置一个策略, 支持三色 color 的非聚合 Policy 命令, 根据配置参数解析出令牌桶的工作模式, 是单速单桶、单速双桶还是双速双桶, 并且给不同颜色报文设置相应动作。no 命令删除模式配置。单桶模式只有特定的交换机支持。
accounting no accounting	为分类后的流量设置统计功能。在策略分类表配置模式下将功能开启后, 会对策略分类表中的流量增加统计功能, 单桶模式, 报文经过 policy 时仅会有绿色和红色两种颜色, 打印统计信息时 in-profile 表示绿色, out-profile 表示红色; 双桶模式下, 会有三种颜色的报文, in-profile 表示绿色, out-profile 表示红色和黄色。

3. 策略应用到端口或 vlan 出方向

命令	解释
接口配置模式	
service-policy output <policy-map-name> no service-policy output {<policy-map-name>}	在端口出方向上应用一个策略表；本命令的 no 操作为删除应用到交换机端口的某个指定策略表或删除该端口出方向应用的所有策略表。
全局配置模式	
service-policy output <policy-map-name> vlan <vlan-list> no service-policy output {<policy-map-name>} vlan <vlan-list>	在交换机 VLAN 出方向上应用一个策略表；本命令的 no 操作为删除应用到交换机 VLAN 接口的某个指定策略表或删除该 vlan 接口出方向应用的所有策略表。

4. 清除特定端口或者 VLAN 的 Accounting 数据

命令	解释
特权配置模式	
clear mls qos statistics [interface <interface-name> vlan <vlan-id>]	清除特定端口或者 vlan Policy Map 的 Accounting 数据，如果不带参数，清除所有的 Policy Map 的 Accounting 数据。

5. 显示 QoS 的配置信息

命令	解释
特权配置模式	
show mls qos {interface [<interface-id>] [policy queuing] vlan <vlan-id>}	显示 QoS 在交换机端口上的配置信息。
show class-map [<class-map-name>]	显示 QoS 的分类表信息。
show policy-map [<policy-map-name>]	显示 QoS 的策略表信息。

6.16.3 Egress QoS 举例

案例 1:

在端口 1 的出口方向,对于 dscp 值为 0 的报文,选取的动作为将该报文 cos 值改为 4。

创建一个分类表

```
Switch(config)#class-map c1
```

```
switch(config-classmap-c1)#match ip dscp 0
```

```
switch(config-classmap-c1)#exit
创建一个策略表
switch(config)#policy-map p1
switch(config-policy-map-p1)#class c1
switch(config-policy-map-p1-class-c1)#set cos 4
switch(config-policy-map-p1-class-c1)#exit
switch(config-policy-map-p1)#exit
将策略绑定在端口
switch(config)#in e 1/1
switch(config-if-ethernet1/1)#service-policy output p1
```

案例 2:

在 vlan 10 的出方向，对于 ipv6 dscp 值为 7 的报文，选取的动作为将该报文 cos 值修改为 4。

```
创建一个分类表
switch(config)#class-map c1
switch(config-classmap-c1)#match ipv6 dscp 7
switch(config-classmap-c1)#exit
创建一个策略表
switch(config)#policy-map p1
switch(config-policy-map-p1)#class c1
switch(config-policy-map-p1-class-c1)#set cos 4
switch(config-policy-map-p1-class-c1)#exit
switch(config-policy-map-p1)#exit
将策略绑定在 vlan
switch(config)#service-policy output p1 vlan 10
```

6.16.4 Egress QoS 排错帮助

- ☞ 目前仅部分设备支持 Egress QoS，请确认当前设备支持该功能
- ☞ 如果配置的 policy 无法绑定到端口或 VLAN，请确认分类表中 match 的选项当前设备是否支持
- ☞ 如果终端打印提示资源不足，请确认有足够的资源下发当前配置的 policy
- ☞ 如果配置了 match acl 的 policy 无法绑定到端口或 VLAN，请确认 ACL 中存在规则并且包含有 permit 规则

6.17 灵活 QinQ 操作

6.17.1 灵活 QinQ 功能介绍

6.17.1.1 QinQ 技术

Dot1q-tunnel 也称 QinQ (802.1Q-in-802.1Q)，是对 802.1Q 的扩展，其核心思想是将用户私网 VLAN 标签（CVLAN tag）封装到公网 VLAN 标签（SPVLAN tag）上，报文带着两层 VLAN 标签穿越服务商的骨干网络，从而为用户提供一种较为简单的二层隧道。它简单而易于管理，仅仅通过静态配置即可实现，特别适用于小型的以三层交换机为骨干的企业网或小规模城域网。

QinQ 分为两种，基本 QinQ 和灵活 QinQ，两者优先级关系为：灵活 QinQ 优先级大于基本 QinQ。

6.17.1.2 基本 QinQ

基本 QinQ 是基于端口的：当端口上配置了 QinQ 后，不论从该端口收到的报文是否带 tag，设备都会为该报文打上本端口缺省的 VLAN tag。基本 QinQ 使用起来比较简单，但设置 VLAN tag 的方式欠缺灵活。

6.17.1.3 灵活 QinQ

灵活 QinQ 是基于数据流的：通过匹配具体的流来选择是否打上外层 VLAN Tag，打上何种外层 VLAN Tag：如根据用户 VLAN tag，MAC 地址，IPv4/IPv6 地址，IPv4/IPv6 协议，应用程序端口号等实施灵活 QinQ 特性。这样可以根据不同用户，不同业务等对报文进行外层 VLAN Tag 封装，对多种业务实施不同的承载方案。

6.17.2 灵活 QinQ 配置

灵活 QinQ 配置任务序列

灵活 QinQ 数据流的匹配采用 QOS 的 policy map 规则下发，配置序列如下：

- 1、创建分类表，对不同的数据流进行分类
- 2、创建灵活 QinQ 策略表，与分类表进行关联，并设置相应的操作
- 3、绑定灵活 QinQ 策略表到端口

1. 配置分类表（classmap）

命令	解释
全局配置模式	
class-map <class-map-name>	建立一个 class-map（分类表），并进入

no class-map <class-map-name>	class-map 模式; 本命令的 no 操作为删除指定的 class-map。
match {access-group <acl-index-or-name> ip dscp <dscp-list> ip precedence <ip-precedence-list> ipv6 access-group <acl-index-or-name> ipv6 dscp <dscp-list> vlan <vlan-list> c-vlan <vlan-list> cos <cos-list>} no match {access-group ip dscp ip precedence ipv6 access-group ipv6 dscp vlan c-vlan cos}	设置分类表中的匹配标准, 可以根据 ACL、CoS、VLAN ID、IPV4 Precedent、DSCP 等对数据流进行分类; 本命令的 no 操作为删除指定的匹配标准。

2. 配置灵活 QinQ 策略表 (policymap)

命令	解释
全局配置模式	
policy-map <policy-map-name> no policy-map <policy-map-name>	建立一个 policy-map (策略表), 并进入 policy-map (策略表) 模式; 本命令的 no 操作为删除指定的 policy-map。
class <class-map-name> [insert-before <class-map-name>] no class <class-map-name>	建立一个 policy-map (策略表) 之后, 可以将其对应于一个 class (策略分类表), 并进入策略分类表模式, 之后就可以对不同的数据流采取不同的策略或者分配一个新的 DSCP 值; 本命令的 no 操作为删除指定策略分类表。
set {internal priority <new-inp> cos <new-cos> s-vid <new-vid> c-vid <new-vid>} no set {internal priority cos s-vid c-vid}	为匹配分类表的数据包分配新的 Cos 值和 Vid 等操作; no 命令为取消分配新的值。
add { s-vid <new-vid> c-vid <new-vid> } no add {s-vid c-vid}	对匹配分类表的数据包增加指定的外层 tag 或者内层 tag, no 命令为取消操作。
delete {c-vid} no delete c-vid	对匹配分类表的数据包删除内层 VLAN Tag, no 命令为取消操作。

3. 绑定灵活 QinQ 策略表到端口

命令	解释

端口配置模式	
service-policy <policy-map-name> in no service-policy <policy-map-name> in	在端口上应用策略表；本命令的 no 操作作为删除应用到交换机端口的某个指定策略表。

4. 显示端口绑定的灵活 QinQ 策略表

命令	解释
特权配置模式	
show mls qos {interface [<interface-id>]}	显示灵活 QinQ 在交换机端口上的配置信息。

6.17.3 灵活 QinQ 举例

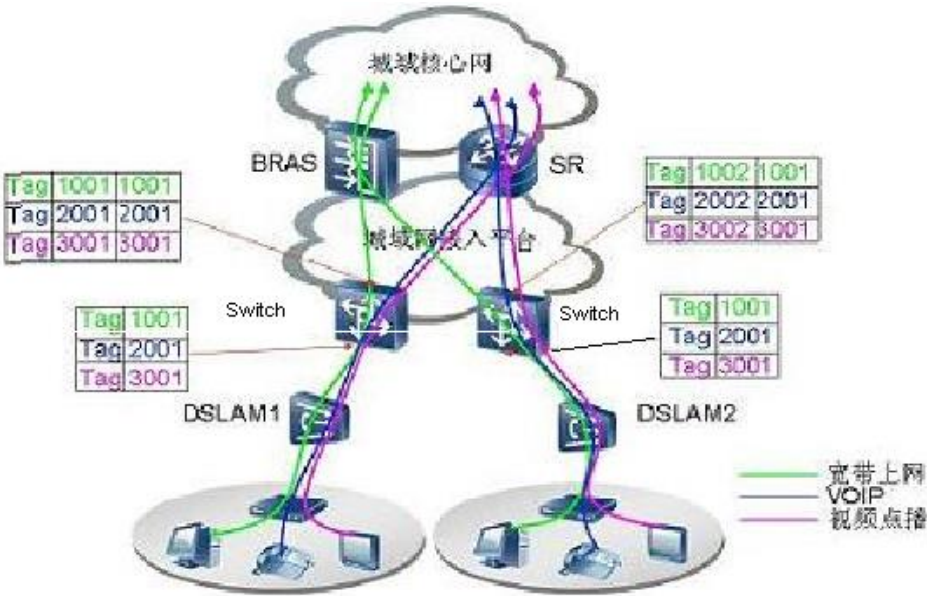


图 6-48 灵活 QinQ 应用拓扑

如上所示：比如 DSLAM1 下的用户甲，分配三个 VLAN，VLAN 标签值分别为 1001，2001，3001，VLAN1001 对应宽带上网，VLAN2001 对应 VOIP 业务，VLAN3001 对应 VOD 业务，在业务进入 Switch 交换机的下行端口后，下行端口启用灵活 QINQ 功能，根据用户 VLAN ID 分别打上不同的外层标签，如果用户数据外层标签为 1001，直接在外层增加一层标签 1001（该标签在公网唯一），业务进入 Switch 上的宽带上网 VLAN1001，数据分流到 BRAS 设备，如果标签为 2001，根据流规则，外层增加 VLAN TAG2001，业务分流到 SR 设备，同理，标签为 3001 的数据会增加一层外层标签 3001，然后分流到 SR。同理，在 DSLAM2 下的乙用户，其不同的业务可以分配不同的 VLAN 标签，注意，这里分配的 VLAN 标签可以和甲用户分配的标签相同，在进入 Switch 后根据内层标签增加一个外层标签，注意，外层标签和甲用户分配的不同，主要是为了区分 DSLAM 的位置，也是为了最终定位用

户。

配置步骤如下：

假如 DSLAM1 流量业务进入 Switch 交换机下行端口为端口 1，此 Switch 配置如下：

```
Switch(config)#class-map c1
Switch(config-classmap-c1)#match vlan 1001
Switch(config-classmap-c1)#exit
Switch(config)#class-map c2
Switch(config-classmap-c2)#match vlan 2001
Switch(config-classmap-c2)#exit
Switch(config)#class-map c3
Switch(config-classmap-c3)#match vlan 3001
Switch(config-classmap-c3)#exit
Switch(config)#policy-map p1
Switch(config-policymap-p1)#class c1
Switch(config-policymap-p1-class-c1)#add s-vid 1001
Switch(config-policymap-p1)#class c2
Switch(config-policymap-p1-class-c2)#add s-vid 2001
Switch(config-policymap-p1)#class c3
Switch(config-policymap-p1-class-c3)#add s-vid 3001
Switch(config-policymap-p1-class-c3)#exit
Switch(config-policymap-p1)#exit
Switch(config)#interface ethernet 1/1
Switch(config-if-ethernet1/1)#service-policy p1 in
```

DSLAM2 流量业务进入 Switch 交换机下行端口为端口 1，此 Switch 配置如下：

```
Switch(config)#class-map c1
Switch(config-classmap-c1)#match vlan 1001
Switch(config-classmap-c1)#exit
Switch(config)#class-map c2
Switch(config-classmap-c2)#match vlan 2001
Switch(config-classmap-c2)#exit
Switch(config)#class-map c3
Switch(config-classmap-c3)#match vlan 3001
Switch(config-classmap-c3)#exit
Switch(config)#policy-map p1
Switch(config-policymap-p1)#class c1
Switch(config-policymap-p1-class-c1)#add s-vid 1002
Switch(config-policymap-p1)#class c2
```

```
Switch(config-policymap-p1-class-c2)#add s-vid 2002
Switch(config-policymap-p1)#class c3
Switch(config-policymap-p1-class-c3)#add s-vid 3002
Switch(config-policymap-p1-class-c3)#exit
Switch(config-policymap-p1)#exit
Switch(config)#interface ethernet 1/1
Switch(config-if-ethernet1/1)#service-policy p1 in
```

6.17.4 灵活 QinQ 排错帮助

- ☞ 如果配置的灵活 QinQ 策略无法绑定到端口，请确认灵活 QinQ 支持所配置的分类表和策略表动作
- ☞ 如果配置的灵活 QinQ 策略无法绑定到端口，如果分类表匹配的是 ACL 规则，请确认 ACL 中包含有 permit 规则
- ☞ 如果配置的灵活 QinQ 策略无法绑定到端口，请确认交换机中存在足够的 VFP 资源下发绑定

6.18 Vlan-shaping

6.18.1 Vlan-shaping 介绍

VLAN Shaping 是指对特定的 VLAN 实现流量整形的差异化服务。在 Vlan Shaping 中，数据包中的 vlan id 相当于差异化服务编码。这与传统的流量整形是一个核心差别。传统的流量整形是基于端口和队列的，实现差异服务的基础主要是数据包的 cos/dscp/exp 等字段。因此传统的 QoS 服务更多的是基于业务，在当前多用户、多业务共享数据链路的应用中无法为不同的用户提供独立的 QoS 服务。

6.18.1.1 Vlan-shaping 术语

QoS: Quality of Service 服务品质保证，提供稳定、可预测的数据传送服务，来满足使用程序的要求，QoS 不能产生新的带宽，而是根据使用程序的需求以及网络管理的设置来有效的管理网络带宽。

VID: VLAN ID,位于 Vlan Tag 中，用于识别 vlan 的 12bit 值，范围为 1~4094。

Queue: Cos 出口调度队列。

EUC Queue: Extended Unicast Queue 扩展单播队列，用于 Vlan 流量调度的队列。因此在本文中，euc queue 和 vlan shaping queue 是同一个概念。

Forwarding-class: 转发类，用于标识一类流量，在多级调度中指原始输入队列。如果把

交换机的多级调度结构看作一棵树的话，S1 节点相当于根节点，S2 节点相当于分支节点，原始输入队列（单播队列，组播队列，扩展单播队列）相当于叶节点，即转发类。目前，此转发类特指扩展单播队列（即 vlan shaping 队列）

Forwarding-class_match:转发类的匹配规则，用于流量分类，符合分类规则的流量将进入该转发类所对应的原始队列中参与出口队列调度。目前，此转发类只能匹配 outer tag 中的 vlan id。

Forwarding-profile: 转发策略，用于设置对某类流量设置的与队列调度相关的动作，比如队列调度算法、队列调度权重、最小最大带宽等。目前，此转发策略只能设置最小最大带宽和引用某个丢弃策略。一个转发策略可以被一个转发类引用，表示对该转发类所标识的流量设置的队列调度动作。

Drop-profile:丢弃策略，用于设置对某类流量设置的与队列丢弃相关的动作，比如设置 wred 参数。一个丢弃策略可以被一个转发策略引用，表示按照丢弃策略设置的 wred 算法进行队列丢弃。

6.18.1.2 Vlan-shaping 实现

Vlan 流量出口调度整形原理图如下：

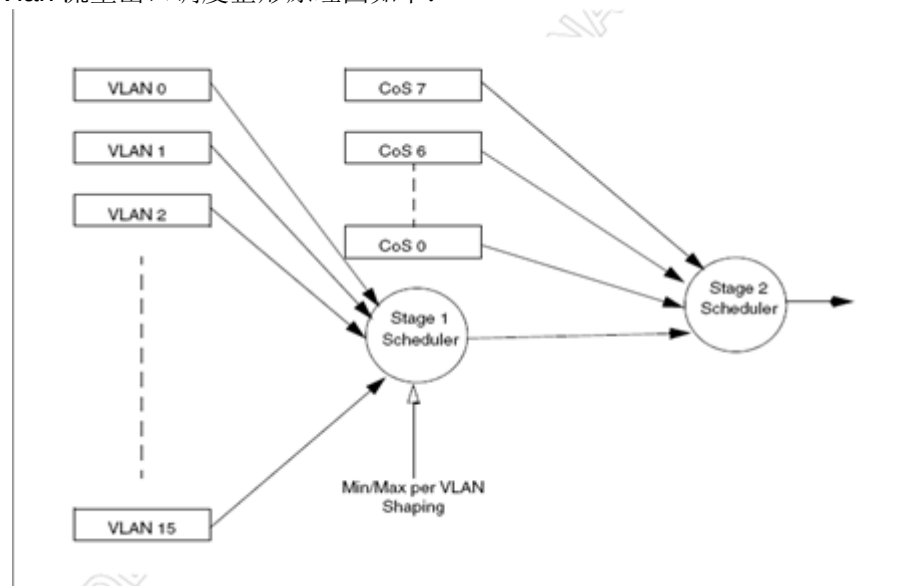


图 6-49 vlan-shaping 实现示意图

在上图中，交换机提供了 16 个 Vlan Shaping，vlan shaping 队列通过调度节点“Stage1 Scheduler”进行调度。“Stage1 Scheduler”的调度结果又作为“Stage2 Scheduler”的第 9 个队列与其余的 8 个普通 cos 队列参与竞争，最终的胜出者才会从出端口出去。因此从本质上来说，Vlan Shaping 相当于一个二级调度机制。

在使能了 vlan shaping 的场合，数据包根据 tag 中的 vlan id 值被放入指定的 vlan shaping 队列参与调度，此时数据包原始的 cos/dscp/exp 等字段无法决定队列号。在没有使能 vlan shaping 的场合，数据包根据端口 trust 类型(trust cos/trust dscp/trust exp)来决定数据包被放入的队列号。

6.18.1.3 QoS 基本模型

VLAN Shaping 模型包括 VLAN、CE、PE、PCN。其基本概念如下：

VLAN: VLAN Shaping 中所说的 VLAN 指的是 OUTER VLAN。运营商通过 Outer Vlan Id 来标识不同的用户或者业务。

CE(Custom Edge): 用户侧边缘设备。其一端提供了链接到运营商网络的上行端口，另一端提供了链接到用户网络的下行端口。

PE(Provider Edge): 运营商边缘设备。其上行口链接到运营商核心网络，下行口通过链接 CE 的上行口提供了到用户网络的通路。

PCN(Provideer Core Network): 运营商核心网络

Vlan shaping 的一个典型应用是运营商实施城域网部署时，在 PE 设备上行口通过实施 vlan shaping 对来自不同用户的不同 vlan 流量进行指定的 QoS 服务。

在下图中，网络中存在着语音、数据、视频等三种不同的业务，这三种业务对网络的延迟性和带宽要求是不一样的。语音业务虽然对带宽要求不高，但对网络延迟非常敏感。视频业务不仅对网络延迟比较敏感，对带宽要求也比较高。普通数据通信对带宽和延迟则介于另外两类业务之间。传统的 QoS 服务能够很好的为不同的业务施行差异化服务。

然而，在实际应用中，运营商提供给用户的链接通路往往是有限的。多个用户会共享一条上行链接通路。在下图中，存在两类不同的用户，分别位于 VLAN2-199，VLAN200-300 中。这两类客户流量通过汇聚设备到达用户侧边缘设备 CE 后共享一条链接通路到运营商边缘网络设备 PE 上。

对运营商而言，在共享链路上对不同的用户实施不同的 QoS 服务能最大程度保护和利用现有的硬件资源。运营商不必为不同的客户提供不同的链接通路，在每一个链接通路上实施与相应客户对应的 QoS 服务。在这条共享链路上，唯一要做的就是能够区分出流量到底来自哪个客户。Vlan id 正好能够标识出这类用户，基于 vlan id 在 PE 设备的上行口实施 QoS。

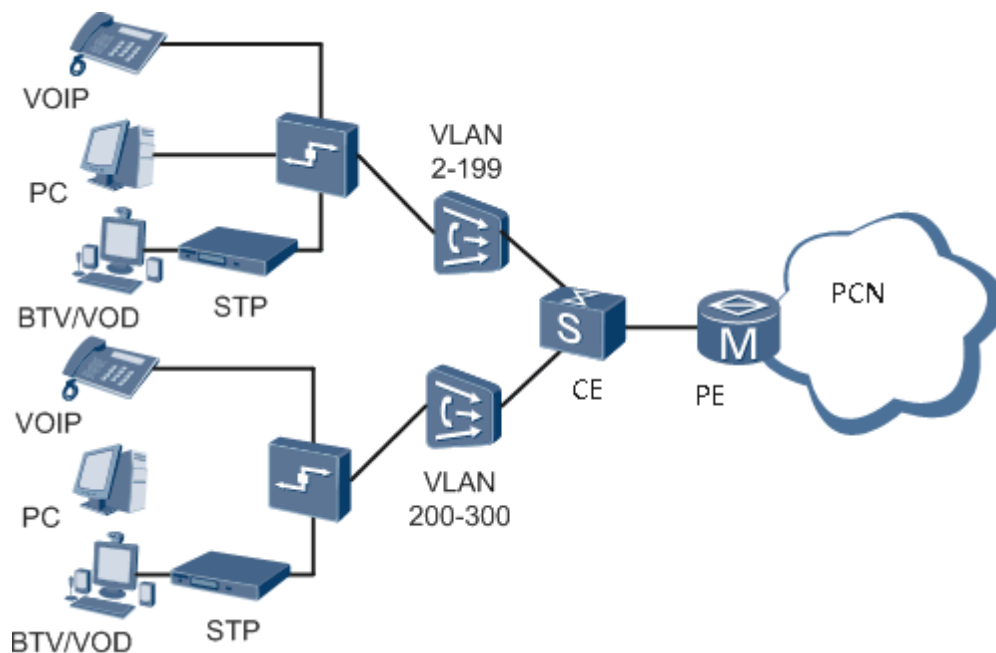


图 6-50 Vlan shaping 在多业务、多用户场合的典型应用

6.18.2 Vlan-shaping 配置

Vlan-shaping 配置任务序列如下：

配置转发类（forwarding-class）

建立一个转发类，可以匹配不同的 vlan。之后，对于不同转发类绑定到不同调度策略中，对该转发类里面的数据流采取不同的策略。该配置项是必须的。

配置丢弃策略(drop-profile)

建立一个丢弃策略，用于设置队列丢弃算法 WRED 的各项参数。该配置项不是必须的，在没有开启 wred 算法时或者使用默认的 wred 参数时，可以不用配置该配置项。

配置转发策略（forwarding-policy）

建立一个转发策略，可以设置带宽限制和引用丢弃策略。之后，可以和转发类一起绑定到调度策略里，以对转发类里面的数据流进行调度控制。该配置项不是必须的，如果不配置该配置项，则没有带宽限制和使用默认的 wred 参数。

配置调度策略（schedule policy）

建立一个调度策略，然后把转发类和转发策略绑定到 level 2 的第 1 个节点的各个队列上，接着配置各个调度节点的队列调度算法、队列权重。该配置项是必须的。其中引用转发策略不是必须的。如果没有引用转发策略，则没有带宽限制和使用默认的 wred 参数。

在端口上开启 ets 功能(多级调度功能)

只有在端口上开启了 ets 功能，才能在端口上绑定自定义的调度策略。该配置项是必须的。

将 vlan-shaping 应用到端口

配置端口的绑定策略。策略只有绑定到具体的端口，才在此端口生效。该配置项是必须的。

1. 配置转发类（forwarding-class）

命令	解释
全局配置模式	
mls QoS forwarding-class <fc-name> [id <fc-id>] no mls QoS forwarding-class <fc-name>	建立一个 forwarding-class（转发类），并进入 forwarding-class 模式；本命令的 no 操作为删除指定的 forwarding-class。
转发类配置模式	
match s-vid<svid-list> no match s-vid[svid-list]	设置转发类中的匹配标准，可以根据 vlan-id 对数据流进行分类；本命令的 no 操作为删除指定的匹配标准。

2. 配置转发策略（forwarding-profile）

命令	解释
全局配置模式	
mls QoS forwarding-profile <fp-name> [id <fp-id>] no mls QoS forwarding-profile <fp-name>	建立一个 forwarding-profile（转发策略），并进入 forwarding-profile（转发策略）模式；本命令的 no 操作为删除指定的 forwarding-profile。
转发策略配置模式	
Bandwidth <minBandwidth> <maxBandwidth> no bandwidth	建立一个 forwarding-profile(转发策略)之后，可以对其设置一个带宽限制，用于限制引用它的数据流的带宽；本命令的 no 操作为删除带宽限制。
drop <dp-name> no drop	引用一个丢弃策略，对绑定了该转发策略的数据流进行该丢弃策略；本命令的 no 操作为取消引用。

3. 配置调度策略（schedule policy）

命令	解释
全局配置模式	
mls QoS schedule policy <fp-name>	建立一个 schedule policy（调度策略），

no mls QoS schedule policy <fp-name>	并进入 schedule policy（调度策略）模式；本命令的 no 操作为删除指定的调度策略（schedule policy）。
调度策略配置模式	
mls QoS schedule level <level-id> node <node-id> no mls QoS schedule level <level-id> node <node-id>	进入该调度节点进行设置；本命令的 no 操作为删除指定的节点配置。
调度节点配置模式	
mls QoS schedule algorithm {sp wrr wdrr} no mls QoS schedule algorithm	配置队列调度算法。
mls QoS schedule {wrr wdrr} weight <weight1...weight18> no mls QoS schedule {wrr wdrr} weight	配置队列调度权重。
mls QoS bandwidth <minBandwidth> <maxBandwidth> no mls QoS bandwidth	设置所有 Vlan shaping 队列的总体最小带宽限制和最大带宽保证。本命令的 no 操作为删除带宽限制的配置。
mls QoS schedule queue <queueID> input forwarding-class <fc-name>[profile <fp-name>] no mls QoS schedule queue <queueID> input forwarding-class <fc-name>	设置调度节点的某个队列输入来源是转发类，并可以选择性的为转发类指定自定义的转发策略。本命令的 no 操作为删除绑定设置。

4. 丢弃策略

命令	解释
全局配置模式	
mls QoS drop-profile <dp-name> [id <dp-id>] no mls QoS drop-profile <dp-name>	建立一个丢弃策略，并进入丢弃策略模式；本命令的 no 操作为删除指定的丢弃策略。
丢弃策略配置模式	
dp <dp> drop-startpoint <start> drop-endpoint <end> max-drop-rate <rate> no dp <dp>	设置丢弃策略中的 wred 参数，包括颜色类型，最小阈值，最大阈值和最大丢弃概率。本命令的 no 操作为删除对该颜色类型的设置，恢复到默认值。

5. 设置 vlan shaping 队列相对于其他普通 cos 队列的优先级

命令	解释
全局配置模式	
mls QoS euc <id> priority <prio>	全局设置 vlan shaping 队列相对于其他

no mls QoS euc <id> priority	普通 cos 队列的优先级。本命令的 no 操作为删除优先级设置，使之回到默认值。

6. 显示 vlan-shaping 的配置信息

命令	解释
全局配置模式	
show mls QoS vlan shaping capable ports	显示出目前所有能够支持 vlan shaping 功能的端口名称。
show mls QoS forwarding-class [fc-name]	显示出指定的转发策略信息。
show mls QoS drop-profile [dp-name]	显示出指定的丢弃策略信息 show mls QoS drop-profile [dp-name]
show mls QoS schedule interface [<interface-name> <interface-list>]	显示指定端口上所配置的 vlan shaping 信息, 包括 vlan shaping 队列调度算法, 队列权重值等。
show mls QoS vlan shaping priority	显示 vlan shaping 队列相对于其他普通队列的优先级。

6.18.3 Vlan-shaping 举例

案例 1:

新建转发类 fc1,fc2 设置匹配规则是 vlan1,vlan2 和 vlan3,vlan4 流量;新建丢弃策略 dp1, 配置丢弃策略 dp1, 对丢弃优先级为 0 的包,当队列长度达到最大队列长度的 20%时开始丢包, 当队列长度达到最大长度的 40%时, 开始全速丢包, 但丢弃的包占总包的最大比例为 80%; 新建转发策略 fp1, 最小带宽保证为 10kbps,最大带宽限制为 200kbps, 关联丢弃策略 dp1; 新建调度策略 p1,设置 p1 中的 S3.1 节点的第一个队列的输入来源是 fc1, 关联转发策略 fp1, 设置 p1 中的 S3.1 节点的第二个队列的输入来源是 fc2, 设置 vlan-shaping 队列 S3.1 节点最小带宽保证为 64kbps,最大带宽限制为 256kbps, 设置 vlan shaping 队列 1-8 的 wrr 队列权重分别为 1,3,5,2,7,4,9,14.设置队列 9-16 的 wrr 队列权重分别为 8,3,2,14,17,4,7,0。最后把调度策略 p1 绑定到端口 ethernet1/1 上。

案例配置:

```
(config)#mls qos forwarding-class fc1
(config-forwarding-class-fc1)#match vlan 1-2
(config-forwarding-class-fc1)#exit
(config)#mls qos forwarding-class fc2
(config-forwarding-class-fc1)#match vlan 3-4
(config-forwarding-class-fc1)#exit
(config)#mls qos forwarding-class dp1
```

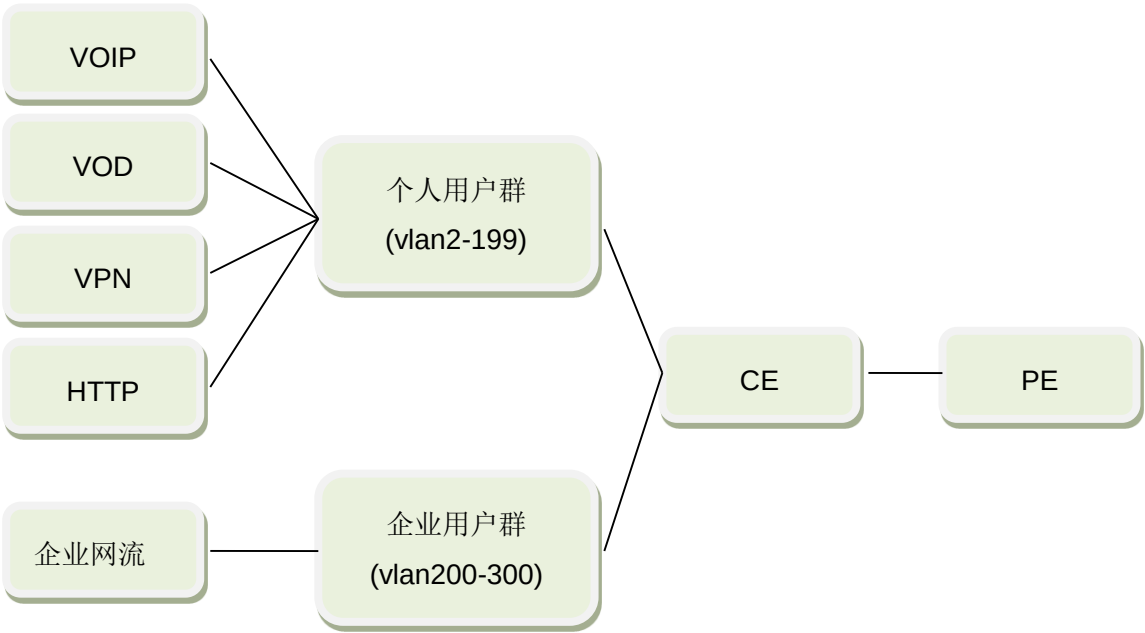


```
(config-drop-profile-dp1)# dp 0 drop-startpoint 20 drop-endpoint 40 max-drop-rate 80
(config-drop-profile-dp1)#exit
(config)#mls qos forwarding-policy fp1
(config-forwarding-policy-fp1)# bandwidth 10 200
(config-forwarding-policy-fp1)#drop dp1
(config-forwarding-policy-fp1)#exit
(config)#mls qos schedule policy p1
(config-schedule-policy-p1)#mls qos schedule level 3 node 1
(config-schedule-policy-p1-lel2nod1)#mls qos schedule queue 1 input
forwarding-class fc1 profile fp1
(config-schedule-policy-p1-lel2nod1)#mls qos schedule queue 2 input
forwarding-class fc2
(config-schedule-policy-p1-lel2nod1)#mls qos bandwidth 64 256
(config-schedule-policy-p1-lel2nod1)#mls qos schedule algorithm wrr
(config-schedule-policy-p1-lel2nod1)#mls qos schedule wrr weight 1 3 5 2 7 4 9 14 8
3 2 14 17 4 7 0
(config-schedule-policy-p1-lel2nod1)#exit
(config-schedule-policy-p1)#exit
(config)#interface ethernet1/1
(config-if-ethernet1/1)#mls qos ets enable
(config-if-ethernet1/1)#mls qos schedule policy bind p1
```

配置结果：

全局启动了 vlan-shaping 功能，端口 ethernet1/1 出口绑定 vlan-shaping 的调度策略 p1。当端口 ethernet1/1 进入的流量超过负载，发生拥塞的时候，vlan1,2 的流量就会进入 level 3 node 1 queue 1 进行调度，vlan3,4 的流量就会进入 level 3 node 1 queue 2 进行调度，参与 vlan-shaping 调度的队列 1, 2 的最小带宽 64kbs 最大带宽 256kbs 队列 1 的最小带宽 10kbs，最大带宽 200kbs，队列 1 和队列 2 按照 wrr 算法进行调度并且出去的流量比例是 1:3。队列 1 中对丢弃优先级为 0 的包,当队列长度达到最大队列长度的 20%时开始丢包，当队列长度达到最大长度的 40%时，开始全速丢包，但丢弃的包占总包的最大比例为 80%。

案例 2:



在 service-vlan 组网中，接入广域网设备的用户基本分为个人用户和企业用户。个人用户的业务类型一致，流量管理要求也基本一致。企业用户的业务类型有很大不同，并要求能够区分不同的企业客户进行不同的流量管理

项目	需求内容
个人用户群的流量策略	承载个人用户群 A 的 service vlan id 范围为 100~299; 个人用户分四种业务： ①VoIP 业务：802.1p 优先级 6 和 7，其中优先级为 6 的流量最小带宽为 20M,优先级为 7 的流量最小带宽为 30M ②Vod 业务：802.1p 优先级为 4 和 5 ③VPN 业务：802.1p 优先级 2 和 3 ④http 浏览业务：802.1p 优先级 0 和 1 ⑤优先调度 VoIP 业务，对于企业网流量，Vod,VPN 和 http 业务分别按 4:3:2:1 的比例进行调度
企业用户群的流量策略	承载企业用户群的 service vlan id 范围为 500~506; 由于存在两组企业用户，使用两组不同的 service vlan 承载不同的企业用户。 ①A 组企业用户：service vlan id 范围为 501~503 ②B 组企业用户：service vlan id 范围为

	504~506 ③企业用户群总流量要求：限速为 100M，最小带宽为 50M。 ④在调度 A 组和 B 组流量时，按 2:1 的比例进行调度

配置思路：

根据组网需求，对于企业用户数据不区分业务类型，都要求优先发送；而对于个人用户数据则要根据报文优先级进行调度，因此需要 **vlan shaping** 功能来实现需求。

①把企业用户数据放入 **vlan shaping** 队列中，把个人用户数据放入普通 **cos** 队列中。企业用户根据 **vlan id** 决定队列号，个人用户数据根据优先级决定队列号

②个人用户数据通过 **S1** 节点进行调度，企业用户数据通过 **S2** 节点进行调度，**S2** 节点的输出放到 **S1** 节点的第 9 个队列上调度

③VoIP,Vod,VPN 和 http 业务分别按照 802.1p 优先级决定它们在 **S1** 节点上的队列号。因此 VoIP 的队列号为 6, 7; Vod 的队列号为 4, 5; VPN 的队列号为 2, 3; http 的队列号为 0, 1;

④优先调度 VoIP 流量，然后按比例 3: 2: 1: 4 调度 Vod,VPN,http 和企业网流量。因此需要配置 **S1** 节点的调度算法为 **WRR**，VoIP 队列的权重为 0，Vod 队列权重为 3，VPN 队列权重为 2，http 队列权重为 1，**vlan shaping** 队列权重为 4

⑤存在两组企业用户 **A** 和 **B**，由于组内企业用户没有差异化服务需求，因此只用构造两个 **vlan shaping** 队列，分别匹配 **Vlan501~503,Vlan504~506**

⑥A 和 B 按比例 2:1 进行调度,因此需要设置 **Vlan shaping** 调度节点的调度算法为 **WRR**，A 和 B 的队列权重分别为 2 和 1

⑦由于企业网用户群总流量要求为最大带宽 100M，最小带宽 50M。因此相当于设置 **vlan shaping** 调度节点的总体最小最大带宽为 50M,100M

⑧VoIP 业务有最小最大带宽需求，相当于对 VoIP 业务对应的队列号 6,7 设置最小最大带宽。

配置步骤（假设交换机出端口为 1/1）：

①首先配置 VoIP 业务的最小最大带宽需求

```
(config)#interface ethernet1/1
```

```
(config-if-ethernet1/1)#mls qos queue 6 bandwidth 20000 0
```

```
(config-if-ethernet1/1)#mls qos queue 7 bandwidth 0 30000
```

②构造出两个转发类，分别匹配 **vlan id501~503**，**vlan id504~506**

```
(config)#mls qos forwarding-class enterpriseA
```

```
(config-forwarding-class-enterpriseA)#match s-vid 501-503
```

```
(config-forwarding-class-enterpriseA)#exit
```

```
(config)#mls qos forwarding-class enterpriseB
```

```
(config-forwarding-class-enterpriseB)#match s-vid 504-506
```

③新建调度策略 P1,设置 S1 节点的调度算法为 WRR, 队列 6, 7 的权重为 0; 队列 4, 5 的权重为 3; 队列 2, 3 的权重为 2, 队列 0, 1 的权重为 1; 队列 8 的权重为 4

```
(config)#mls qos schedule policy p1
```

```
(config-schedule-policy-p1)#mls qos schedule level 1 node 1
```

```
(config-schedule-policy-p1-lel1nod1)#mls qos schedule algorithm wrr
```

```
(config-schedule-policy-p1-lel1nod1)#mls qos schedule wrr weight 1 1 2 2 3 3 0 0 4
```

④设置 vlan shaping 调度节点的调度算法为 wrr, 队列 0 和 1 的权重分别为 2 和 1

```
(config-schedule-policy-p1)#mls qos schedule level 2 node 1
```

```
(config-schedule-policy-p1-lel2nod1)#mls qos schedule algorithm wrr
```

```
(config-schedule-policy-p1-lel2nod1)#mls qos schedule wrr weight 2 1
```

⑤设置 vlan shaping 调度节点的队列输入来源。队列 0 的来源为转发类 enterpriseA, 队列 1 的来源为转发类 enterpriseB。

```
(config-schedule-policy-p1-lel2nod1)#mls qos schedule queue 1 input forwarding-class enterpriseA
```

```
(config-schedule-policy-p1-lel2nod1)#mls qos schedule queue 2 input forwarding-class enterpriseB
```

⑥设置 vlan shaping 调度节点的最小带宽为 50M, 最大带宽为 100M。

```
(config-schedule-policy-p1-lel2nod1)#mls qos bandwidth 50000 100000
```

⑦把调度策略 p1 绑定到端口 1/1

```
(config)#interface ethernet1/1
```

```
(config-if-ethernet1/1)#mls qos ets enable
```

```
(config-if-ethernet1/1)#mls qos schedule policy bind p1
```

6.18.4 Vlan-shaping 排错帮助

流量实际调度效果与用户实际配置不符时, 在确保硬件、线缆等没有问题的前提下, 检查可能存在的情况和建议的解决方法:

- 1、确认是否在出端口绑定了 vlan-shaping 功能, 在绑定以后, 相应的流量才会按照相应的 vlan-shaping 队列比例传输, 否则按照流量 cos 值进行普通 QoS 的调度;
- 2、确认端口的流量是否发生拥塞, 否则, 实际效果可能与配置项不符;
- 3、确认是否是该 vlan 匹配到转发类并且绑定到了调度策略里面,

第7章 可靠性操作

7.1 MSTP

7.1.1 MSTP 介绍

MSTP 是基于 STP 和 RSTP 的一种新的生成树协议。它运行在一个 Bridged-LAN 里的所有网桥上，负责为这个 Bridged-LAN（包括运行 MSTP、RSTP 和 STP 的网桥）计算出一个简单连通的树形活动拓扑(CIST)，为每个 MST 域（MSTP 域）计算出若干个各自独立的多重生成树实例(MSTI)。它应用 RSTP 的快速收敛特性，允许多个具有相同拓扑的 VLAN 映射到一个生成树实例上，而这个生成树拓扑同其它生成树实例相互独立。这种机制一方面用多重生成树实例为映射到它的 VLAN 的数据流量提供了独立的发送路径，实现不同实例间 VLAN 数据流量的负载分担；另一方面，若干个 VLAN 共享同一个拓扑实例（MSTI），同每个 VLAN 对应一个生成树（PVST）的实现方法相比，大大减少了每个网桥需要维持的生成树实例的数量，节约了 CPU 资源，降低了非业务带宽占用。

7.1.1.1 MSTP 域

由于多个 VLAN 可以映射到一个单一的生成树实例，IEEE 802.1s 委员会提出了 MST 域的概念，用来解决如何判断某个 VLAN 映射到哪个生成树实例的问题。

一个 MSTP 域由一个或若干个具有相同 MCID（MST Configuration Identification）的网桥和将网桥互连起来的局域网（域中的某个网桥是该局域网的指定桥，并且该局域网连接的网桥不是运行 STP 的网桥）组成。域中每个网桥维持相同的 MSTIs。

每个域中的网桥具有下述三个属性：

- ☞ 一个包含数字和字母的配置名字（Configuration Name）
- ☞ 一个配置修正级别（Revision Level）
- ☞ 网桥中 VLAN 向生成树实例映射的配置摘要（Configuration Digest）

以上三部分属性组成域的 MCID，只有这三部分属性完全相同，即 MCID 相同的网桥才被 MSTP 认为属于同一个 MST 域。

在整个 Bridged-LAN 的 CIST 中，MSTP 将 MST 域当做一个网桥对待，如下图所示：

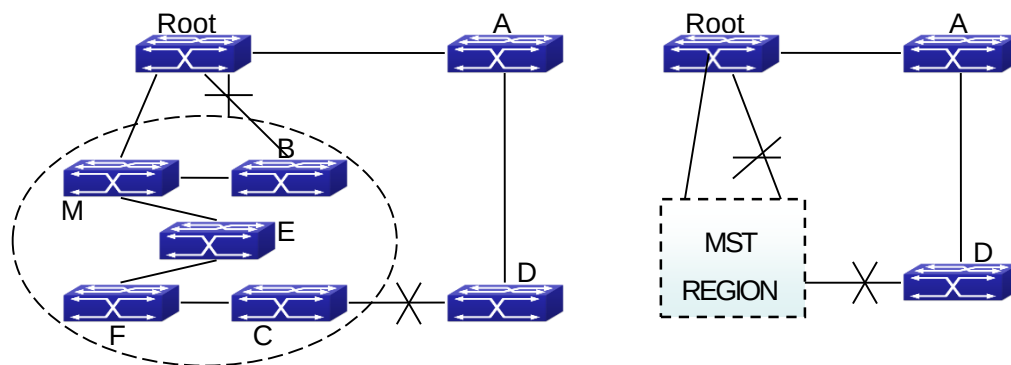


图 1-1 CIST 与 MST 域理解

对上图中的网络，如果网络中的网桥运行 STP 或 RSTP，应当 block（阻塞）网桥 M 和网桥 B 之间的一个端口。但是，如果图中虚线区域部分中的网桥运行 MSTP 并被配置在一个 MST 域中，由于 MSTP 将该域当做一个网桥对待，将会 block 网桥 B 上与根桥（Root）间的端口；同样，MSTP 会 block 网桥 D 上的某个端口。

7.1.1.1.1 MST 域内操作

在一个域内，IST 将域中所有的网桥连接起来。IST 在运行过程中，选出 CIST Regional Root 为它的根网桥，该网桥到 CIST Root 的路径代价和 BridgeID 最小。如果网络中只有一个域，该网桥就是整个网络的 CIST Root；如果 CIST Root 在该域外，则域边缘的某个网桥被选择为 CIST Regional Root。域中 CIST Regional Root 上的根端口是域中所有 MSTIs 的 Master Port。

当一个 MSTP 网桥初始化时，它发送 BPDU，声称自己是 CIST Regional Root，将到 CIST Root 和 CIST Regional Root 的路径代码设为 0。该网桥同时初始化所有 MSTIs，声称自己是所有 MSTIs 的根。如果该网桥收到更优的 CIST/MSTI 根信息（较小的路径代价，BridgeID 等），它就不再认为自己是 CIST 或相应 MSTI 的根。

在一个域中，只有 IST 发送和接收 BPDU。BPDU 中带有各个 MSTI 的信息。因为 MST BPDU 中包含了所有生成树实例的信息，大大减少了为支持多个生成树而需要处理的 BPDU 数量。

MST 域中的所有实例公用相同的协议定时器，但各个实例有各自独立的与拓扑相关的参数，如 Regional Root，根路径代价等。

7.1.1.1.2 MST 域间操作

当网络中存在多个 MST 域或 802.1D 网桥（运行 STP 的网桥）时，MSTP 通过 CST 维持域间或域同 802.1D 网桥间的连接。IST 将域中的网桥连接起来，作为一个虚拟的网桥同相邻的域或 802.1D 网桥连接。

MSTI 的作用范围仅局限于它所在的 MST 域中。一个域中的某个 MST 实例同其它域中的任何 MST 实例无关。域中的网桥通过边缘端口收到另外一个域发送来的 MST BPDU，它

只处理数据中 CIST 的相关信息；而将其中的 MSTI 信息丢弃。

7.1.1.2 端口角色

MSTP 网桥为每一个运行 MSTP 的端口对应于每一个生成树分配一个端口角色。

- ☞ CIST 的端口角色有：Root Port, Designated Port, Alternate Port 和 Backup Port。
- ☞ 每个 MSTI 的端口角色除了上述角色外，还增加了一个新的角色：Master Port。

CIST 和每个 MSTI 的 Root Port, Designated Port, Alternate Port 和 Backup Port 等角色的指定同 RSTP 对应的角色指定相同。

7.1.1.3 MSTP 流量分担的实现

在一个 MST 域中，通过将 VLAN 映射到不同的生成树实例，形成不同的拓扑。各个拓扑实例（包括 IST 和 MSTIs）间相互独立，可以在网桥中配置每个实例各自对应的参数（如网桥优先级（Bridge Priority）、端口代价（Port Cost）等），为网桥和端口指定相应的角色，从而形成拓扑实例中 VLAN 数据流量对应的各自路径，实现 VLAN 流量的分担。具体配置可参考下面 MSTP 举例部分。

7.1.2 MSTP 配置

MSTP 配置任务序列如下：

1. 启动 MSTP 并设置运行模式
2. 配置实例参数
3. 配置 MSTP 域参数
4. 配置 MSTP 的时间参数
5. 配置 MSTP 的快速迁移特性
6. 配置 MSTP 的格式
7. 配置端口的 spanning-tree 属性
8. 配置 MSTP 的认证字窥探属性
9. 配置 MSTP 的拓扑改变 FLUSH 模式

1.启动 MSTP 并设置运行模式

命令	解释
全局配置模式和端口配置模式	
spanning-tree no spanning-tree	在交换机的全局配置模式和端口配置模式下分别启动 MSTP 协议的命令；本命令的 no 操作为关闭 MSTP 协议。
全局配置模式	

spanning-tree mode {mstp stp rstp} no spanning-tree mode	设置 MSTP 的运行模式。
端口配置模式	
spanning-tree mcheck	强制端口迁移到 MSTP 模式下运行。

2.配置实例参数

命令	解释
全局配置模式	
spanning-tree mst <instance-id> priority <bridge-priority> no spanning-tree mst <instance-id> priority	设置交换机在指定实例的网桥优先级。
spanning-tree priority <bridge-priority> no spanning-tree priority	设置交换机的网桥优先级。
端口配置模式	
spanning-tree mst <instance-id> cost <cost> no spanning-tree mst <instance-id> cost	设置当前端口在指定实例的端口路径代价。
spanning-tree mst <instance-id> port-priority <port-priority> no spanning-tree mst <instance-id> port-priority	设置当前端口在指定实例的端口优先级。
spanning-tree mst <instance-id> rootguard no spanning-tree mst <instance-id> rootguard	设置当前端口在指定实例下是否进行根防护，设置根防护的端口不能转为根端口。
spanning-tree rootguard no spanning-tree rootguard	设置当前端口在实例 0 下是否进行根防护，设置根防护的端口不能转为根端口。
spanning-tree [mst <instance-id>] loopguard no spanning-tree [mst <instance-id>] loopguard	配置 spanning-tree 的指定实例上启动 loopguard 功能，本命令的 no 操作恢复为不在该实例下启动该功能。

3.配置 MSTP 域参数

命令	解释
全局配置模式	

spanning-tree mst configuration no spanning-tree mst configuration	进入 MSTP 域配置模式；本命令的 no 操作为恢复交换机的 MSTP 域参数的缺省值。
MSTP 域配置模式	
show	显示当前运行系统的信息。
instance <instance-id> vlan <vlan-list> no instance <instance-id> [vlan <vlan-list>]	创建 Instance 及配置 VLAN 与 Instance 的映射关系。
name <name> no name	配置 MSTP 域的名字。
revision-level <level> no revision-level	配置 MSTP 域的修正级别数值。
abort	退出 MSTP 域配置模式回到全局配置模式，不保存当前对 MSTP 域的配置。
exit	退出 MSTP 域配置模式回到全局配置模式，并保存当前对 MSTP 域的配置。
no	取消一个命令或设置初始值。

4.配置 MSTP 的时间参数

命令	解释
全局配置模式	
spanning-tree forward-time <time> no spanning-tree forward-time	设置交换机转发延时的时间值。
spanning-tree hello-time <time> no spanning-tree hello-time	设置交换机发送 BPDU 报文的 Hello 时间值。
spanning-tree maxage <time> no spanning-tree maxage	设置交换机 BPDU 信息的最大老化时间值。
spanning-tree max-hop <hop-count> no spanning-tree max-hop	设置 BPDU 支持在 MSTP 域中传输的最大跳数。

5.配置 MSTP 的快速迁移特性

命令	解释
端口配置模式	
spanning-tree link-type p2p {auto force-true force-false} no spanning-tree link-type	设置端口的链路类型。

spanning-tree portfast [bpdufilter bpduguard] [recovery <30-3600>] no spanning-tree portfast	设置和取消端口为边缘端口,参数 bpdufilter 和 bpduguard 以及无参数分别表示收到 BPDU 丢弃、收到 BPDU 关闭端口和收到 BPDU 转为非边缘端口。

6.配置 MSTP 的格式

命令	解释
端口配置模式	
spanning-tree format standard spanning-tree format privacy spanning-tree format auto no spanning-tree format	设置端口的格式，standard 格式为遵守 IEEE 标准的格式，privacy 为私有的格式，auto 格式通过自动识别对端的格式决定自己的格式，是默认的格式。在收到对端的格式前使用默认的格式。

7. 配置端口的 spanning-tree 属性

命令	解释
端口配置模式	
spanning-tree cost <cost> no spanning-tree cost	设置端口路径代价。
spanning-tree port-priority <port-priority> no spanning-tree port-priority	设置端口的优先级。
spanning-tree rootguard no spanning-tree rootguard	设置端口为根端口。
全局配置模式	
spanning-tree transmit-hold-count <tx-hold-count-value> no spanning-tree transmit-hold-count	设置端口的最大发送速率。
spanning-tree cost-format {dot1d dot1t}	修改路径开销值的表示方式。

8. 配置 MSTP 的认证字窥探属性

命令	解释
端口配置模式	

spanning-tree digest-snooping no spanning-tree digest-snooping	设置端口上使用对端的认证字，由于个别厂商使用不同于标准的密钥，为了与其在域内互通，在端口上配置了 digest-snooping 命令后，我们在收到对方报文后记录对方的认证字并使用记录的认证字发给对端，以解决与其在域内互通的要求。

9. 配置 MSTP 的拓扑改变 FLUSH 模式

命令	解释
全局配置模式	
spanning-tree tcflush {enable disable protect} no spanning-tree tcflush	设置传播拓扑改变消息时进行 FLUSH 的模式，协议要求是每次拓扑改变都 FLUSH，但在实际环境中，可能不需要也不希望有过多的刷新造成流量不稳定，因此允许根据实际环境设置不同的处理方式， disable 表示不因拓扑改变而刷新， protect 表示每 10 秒最多进行一次刷新，以避免拓扑改变攻击造成的过多刷新。全局的配置作用于所有没有单独配置的端口。本命令的 NO 形式恢复为缺省的 enable 模式，即按协议要求每次刷新。
端口配置模式	
spanning-tree tcflush {enable disable protect} no spanning-tree tcflush	上述命令的端口模式，端口上配置了刷新模式的，不受全局模式的影响。本命令的 NO 形式取消端口上模式的配置，即恢复为默认的使用全局的刷新模式。

7.1.3 MSTP 举例

MSTP 典型应用案例如下：

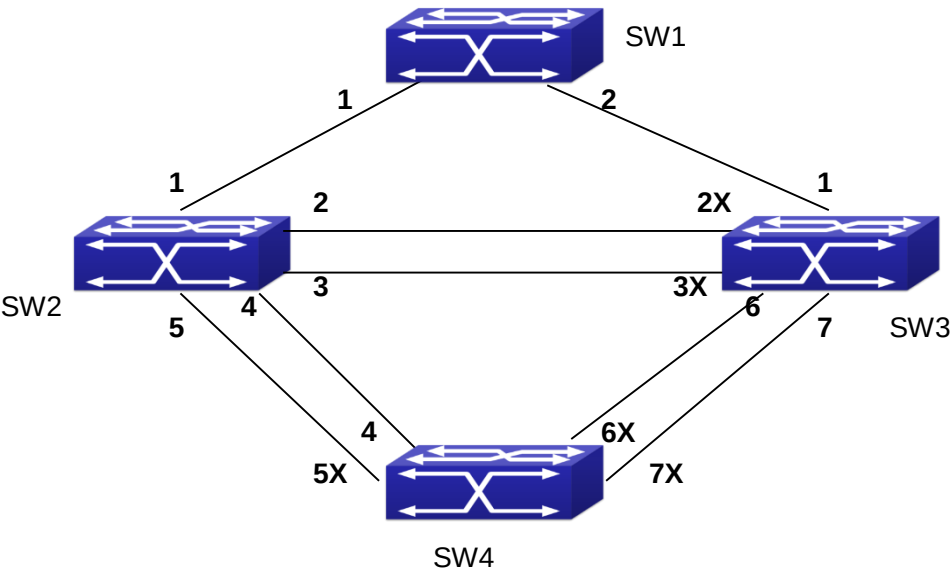


图 1-2 MSTP 典型配置举例

上图中，SW1-SW4 之间的连线如图所示，运行 MSTP 协议。缺省条件下，各交换机运行在 MSTP 模式下，它们的网桥优先级、端口优先级、端口路径代价都为缺省值（都相等）。各交换机的缺省配置如下：

网桥名称		SW1	SW2	SW3	SW4
网桥 MAC 地址		...00-00-01	...00-00-02	...00-00-03	...00-00-04
网桥优先级		32768	32768	32768	32768
端口优先级	端口 1	128	128	128	
	端口 2	128	128	128	
	端口 3		128	128	
	端口 4		128		128
	端口 5		128		128
	端口 6			128	128
	端口 7			128	128
端口路径代价	端口 1	200000	200000	200000	
	端口 2	200000	200000	200000	
	端口 3		200000	200000	
	端口 4		200000		200000
	端口 5		200000		200000
	端口 6			200000	200000
	端口 7			200000	200000

在缺省情况下，MSTP 会自动建立起一个以 SW1 为根网桥的拓扑（用蓝线标记），标记“x”的端口状态为 Discarding，其它端口状态为 Forwarding。

配置更改：

步骤一：配置端口到 vlan 的映射关系：

- ✎ 在交换机 SW2, SW3, SW4 上创建 vlan 20, 30, 40, 50;
- ✎ 在交换机 SW2, SW3, SW4 配置端口 1-7 模式为 trunk。

步骤二：配置交换机 SW2, SW3, SW4 在同一个 MSTP 域内：

- ✎ 配置交换机 SW2, SW3, SW4 域名都为 mstp;
- ✎ 在交换机 SW2, SW3, SW4 上将 vlan 20 和 vlan 30 映射到实例 3 上; 将 vlan 40 和 vlan 50 映射到实例 4 上。

步骤三：配置交换机 SW3 为实例 3 的根网桥; 配置 SW4 交换机为实例 4 的根网桥：

- ✎ 在交换机 SW3 上配置实例 3 对应的网桥优先级为 0;
- ✎ 在交换机 SW4 上配置实例 4 对应的网桥优先级为 0。

配置步骤如下：

交换机 SW2:

```
SW2(config)#vlan 20
SW2(Config-Vlan20)#exit
SW2(config)#vlan 30
SW2(Config-Vlan30)#exit
SW2(config)#vlan 40
SW2(Config-Vlan40)#exit
SW2(config)#vlan 50
SW2(Config-Vlan50)#exit
SW2(config)#spanning-tree mst configuration
SW2(Config-Mstp-Region)#name mstp
SW2(Config-Mstp-Region)#instance 3 vlan 20;30
SW2(Config-Mstp-Region)#instance 4 vlan 40;50
SW2(Config-Mstp-Region)#exit
SW2(config)#interface ethernet 1/1-7
SW2(Config-If-Port-Range)#switchport mode trunk
SW2(Config-If-Port-Range)#exit
SW2(config)#spanning-tree
```

交换机 SW3:

```
SW3(config)#vlan 20
SW3(Config-Vlan20)#exit
```

```
SW3(config)#vlan 30
SW3(Config-Vlan30)#exit
SW3(config)#vlan 40
SW3(Config-Vlan40)#exit
SW3(config)#vlan 50
SW3(Config-Vlan50)#exit
SW3(config)#spanning-tree mst configuration
SW3(Config-Mstp-Region)#name mstp
SW3(Config-Mstp-Region)#instance 3 vlan 20;30
SW3(Config-Mstp-Region)#instance 4 vlan 40;50
SW3(Config-Mstp-Region)#exit
SW3(config)#interface ethernet 1/1-7
SW3(Config-If-Port-Range)#switchport mode trunk
SW3(Config-If-Port-Range)#exit
SW3(config)#spanning-tree
SW3(config)#spanning-tree mst 3 priority 0
```

交换机 SW4:

```
SW4(config)#vlan 20
SW4(Config-Vlan20)#exit
SW4(config)#vlan 30
SW4(Config-Vlan30)#exit
SW4(config)#vlan 40
SW4(Config-Vlan40)#exit
SW4(config)#vlan 50
SW4(Config-Vlan50)#exit
SW4(config)#spanning-tree mst configuration
SW4(Config-Mstp-Region)#name mstp
SW4(Config-Mstp-Region)#instance 3 vlan 20;30
SW4(Config-Mstp-Region)#instance 4 vlan 40;50
SW4(Config-Mstp-Region)#exit
SW4(config)#interface ethernet 1/1-7
SW4(Config-If-Port-Range)#switchport mode trunk
SW4(Config-If-Port-Range)#exit
SW4(config)#spanning-tree
SW4(config)#spanning-tree mst 4 priority 0
```

在上述配置之后，整个网络的实例 CIST（实例 0）以 SW1 为根网桥，在 SW2，SW3，SW4 所在的 MSTP 域内，实例 0 的域根（region root）是 SW2，实例 3 的域根是 SW3；实例 4 的域根是 SW4。vlan 20 和 vlan 30 的流量沿着实例 3 的拓扑发送；vlan 40 和 vlan 50 的流量沿着实例 4 的拓扑发送；其它 vlan 的流量沿实例 0 的拓扑发送。交换机 SW2 上的端口 1 是实例 3 和实例 4 的 Master Port。

MSTP 计算结果包括三个拓扑，实例 0，实例 3 和实例 4，分别如下所示（均用蓝线标记），标记“x”的端口状态为 Discarding，其它端口状态为 Forwarding。由于实例 3 和 4 只在 MSTP 域内有效，图中相关部分只显示其在 MSTP 域内的拓扑。

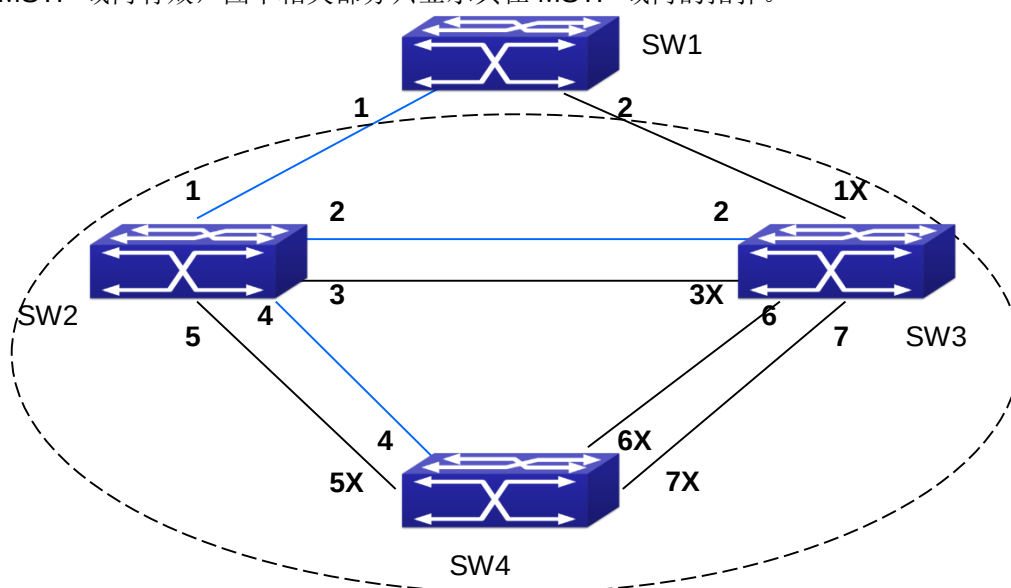


图 1-3 MSTP 变更后的实例 0 的拓扑

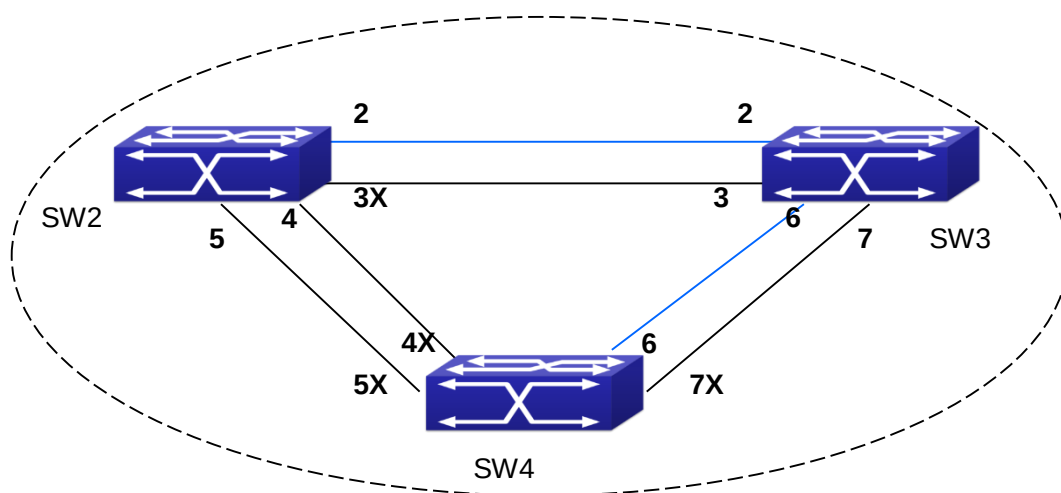


图 1-4 MSTP 变更后 MSTP 域内实例 3 的拓扑

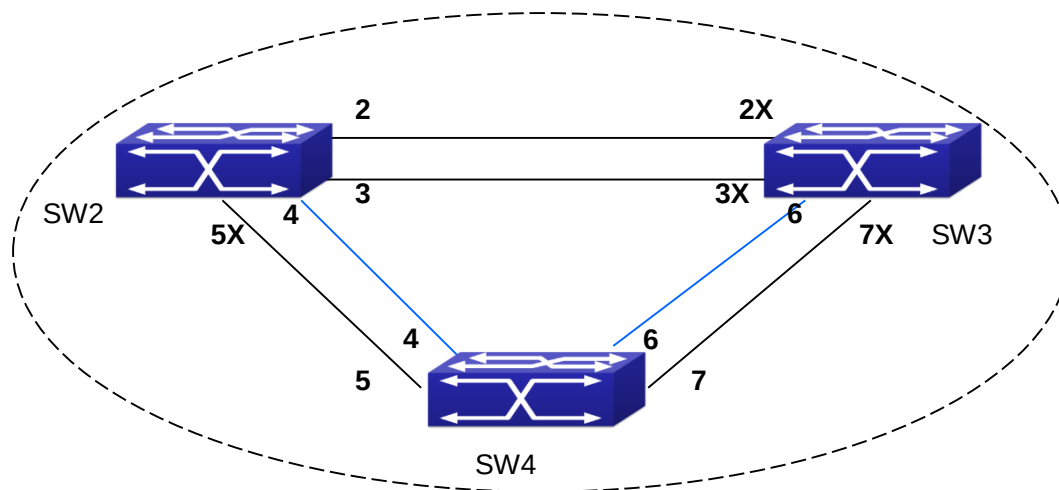


图 1-5 MSTP 变更后 MSTP 域内实例 4 的拓扑

7.1.4 MSTP 排错帮助

- ☞ 如果要在交换机端口上运行 MSTP，首先必须在全局打开 MSTP 开关。在没有打开全局 MSTP 开关之前，打开端口的 MSTP 开关是不允许的。
- ☞ MSTP 定时器参数之间是有相关性的，错误配置可能导致交换机不能正常工作。各定时器之间的关联关系为： $2 \times (\text{Bridge\_Forward\_Delay} - 1.0 \text{ seconds}) \geq \text{Bridge\_Max\_Age}$
 $\text{Bridge\_Max\_Age} \geq 2 \times (\text{Bridge\_Hello\_Time} + 1.0 \text{ seconds})$
- ☞ 用户在修改 MSTP 参数时，应该清楚所产生的各个拓扑。除了全局的基于网桥的参数配置外，其它的是基于各个实例的配置，在配置时一定要注意配置参数对应的实例是否正确。

7.2 ERPS

7.2.1 ERPS 介绍

ERPS（Ethernet Ring Protection Switching）是 ITU-T 定义的一种二层防环协议，标准号为 ITU-T G.8032/Y1344，又称为 G.8032。G.8032 以太环网标准吸取了 EAPS、RPR、SDH、STP 等众多环网保护技术的优点，优化了检测机制，可以检测双向/单向故障，支持多环、多域的结构，在实现 50ms 倒换的同时，支持主备、负荷分担多种工作方式，成为了以太环网技术最新的成熟标准。

ERPS 是一种用于环网保护的防环协议。协议包括：环路故障时链路切换，环路恢复时的通知，以及环路恢复后的链路倒换部分，但它不包括链路故障的发现部分。故障发现可使用 802.1ag 协议定义的 CCM 功能，也可以使用物理层链路故障检测机制。选用何种检测机制都可以，总原则是它必须是灵敏的，可在较短的时间内发现链路故障，并通知给 erps 模

块，erps 要求的环路故障倒换时间是：在 16 个节点，链路长度在 1200km 以内，流量中断时间不超过 50ms，这对链路故障发现和环路保护协议倒换时间要求很高。

7.2.1.1 ERPS 术语

Ethernet ring: 以太环，由许多环节点组成的一个封闭的物理环形网络，环上的每个节点有且只有两个端口连接本环形网络。

Ring protection link: RPL，一条环网上的链路，在环网健康时，为破除环路，被环网上的节点 block 的链路，本链路不能传输数据流量。

RPL owner node: RPL 拥有节点，环网上的连接 RPL 的节点，在环网健康时，负责阻塞环网上 RPL。同时，在环网恢复时，配置为 reversion 方式时，它还会发起链路倒换。

RPL neighbour node: RPL 邻节点，环网上的连接 RPL 的另一个节点，在环网健康时，负责阻塞环网上 RPL。

Interconnection node: 交叉节点，当多个环交叉时，处于交叉位置的节点。在交叉节点，有一个或多个环可以通过交叉节点上的一个端口连接，但最多只有一个环可以通过交叉节点上的两个端口连接。通过一个端口连接的环是子环，通过两个端口连接的环是主环。

R-APS virtual channel: R-APS 虚通道，在子环通路外，两个交叉节点间使子环连通的链路，它的传输特性与子环外网络有关。

major-ring: 主环，连接交叉节点上两个端口的环。

sub-ring: 子环，通过两个交叉节点与其它网络相连的环，子环本身不是一个环形网络，只有通过交叉节点与其它环相连时才组成一个环网。

ERP instance: ERP 实例，是多个保护 vlan 的集合，在这个实例中的 vlan，它们的报文传输经过相同的环网链路，每个 vlan 只能属于一个实例。

Revertive switch: 可逆倒换，RPL 拥有节点在得知环网故障恢复后，恢复 RPL 的阻塞状态，使网络流量传输路径恢复到故障前的链路。

Non-revertive switch: 不可逆倒换，RPL 拥有节点在得知环网故障恢复后，不阻塞 RPL，网络流量传输路径与故障发生时一致。

7.2.1.2 ERPS 功能介绍

7.2.1.2.1 故障倒换

下面以单环单条链路故障为例，说明 erps 协议的工作过程。如图 2-1 所示。

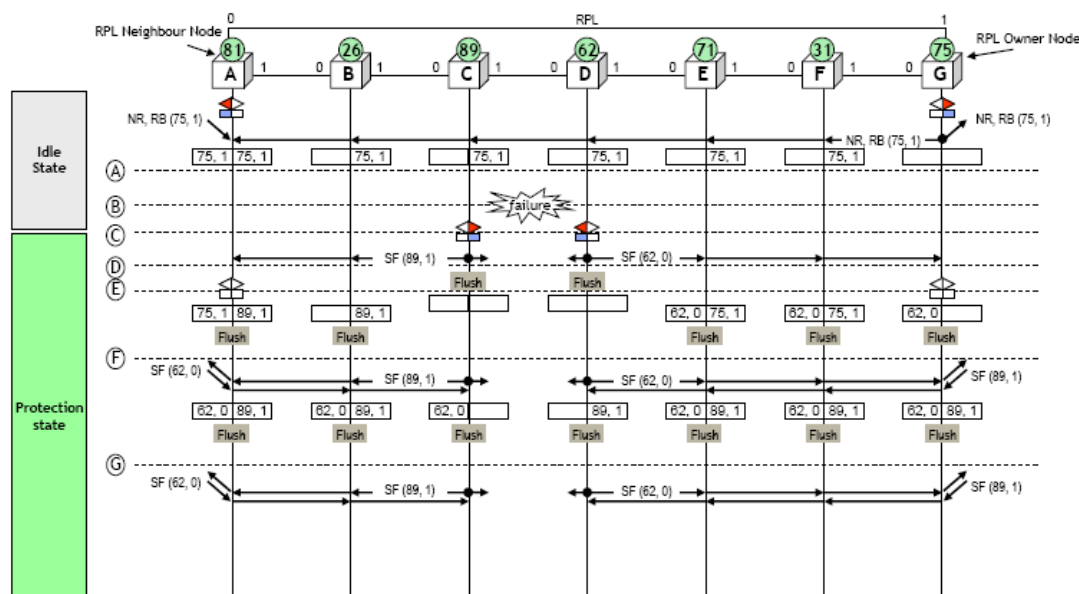


图 7-1 单链路故障

故障倒换的步骤：

- a) 环网状态正常，RPL 拥有节点 G 周期性地发送 R-APS (NR,RB) 报文，本报文说明 RPL 链路处于阻塞状态，环网健康；
- b) 节点 C 和 D 间的链路发生故障；
- c) 节点 C 和 D 检测到故障，它们分别阻塞连接故障链路的端口，执行 flush FDB；
- d) 同时，节点 C 和 D 分别通过连接到环网的端口周期性地发送故障通知报文 R-APS (SF)；
- e) 所有收到 R-APS (SF) 报文的节点执行 flush FDB，同时，RPL 拥有节点 G 和 RPL 邻节点 A 设置 RPL 连接端口为 forward，节点 G 停止发送 R-APS(NR,RB) 报文；
- f) 由于 RPL 链路解除阻塞，所有节点能收到两个 R-APS (SF) 报文（节点 C 和节点 D 发送的），在收到新的 R-APS (SF) 报文后，会再次执行 flush FDB；
- g) 链路故障消息 R-APS (SF) 在环网内一直传输；

7.2.1.2.2 故障恢复

当环链路的故障恢复时，环上节点有两种处理方式：一种是可逆倒换方式，即当故障链路恢复正常后，环网阻塞 RPL，恢复故障链路的转发状态，此时，数据报文的转发路径同环网前一次正常时一致。另一种是不可逆倒换方式，即当故障链路恢复正常后，此链路保持阻塞状态，数据报文继续按照当前的路径转发。

两种方式适用的环境有所不同，当阻塞 RPL 可使网络中的数据流量转发路径最优时，应使用可逆倒换方式；当网络中各链路路径代价差别不大，阻塞哪条路径对于报文转发都没有差别，为避免环路恢复导致数据流量二次中断，可采用不可逆倒换方式。

一、可逆倒换方式

下面以单环单条链路故障为例，说明可逆倒换方式的故障恢复过程。如图 2-2 所示。

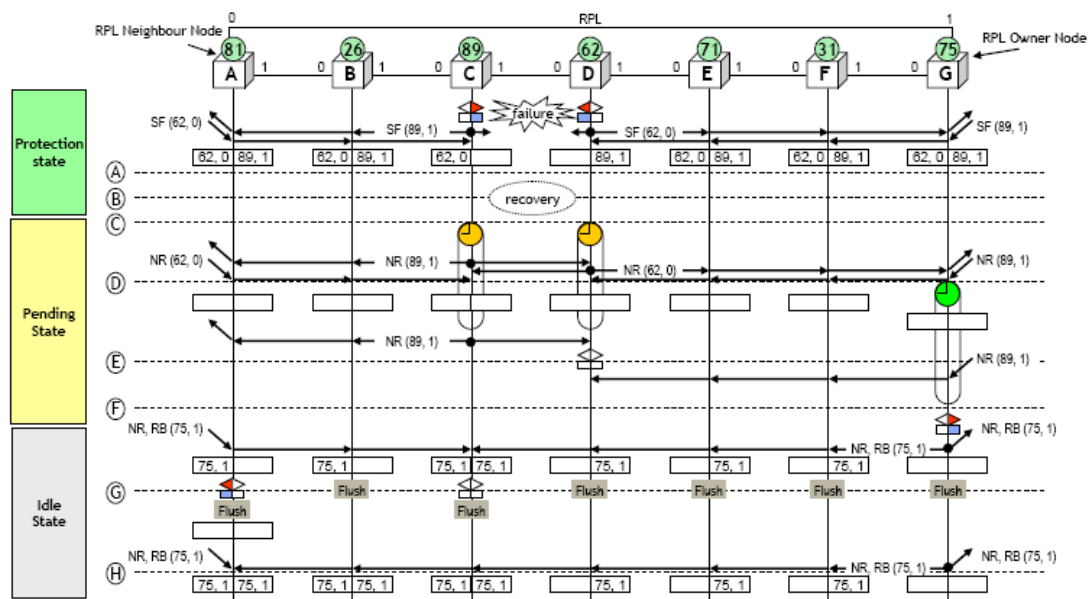


图 7-2 单链路可逆倒换方式故障恢复

可逆倒换方式故障恢复的步骤：

- 故障仍然存在,检测到故障的节点继续周期性地发送携带故障信息的 R-APS(SF) 报文;
- 链路上的故障恢复;
- 节点 C 和 D 检测到故障恢复,它们启动 guard 定时器,同时在环网的端口上发送故障恢复报文 R-APS (NR);
- 当 RPL 拥有节点检查到 R-APS (NR) 报文,它启动 WTR 计时器,同时清除本地记录的节点故障信息;
- 节点 C 和 D 的定时器超时后,它们接收到对端发送的 R-APS (NR) 报文,节点 D 认为节点 C 的优先级更高,因此,它停止发送携带本地信息的 R-APS(NR) 报文,并解除端口的 block;
- 当 RPL 拥有节点 G 的 WTR 定时器超时后,它阻塞连接 RPL 的端口,通过环网的端口发送 R-APS (NR, RB) 报文,通知其它节点 RPL 链路已阻塞,同时,节点 G 执行 flush FDB;
- 节点 C 收到 RPL 拥有节点发送的 R-APS (NR, RB) 报文,它解除本地端口的阻塞,同时停止发送 NR 报文。RPL 邻节点 A 收到此报文,它阻塞连接 RPL 的端口。另外,所有节点收到 R-APS (NR, RB) 报文后,执行 flush FDB。

二、不可逆倒换方式

下面以单环单条链路故障为例，说明不可逆倒换方式的故障恢复过程。如图 2-3 所示。

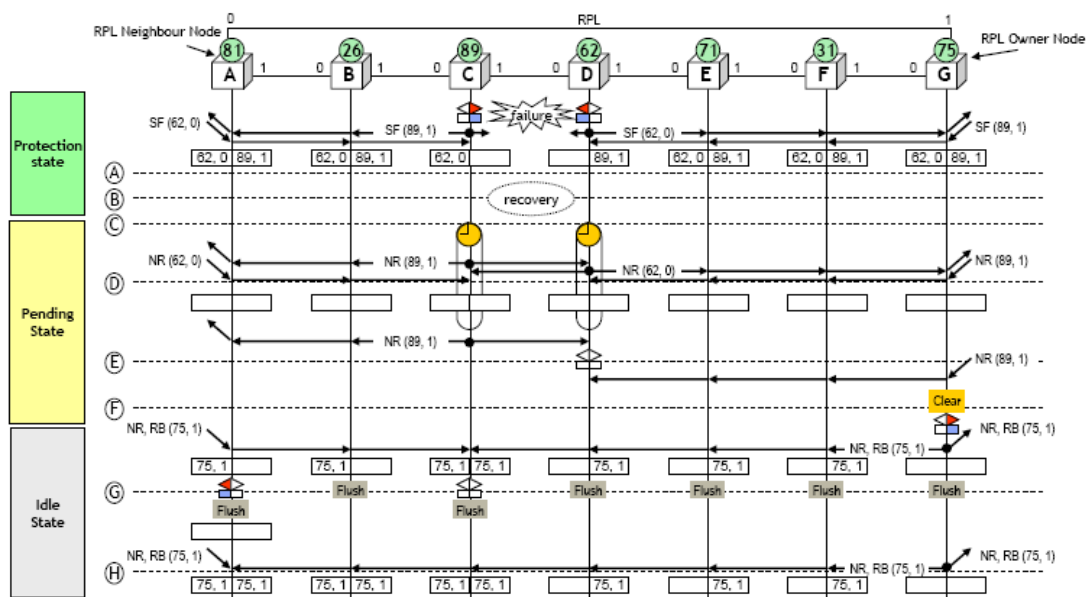


图 7-3 单链路不可逆倒换方式故障恢复

不可逆倒换方式的故障恢复步骤：

- 故障仍然存在,检测到故障的节点继续周期性地发送携带故障信息的 R-APS(SF) 报文;
- 链路上的故障恢复;
- 节点 C 和 D 检测到故障恢复,它们启动 guard 定时器,同时在环网的端口上发送故障恢复报文 R-APS (NR);
- 当 RPL 拥有节点 G 检查到 R-APS (NR) 报文,由于 G 配置了不可逆倒换方式 (non-revertive),它清除本地记录的节点故障信息,但不启动 WTR 计时器;
- 节点 C 和 D 的定时器超时后,它们接收到对端发送的 R-APS (NR) 报文,节点 D 认为节点 C 的优先级更高,因此,它停止发送 R-APS (NR) 报文,并解除本地端口的 block;
- 若 RPL 拥有节点 G 执行 clear 命令,它恢复为可逆倒换方式 (revertive),它会阻塞连接 RPL 的端口,通过环网的端口发送 R-APS 报文 (NR, RB),通知其它节点 RPL 链路已阻塞,同时,执行 flush FDB;
- 节点 C 收到 RPL 拥有节点发送的报文,它解除本地端口的阻塞,同时停止发送 R-APS (NR) 报文; RPL 邻节点 A 收到此报文,它阻塞连接 RPL 的端口。另外,所有节点收到报文后,执行 flush FDB。

7.2.1.2.3 交叉环模型

erps 协议可支持交叉环的保护倒换。交叉环模型分两种：有虚通道的交叉环模型和无虚通道的交叉环模型。

一、有虚通道的交叉环

如图 2-4,三个环网相交,其中 ring 1 是主环,它由环节点 A,B,G,H 以及四个节点间的链路组成,ring 1 在健康状态时,阻塞节点 A,B 间的链路。ring 2 是另一个主环,它由环节点 C,D,E,F 以及四个节点间的链路组成,ring 2 在健康状态时,阻塞节点 C,D 间的链

路。ring 3 是一个子环，它由节点 B,C,F,G 及 B-C 和 G-F 链路组成，ring 3 在健康状态时，阻塞 B-C 链路。B-G 链路是 ring 1 和 ring 3 的交叉链路，属于 ring 1，C-F 链路是 ring 2 和 ring 3 的交叉链路，属于 ring 2。ring 1 和 ring 2 是一个闭合环网，ring 3 并不是一个环网。若将 ring 1 看成 ring 3 的交叉节点 B,G 间的链路（虚通道），ring 2 看成 ring 3 的交叉节点 C,F 间的链路（虚通道），那么 ring 3 也是一个环网。

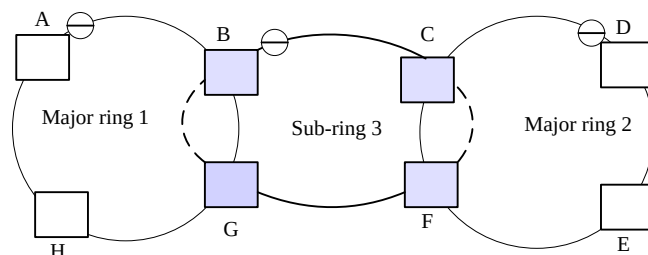


图 7-4 有虚通道的交叉环拓扑

ring 1 和 ring 2 提供的 R-APS 虚通道将 ring 3 协议报文看成是数据报文，报文传输方式与数据报文相同，ring 3 节点 B 发送和接收节点 G 发送的 ring 3 erps 协议报文，同时，节点 G 发送和接收节点 B 发送的 ring 3 erps 协议报文。为了在 ring 1 和 ring 2 中将 ring 3 的 erps 报文与本主环内的 erps 报文区分，可以对每个环的协议报文传输使用不同的 control vlan。

当子环 ring 3 有拓扑变化时，需通知环 ring 1 和 ring 2，主环上的节点执行 flush FDB，主环 ring 1 和 ring 2 的拓扑变化不会影响子环 3，另外，环 ring 1 和 ring 2 的拓扑变化也不会相互影响。

二、无虚通道的交叉环

将上面的三个相交环网换一种方式理解，建模如图 2-5。其中 ring 1 是主环，它由环节点 A,B,G,H 以及四个节点间的链路组成，ring 1 在健康状态时，阻塞 A-B 链路。ring 2 是一个子环，它由环节点 C,D,E,F 以及 C-D,D-E,E-F 链路组成，ring 2 在健康状态时，阻塞 C-D 链路。ring 3 是另一个子环，它由节点 B,C,F,G 以及 B-C,C-F,F-G 链路组成，ring 3 在健康状态时，阻塞 B-C 链路。B-G 链路是 ring 1 和 ring 3 的交叉链路，属于 ring 1，C-F 链路是 ring 2 和 ring 3 间的交叉链路，属于 ring3。ring 1 是一个闭合的环网，ring 2 和 ring 3 不是环网。

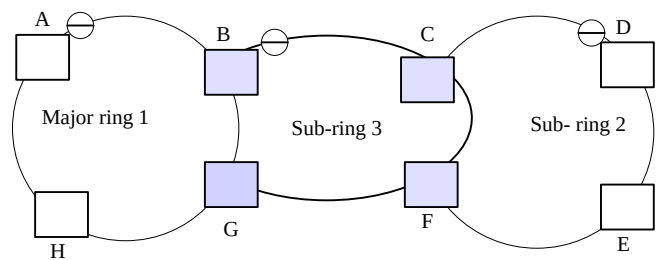


图 7-5 无虚通道的交叉环拓扑

虽然 ring 2 和 ring 3 不是环网，但这两个子环的 erps 报文也需要能传输到所有的环上节点，因此，若 B-C,C-D 间链路阻塞，阻塞链路应该也能够传输 erps 协议报文，阻塞链路的端点 B,C 和 C,D 应该也可以接收和发送 erps 协议报文。

当子环 ring 3 有拓扑变化时，需要通知主环 ring 1，主环上的节点执行 flush FDB 操作，对子环 ring 2 没有影响。当子环 ring 2 有拓扑变化时，会影响子环 ring3 和主环 ring1，环上节点需执行 flush FDB。然而，主环 ring 1 的拓扑变化不会影响子环 ring 2 和 ring 3。

7.2.1.3 ERPS 应用场景

ERPS 用于环形组网，位于汇聚层，汇聚环可完成业务的二层汇聚，同时接入上级三层网络进行业务处理。汇聚环运行 ERPS 协议，提供汇聚环的二层冗余保护倒换功能。

7.2.2 ERPS 配置

- ERPS 配置任务序列如下：
- 1) 创建实例，映射对对应要保护的 vlan
 - 2) 创建 ERPS 环，并配置成员端口信息。其它为默认配置：支持 V2 版本、主环封闭类型、监控端口物理状态
 - 3) 配置 ERPS 环实例，并配置保护实例，端口角色，并选择性配置 ERPS 环实例名字、R-APS 等级、定时器等信息，最后配置控制 VLAN，选择端口配置成 RPL owner 和 RPL Neighbor

1. 创建 MSTP 实例

命令	解释
全局配置模式	
spanning-tree mst configuration no spanning-tree mst configuration	进入交换机的 MST 配置模式，在交换机的 MST 配置模式下，可配置交换机有关 MSTP 域的参数；本命令的 no 操作作为恢复交换机的 MSTP 域参数的缺省值

MST 配置模式	
instance <instance value> vlan <vlan-list> no instance [instance-value]	设置配置实例，映射需要保护的 vlan； 本命令的 no 操作为删除指定的实例。

2. 创建 ERPS 环，配置成员端口信息

命令	解释
全局配置模式	
erps ring <ring-name> no erps ring <ring-name>	创建 ERPS 环，并进入 ERPS 环配置模式；本命令的 no 操作为删除指定的 erps 环。
端口配置模式	
erps-ring <ring-name> port0 erps-ring <ring-name> port1 erps-ring <ring-name> port0 erps-ring <ring-name> port1	配置一个端口的为环节点 port0 或者 port1；本命令的 no 操作为删除端口环节点的 port0 或者 port1 属性。

3. 配置 ERPS 环实例

命令	解释
全局配置模式	
erps ring <ring-name> no erps ring <ring-name>	创建 ERPS 环，并进入 ERPS 环配置模式；本命令的 no 操作为删除指定的 erps 环。
ERPS 环配置模式	
eprs-instance <instance-id> no eprs-instance <instance-id>	创建 ERPS 环上实例，并进入 ERPS 环实例配置模式；本命令的 no 操作为删除指定的环节点实例。
description <instance-name> no description	配置 ERPS 实例的描述字符串；本命令 no 操作为删除指定实例描述字符串
rpl {port0 port1} {owner neighbour} no rpl {port0 port1}	配置 ERPS 环实例的成员端口为 RPL owner 或 RPL neighbour；本命令 no 操作为删除指定 owner 或者 neighbor 节点
raps-mel <level-value> no raps-mel	配置 R-APS channel 的 level，协议报文中的 MEL 字段用于检测当前报文是否能通过；本命令 no 操作为删除 R-APS channel 的 level
protected-instance <instance-list> no protected-instance	配置 ERPS 环实例的保护实例，本命令 no 操作为删除 ERPS 环实例的保护实

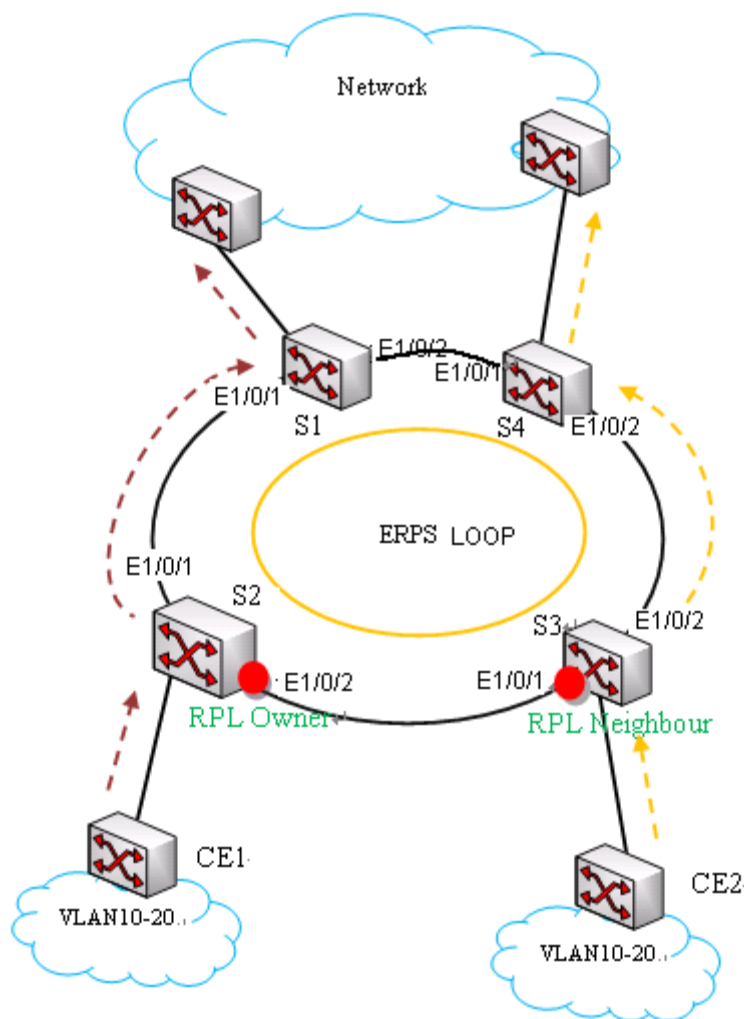
	例
wtr-timer <wtr-times> no wtr-timer	配置 WTR 定时器。WTR 定时器用来避免 RPL 拥有节点由于周期性（间断性）故障引起的频繁保护切换操作；本命令 no 操作为删除 wtr 定时器。
guard-timer <guard-times> no guard-timer	配置 Guard 定时器。Guard 定时器用于以太网环节点避免依据过时的 R-APS 报文作出错误处理，避免形成封闭环路；本命令 no 操作为删除 guard 定时器。
holdoff -timer <holdoff-times> no holdoff -timer	配置 Holdoff 定时器。Holdoff 定时器用于以太网环节点阻塞故障上报时间；本命令 no 操作为删除 Holdoff 定时器。
control-vlan <vlan-id> no control-vlan	配置 R-APS channel 传输 R-APS 报文的控制 VLAN。在 ERPS 环实例中，该 VLAN 只用来传递 ERPS 协议报文，不用来转发用户业务报文，从而提高了 ERPS 协议的安全性；本命令 no 操作为删除 Control Vlan。

4. 显示 ERPS 的配置信息

命令	解释
全局配置模式	
show erps ring {<ring-name> brief}	显示 ERPS 环信息。
show erps instance [ring <ring-name> [instance <instance-id>]]	显示 ERPS 环实例信息。
show erps status [ring <ring-name> [instance <instance-id>]]	显示 ERPS 环实例的状态信息。
show erps statistics [ring <ring-name> [instance <instance-id>]]	显示 ERPS 环实例的统计信息。

7.2.3 ERPS 举例

案例 1:



如上图所示是 ERPS 的配置应用的说明图，S1～S4 组成环形网，提供二层冗余保护倒换功能。为了防止 VLAN10 ~ VLAN20 内的报文产生环路，在组成环网的设备上部署 ERPS 协议。通过 CE1 接入的用户数据转发路径为 S2-S1，通过 CE2 接入的用户数据转发路径 S3-S4。为了实现以太网环路切换保护，可以按如下的步骤进行配置。

1. 配置思路

按照如下思路配置 ERPS 环路冗余保护：

创建 ERPS 环 major_ring1，并配置环成员端口；

配置 ERPS 环 major_ring1 上实例 1，并配置保护实例、成员端口角色、定时器和控制 VLAN。

2. 配置步骤

步骤 1 在 S1 ~ S4 上创建实例 2，映射 VLAN 2，10-20，VLAN2 用于传递协议报文，VLAN10-20 用于传递数据报文。

配置 S1。

```
S1#config
```

```
S1(config)#spanning-tree mst configuration
```

```
S1(Config-Mstp-Region) instance 2 vlan 2;10-20
```

```
S1(Config-Mstp-Region)#exit
```

```
S1(config)#interface e1/1-2
```

```
S1(Config-If-Port-Range)#switchport mode trunk
```

S2、S3、S4 配置同 S1。

步骤 2 创建 ERPS 环，并配置成员端口信息。其它为默认配置：支持 V2 版本、主环封闭类型、监控端口物理状态。

配置 S1。

```
S1(config)#erps-ring major_ring1
```

```
S1(config-erps-ring)#exit
```

```
S1(config)# interface e1/1
```

```
S1(config-if-ethernet1/1)erps-ring major_ring1 port 0
```

```
S1(config-if-ethernet1/1)interface e1/2
```

```
S1(config-if-ethernet1/2)erps-ring major_ring1 port 1
```

S2、S3、S4 配置同 S1。

步骤 3 配置 ERPS 环实例，并配置保护实例，端口角色，并选择性配置 ERPS 环实例名字、R-APS 等级、定时器等信息，最后配置控制 VLAN。将 S2 的 e1/2 端口配置为 RPL Owner，S3 的 e1/1 端口配置为 RPL Neighbour。

配置 S1。

```
S1(config)# erps-ring major_ring1
```

```
S1(config-erps-ring)#erps-instance 1
```

```
S1(config-erps-ring-inst-1)#description instance1
```

```
S1(config-erps-ring-inst-1)#raps-mel 3
```

```
S1(config-erps-ring-inst-1)#protected-instance 2
```

```
S1(config-erps-ring-inst-1)#wtr-timer 8
```

```
S1(config-erps-ring-inst-1)#guard-timer 100
```

```
S1(config-erps-ring-inst-1)#holdoff-timer 5
```

```
S1(config-erps-ring-inst-1)# control-vlan 2
```

S4 配置同 S1。

配置 S2。

```
S2(config)# erps-ring major_ring1
```

```
S2(config-erps-ring)#erps-instance 1
```

```
S2(config-erps-ring-inst-1)#description instance1
```

```
S2(config-erps-ring-inst-1)#rpl port 1 owner
```

```
S2(config-erps-ring-inst-1)#non-revertive
```

```
S2(config-erps-ring-inst-1)#raps-mel 3
```

```
S2(config-erps-ring-inst-1)#protected-instance 2
```

```
S2(config-erps-ring-inst-1)#wtr-timer 8
S2(config-erps-ring-inst-1)#guard-timer 100
S2(config-erps-ring-inst-1)#holdoff-timer 5
S2(config-erps-ring-inst-1)# control-vlan 2
```

配置 S3。

```
S3(config)# erps-ring maijor_ring1
S3(config-erps-ring)#erps-instance 1
S3(config-erps-ring-inst-1)#description instance1
S3(config-erps-ring-inst-1)# rp0 port 1 neighbour
S3(config-erps-ring-inst-1)#raps-mel 3
S3(config-erps-ring-inst-1)#protected-instance 2
S3(config-erps-ring-inst-1)#wtr-timer 8
S3(config-erps-ring-inst-1)#guard-timer 100
S3(config-erps-ring-inst-1)#holdoff-timer 5
S3(config-erps-ring-inst-1)# control-vlan 2
```

步骤 4 查看配置结果。上述配置成功后，以 S2 为例查看配置结果。

```
S2# show erps ring brief
```

Ring-ID	Description	Ring-topo	Port0	Port1	Version
Inst-Count					

1	maijor_ring1	maijor-ring	1/1	1/2	V2	1

```
Switch#show erps instance
```

```
ERPS Ring maijor_ring1
```

```
Instance 1
```

```
Description: instance1
```

```
Protected Instance : 2
```

```
Revertive mode: revertive
```

```
RAPS MEL: 3
```

```
R-APS-Virtual-Channel:
```

```
Control Vlan : 2
```

```
Guard Timer (csec) : 100
```

```
Holdoff Timer (seconds) : 5
```

```
WTR Timer (min) : 8
```

Port	Role	Port-Status
port0	Common	Forwarding
port1	RPL Owner	Blocked

3. 配置文件

S1 的配置文件:

```
S1#config
S1(config)#erps-ring major_ring1
S1(config)#spanning-tree mst configuration
S1(Config-Mstp-Region) instance 2 vlan 2;10-20
S1(Config-Mstp-Region)#exit
S1(config)#interface e1/1-2
S1(Config-If-Port-Range)#switchport mode trunk
S1(Config-If-Port-Range)#exit
S1(config)# interface e1/1
S1(config-if-ethernet1/1)erps-ring major_ring1 port 0
S1(config-if-ethernet1/1)interface e1/2
S1(config-if-ethernet1/2)erps-ring major_ring1 port 1
S1(config-if-ethernet1/2)exit
S1(config)#erps-ring major_ring1
S1(config-erps-ring)#erps-instance 1
S1(config-erps-ring-inst-1)#description instance1
S1(config-erps-ring-inst-1)#raps-mel 3
S1(config-erps-ring-inst-1)#protected-instance 2
S1(config-erps-ring-inst-1)#wtr-timer 8
S1(config-erps-ring-inst-1)#guard-timer 100
S1(config-erps-ring-inst-1)#holdoff-timer 5
S1(config-erps-ring-inst-1)# control-vlan 2
```

S2 的配置文件:

```
S2#config
S2(config)#erps-ring major_ring1
S2(config)#spanning-tree mst configuration
S2(Config-Mstp-Region) instance 2 vlan 2;10-20
S2(Config-Mstp-Region)#exit
S2(config)#interface e1/1-2
S2(Config-If-Port-Range)#switchport mode trunk
S2(Config-If-Port-Range)#exit
S2(config)# interface e1/1
S2(config-if-ethernet1/1)erps-ring major_ring1 port 0
S2(config-if-ethernet1/1)interface e1/2
S2(config-if-ethernet1/2)erps-ring major_ring1 port 1
```

```
S2(config-if-ethernet1/2)exit
S2(config)#erps-ring maijor_ring1
S2(config-erps-ring)#erps-instance 1
S2(config-erps-ring-inst-1)#description instance1
S2(config-erps-ring-inst-1)#rpl port1 owner
S2(config-erps-ring-inst-1)#non-revertive
S2(config-erps-ring-inst-1)#raps-mel 3
S2(config-erps-ring-inst-1)#protected-instance 2
S2(config-erps-ring-inst-1)#wtr-timer 8
S2(config-erps-ring-inst-1)#guard-timer 100
S2(config-erps-ring-inst-1)#holdoff-timer 5
S2(config-erps-ring-inst-1)# control-vlan 2
```

S3 的配置文件:

```
S3#config
S3(config)#erps-ring maijor_ring1
S3(config)#spanning-tree mst configuration
S3(Config-Mstp-Region) instance 2 vlan 2;10-20
S3(Config-Mstp-Region)#exit
S3(config)#interface e1/1-2
S3(Config-If-Port-Range)#switchport mode trunk
S3(Config-If-Port-Range)#exit
S3(config)# interface e1/1
S3(config-if-ethernet1/1)erps-ring maijor_ring1 port 0
S3(config-if-ethernet1/1)interface e1/2
S3(config-if-ethernet1/2)erps-ring maijor_ring1 port 1
S3(config-if-ethernet1/2)exit
S3(config)#erps-ring maijor_ring1
S3(config-erps-ring)#erps-instance 1
S3(config-erps-ring-inst-1)#description instance1
S3(config-erps-ring-inst-1)#rpl port1 neighbour
S3(config-erps-ring-inst-1)#raps-mel 3
S3(config-erps-ring-inst-1)#protected-instance 2
S3(config-erps-ring-inst-1)#wtr-timer 8
S3(config-erps-ring-inst-1)#guard-timer 100
S3(config-erps-ring-inst-1)#holdoff-timer 5
S3(config-erps-ring-inst-1)# control-vlan 2
```

S4 的配置文件:

```
S4#config
S4(config)#erps-ring major_ring1
S4(config)#spanning-tree mst configuration
S4(Config-Mstp-Region) instance 2 vlan 2;10-20
S4(Config-Mstp-Region)#exit
S4(config)#interface e1/1-2
S4(Config-If-Port-Range)#switchport mode trunk
S4(Config-If-Port-Range)#exit
S4(config)# interface e1/1
S4(config-if-ethernet1/1)erps-ring major_ring1 port 0
S4(config-if-ethernet1/1)interface e1/2
S4(config-if-ethernet1/2)erps-ring major_ring1 port 1
S4(config-if-ethernet1/2)exit
S4(config)#erps-ring major_ring1
S4(config-erps-ring)#erps-instance 1
S4(config-erps-ring-inst-1)#description instance1
S4(config-erps-ring-inst-1)#raps-mel 3
S4(config-erps-ring-inst-1)#protected-instance 2
S4(config-erps-ring-inst-1)#wtr-timer 8
S4(config-erps-ring-inst-1)#guard-timer 100
S4(config-erps-ring-inst-1)#holdoff-timer 5
S4(config-erps-ring-inst-1)# control-vlan 2
```

7.2.4 ERPS 排错帮助

如果用户配置好的 ERPS 环无法实现以太网环路切换保护，在确保硬件、线缆等没有问题的前提下，检查可能存在的情况和建议的解决方法：

- ☞ 检查基本配置是否正确，ERPS 环路中的各个节点的配置的保护实例，control-vlan，wtr-timer，guard-timer，raps-mel 等等是否一致。
- ☞ 检查用户数据流量所在的 vlan 不要跟 control vlan 在同一个 vlan，在 ERPS 环实例中，control vlan 只用来传递 ERPS 协议报文，不用来转发用户业务报文，从而提高了 ERPS 协议的安全性。由用户保证配置唯一性。该 VLAN 在发送 R-APS 报文时作为 vlan tag。实例中所有节点的保护 VLAN 配置必须一致。
- ☞ 建议用户配置的 ERPS 节点上的端口 trunk 口，保证数据报文所在的 vlan 和 control vlan 在配置的保护实例中，ERPS 只保护处于保护实例中的各种数据和协议报文。例如：交换机启动某个协议（CFM，EFM，三层接口）导致交换机往外面发送的协议报文，这时候如果发送（协议）报文的 Vlan ID 不在保护实例中，那么拓扑中会出现环路。

- ☞ 若配置端口状态检测方式为 **fastlink**，则硬件需支持 **fastlink** 功能，即硬中断通知端口状态变化，同时关闭交换机上 **mac** 软学习功能。
- ☞ 若配置为与 **CFM** 联动，需要硬件支持 **CC** 功能，且能达到 **3.3ms** 的 **ccm** 发包能力。

7.3 VRRP

7.3.1 VRRP 简介

VRRP (Virtual Router Redundancy Protocol, 虚拟路由器冗余协议) 是一种容错协议，其目的是利用备份机制来提高路由器（或三层以太网交换机）与外界连接的可靠性。它是为具有多播或广播能力的局域网（最明显的例子就是以太网）而设计的，由 **IETF** 提出，目前应用比较广泛。

通常，一个局域网内的所有主机都会设置一条缺省路由以指明缺省网关，主机发出的那些目的地址不在本网段的报文将通过该缺省路由发往缺省网关，从而实现了局域网内各主机与外部网络的通信。然而，当作为缺省网关的路由器与外部网络相连的通信链路出现故障后，本网段内所有以该网关为缺省路由下一跳的各主机将被迫中断与外部网络的通信。

VRRP 就是为解决上述问题而提出的。VRRP 运行于局域网的多台路由器上，它将这几台路由器组织成一台“虚拟”路由器，或称为一个备份组。在这个备份组中，有一个活动路由器（被称为 **Master**）和一个或多个备份路由器（被称为 **Backup**）。活动路由器将实际承担这个“虚拟”路由器的工作任务（如负责转发各主机送给“虚拟”路由器的报文），而备份路由器则作为活动路由器的备份。

在运行的时候，这个“虚拟”路由器拥有自己的“虚拟”IP 地址，备份组内的路由器也有自己的 IP 地址。但是，由于 VRRP 只在路由器或以太网交换机上运行，所以对于该网段上的各主机来说，这个备份组是透明的，它们仅仅知道这个“虚拟”路由器的“虚拟”IP 地址，而并不知道 **Master** 以及 **Backup** 的实际 IP 地址，因此它们将把自己的缺省网关地址设置为该“虚拟”路由器的“虚拟”IP 地址。于是，本局域网内的各主机就通过这个“虚拟”路由器来与其它网络进行通信。而实际上，它们是通过 **Master** 在与其它网络进行通信。而一旦备份组内的 **Master** 发生故障，**Backup** 将会接替其工作，成为新的 **Master**，继续为本局域网内的各主机提供服务，从而保障本局域网内的各主机可以不间断与外部网络的通信。

我们可以总结一下：在 VRRP 备份组内，总有一台路由器或以太网交换机是活动路由器（**Master**），它完成“虚拟”路由器的工作；该备份组中其它的路由器或以太网交换机作为备份路由器（**Backup**，可以不只一台），随时监控 **Master** 的活动。当原有的 **Master** 出现故障时，各 **Backup** 将自动选举出一个新的 **Master** 来接替其工作，继续为网段内各主机提供路由服务。由于这个选举和接替阶段短暂而平滑，因此，网段内各主机仍然可以正常地使用虚拟路由器，实现不间断地与外界保持通信。

7.3.2 VRRP 配置

配置任务序列：

1. 创建/删除虚拟路由器（必须）
2. 配置 VRRP 虚拟 IP 和接口（必须）
3. 启动/关闭虚拟路由器（必须）
4. 配置 VRRP 辅助参数（可选）
 - (1) 配置 VRRP 抢占模式
 - (2) 配置 VRRP 优先级
 - (3) 配置 VRRP 定时器时间间隔
 - (4) 配置 VRRP 对接口的监控
 - (5) 配置 VRRP 会话监控端口的状态

1. 创建/删除虚拟路由器

命令	解释
全局配置模式	
router vrrp <vrid> no router vrrp <vrid>	创建/删除虚拟路由器。

2. 配置 VRRP 虚拟 IP 和接口

命令	解释
VRRP 协议配置模式	
virtual-ip < ip > no virtual-ip	配置 VRRP 虚拟 IP，no 命令删除虚拟 IP。
interface {IFNAME Vlan <ID>} no interface	配置 VRRP 接口，no 命令删除该接口。

3. 启动/关闭虚拟路由器

命令	解释
VRRP 协议配置模式	
enable	启动该虚拟路由器。
disable	关闭该虚拟路由器。

4. 配置 VRRP 辅助参数

(1) 配置 VRRP 抢占模式

命令	解释
VRRP 协议配置模式	
preempt-mode {true false}	配置 VRRP 的抢占模式。

(2) 配置 VRRP 优先级

命令	解释
VRRP 协议配置模式	
priority < priority >	配置 VRRP 优先级。

(3) 配置 VRRP 定时器时间间隔

命令	解释
VRRP 协议配置模式	
advertisement-interval <time>	配置 VRRP 定时器时间值(单位为秒)。

(4) 配置 VRRP 对接口的监控

命令	解释
VRRP 协议配置模式	
circuit-failover {IFNAME Vlan <ID>} <value_reduced> no circuit-failover	配置 VRRP 对接口的监控，no 命令删除对接口的监控。

(5) 配置 VRRP 会话监控端口的状态

命令	解释
VRRP 协议配置模式	
vrrp track interface {ethernet IFNAME IFNAME} priority <priority_value> no vrrp track interface {ethernet IFNAME IFNAME}	配置 VRRP 会话监控端口的状态。

7.3.3 VRRP 典型案例

如下图，SwitchA 和 SwitchB 为在同一组内互为备份的三层以太网交换机。

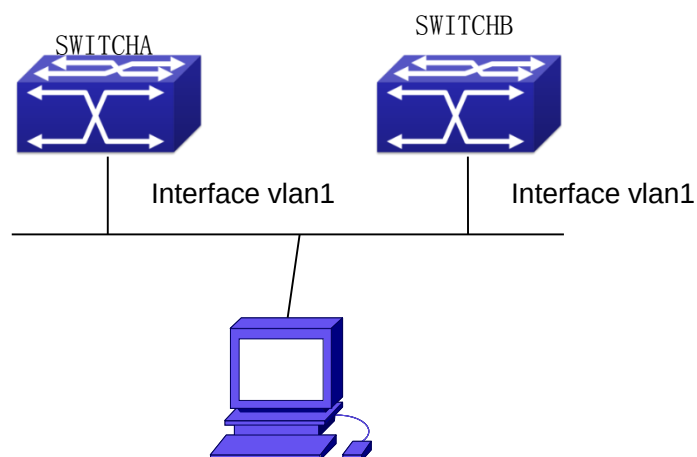


图 7-6 VRRP 网络拓扑示意图

SwitchA 的配置:

```
SwitchA(config)#interface vlan 1
```

```
SwitchA(Config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
```

```
SwitchA(config)#router vrrp 1
```

```
SwitchA(config-router)#virtual-ip 10.1.1.5
```

```
SwitchA(config-router)#interface vlan 1
```

```
SwitchA(config-router)#enable
```

SwitchB 的配置：

```
SwitchB(config)#interface vlan 1
```

```
SwitchB (Config-if-Vlan1)#ip address 10.1.1.7 255.255.255.0
```

```
SwitchB (config)#router vrrp 1
```

```
SwitchB (config-router)#virtual-ip 10.1.1.5
```

```
SwitchB (config-router)#interface vlan 1
```

```
SwitchB (config-router)#enable
```

7.3.4 VRRP 排错帮助

在配置、使用 VRRP 协议时，可能会由于物理连接、配置错误等原因导致 VRRP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 然后，确保在接口上已启动了 VRRP 协议；检查同一备份组内的不同路由器（或三层以太网交换机）认证是否相同；
- ☞ 检查同一备份组内的不同路由器（或三层以太网交换机）配置的 timer 时间是否相同；
- ☞ 检查虚拟 IP 地址是否和接口真实 IP 地址在同一网段内；
- ☞ 如果通过上述检查仍无法解决 VRRP 的问题，那么请使用 `debug vrrp` 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

7.4 MRPP

7.4.1 MRPP 简介

MRPP（Multi-layer ring protection protocol，多环路自动保护协议）是一个用于以太网环路保护的链路层协议。它在以太网环完整时能够防止数据环路引起的广播风暴，而当以太网环上一条链路断开时能迅速恢复环网上各个节点之间的通信通路。

MRPP 协议功能上类似于 STP 协议。同 STP 协议相比，MRPP 具有以下特点：

<1>MRPP 专用于以太网环型拓扑环境

<2>收敛速度快，通常小于 1 秒。理想状况下可以达到 100-50 毫秒。

7.4.1.1 概念介绍

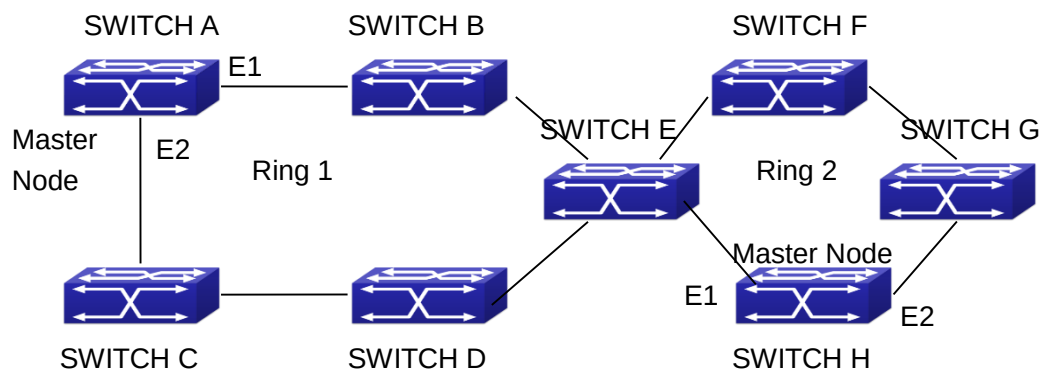


图 7-7 MRPP 示意图

1. 控制VLAN

控制 VLAN 是一个虚拟的 VLAN，仅仅用来区分链路上传递的 MRPP 协议报文。为了避免引起同其它已经配置的 VLAN 的混淆，应尽量避免将控制 VLAN ID 配置为其它已经配置的 VLAN ID。不同的 MRPP 环上应该配置不同的控制 VLAN ID。

2. 以太网环(MRPP环)

环形连接的以太网网络拓扑。

每个 MRPP 环有两种状态。

健康状态：整个环网物理链路是连通的

断裂状态：环网中一处或者多处物理链路断开

3. 节点

以太网环网上每个交换机都称为一个节点。节点由如下角色：

主节点：每个环上有一个主节点，它是发起环路探测和进行环路预防的主要操作节点。

传输节点：每个环上除了主节点外的其他所有节点为传输节点。

节点角色由用户的配置决定。如图 3-1 所示，Switch A 是 Ring 1 的主节点，Switch B、Switch C，Switch D 和 Switch E 为 Ring 1 的传输节点。

4. 主端口和副端口

主节点和传输节点分别有两个端口接入以太网环，其中一个为主端口，另一个为副端口。端口的角色由用户的配置决定。

主节点的主端口和副端口。

主节点的主端口用来发送环路健康检查报文（Hello），副端口用来接收主节点发出的 Hello 报文。当以太网环处在健康状态时，主节点的副端口在逻辑上阻塞其它数据，只允许 MRPP 的报文通过。当以太网环处在断裂状态时，主节点的副端口将解除阻塞状态，转发

数据报文。

传输节点的主端口和副端口在功能上没有区别。

端口的角色由用户的配置决定。如图 3-1 所示，Switch A 的 E1 为主端口，E2 为副端口。

5. 定时器

主节点在发送和接收 MRPP 协议报文时用到两个定时器：Hello 定时器和 Fail 定时器。

Hello 定时器：定义主节点主端口发送健康检测报文的时间间隔的定时器。

Fail 定时器：定义主节点副端口接收健康检测报文的超时时间的定时器。Fail 定时器的值必须大于或等于 Hello 定时器值的 3 倍。

7.4.1.2 MRPP 协议报文类型

报文类型	说明
Hello 报文（健康检查报文）	由主节点的主端口发起，对环路进行检测，如果主节点的副端口能够在配置的超时时间内接收到 Hello 报文，则说明环路正常
LINK-DOWN(链路 Down 事件报文)	传输节点在检测在端口发生 Down 事件后，立即向主节点发送 LINK-DOWN 报文，以通告主节点环路发生故障
LINK-DOWN-FLUSH-FDB 报文	主节点在检测到环路发生故障或者接收到 LINK-DOWN 报文后，打开阻塞的副端口，然后利用两个端口发送该报文，通知各传输节点更新自己的 MAC 地址表
LINK-UP-FLUSH-FDB 报文	由主节点在检测到环路发生故障恢复正常后，利用主端口发出的报文，通知各传输节点更新自己的 MAC 地址表

7.4.1.3 MRPP 协议运行机制

1. 链路Down告警机制

当传输节点发现自己任何一个属于 MRPP 环的端口 Down 时，都会立刻发送链路 Down 报文给主节点。主节点收到链路 Down 报文后立刻解除其副端口的阻塞状态，并发送 LINK-DOWN-FLUSH-FDB 报文通知所有传输节点，使其更新各自的 MAC 地址转发表。

2. 轮询机制

主节点的主端口根据配置的 Hello-timer,定时向其邻居发送 Hello 报文。如果环路是健

康的，主节点的副端口将收到健康检测报文，主节点将保持副端口的阻塞状态。

如果环路是断裂的，主节点的副端口在超时定时器超时后也无法收到健康检测报文。主节点将解除副端口的阻塞状态，同时发送 LINK-DOWN-FLUSH-FDB 报文通知所有传输节点，使其更新各自的 MAC 地址转发表。

3. 环路恢复

主节点在环路故障后，如果副端口收到了主节点发出的 Hello 报文，则证明环路已经恢复，这时主节点阻塞自己的副端口，并向其邻居发送 LINK-UP-FLUSH-FDB 报文。

传输节点上属于 MRPP 环的端口重新 UP 后，主节点可能在过了一段时间后才能发现环路恢复。这段时间对于正常的 VLAN 来说，网络有可能形成一个临时的环路，从而产生广播风暴。为了防止临时环路的产生，传输节点发现自己接入环网的端口重新 UP 后，立即将其临时阻塞（只允许控制 VLAN 的报文通过），只有收到主节点发送的 LINK-UP-FLUSH-FDB 报文之后，才解除该端口的阻塞状态。

7.4.2 MRPP 配置任务序列

- 1) 全局启动 MRPP
- 2) 配置 MRPP 环
- 3) 配置 MRPP 端口查询时间
- 4) 配置兼容模式
- 5) 显示和调试 MRPP 相关信息

1) 全局启动 MRPP

命令	解释
全局配置模式	
mrpp enable no mrpp enable	全局启动和关闭 MRPP。

2) 配置 MRPP 环

命令	解释
全局配置模式	
mrpp ring <ring-id> no mrpp ring <ring-id>	创建 MRPP 环，本命令的 NO 操作删除 MRPP 环及其配置。
MRPP 环配置模式	
control-vlan <vid> no control-vlan	配制控制 VLAN ID，本命令的 NO 操作删除配置的控制 VLAN ID。
node-mode {master transit}	配置 MRPP 环的节点类型（主节点或者是副节点）。

hello-timer <timer> no hello-timer	配置 MRPP 环的主节点发送 Hello 报文定时器，本命令的 NO 操作恢复缺省的定时器值。
fail-timer <timer> no fail-timer	配置 MRPP 环的主节点接收 Hello 报文超时定时器，本命令的 NO 操作恢复缺省的定时器值。
enable no enable	使能 MRPP 环，本命令的 NO 操作关闭使能的 MRPP 环。
端口配置模式	
mrpp ring <ring-id> primary-port no mrpp ring <ring-id> primary-port	指定 MRPP 环的主端口。
mrpp ring <ring-id> secondary-port no mrpp ring <ring-id> secondary-port	指定 MRPP 环的副端口。

3) 配置 MRPP 端口查询时间

命令	解释
全局配置模式	
mrpp poll-time <20-2000>	配置 mrpp 端口状态查询时间间隔。

4) 配置兼容模式

命令	解释
全局配置模式	
mrpp errp compatible no mrpp errp compatible	使能 ERRP 的兼容模式，本命令的 no 操作关闭 ERRP 的兼容模式。
mrpp eaps compatible no mrpp eaps compatible	使能 EAPS 的兼容模式，本命令的 no 操作关闭 EAPS 的兼容模式。
errp domain <domain-id> no errp domain <domain-id>	创建 ERRP 域，本命令的 no 操作删除已经配置的 ERRP 域。

5) 显示和调试 MRPP 相关信息

命令	解释
特权配置模式	
debug mrpp no debug mrpp	打开 MRPP 模块的调试信息，本命令的 NO 操作关闭 MRPP 调试信息输出。
show mrpp [<ring-id>]	显示 MRPP 环的配置信息。
show mrpp statistics [<ring-id>]	显示 MRPP 环的接收数据包统计信息。
clear mrpp statistics [<ring-id>]	清除 MRPP 环的接收数据包统计信息。

7.4.3 MRPP 典型案例

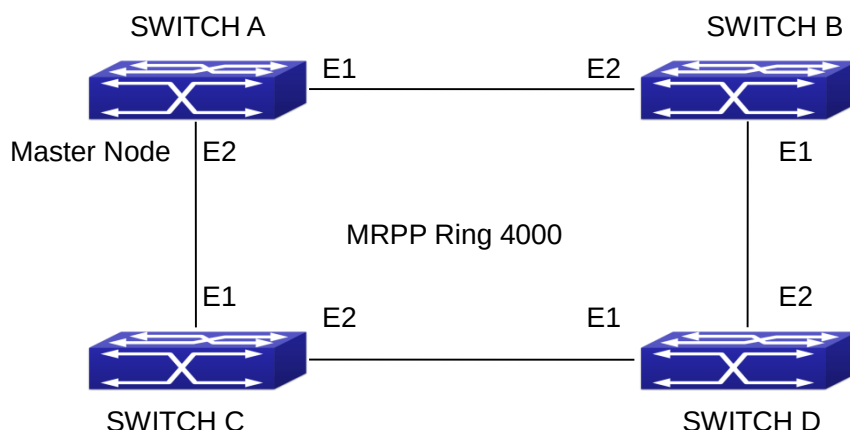


图 7-8 MRPP 典型配置案例

上面的拓扑是大多数情况下使用 MRPP 协议的情形。多个交换机组成一个单一的 MRPP 环，所有的交换机都只配置了一个 MRPP 环 4000，从而组成一个单一的 MRPP 环。

在上面的配置中，SWITCH A 配置为 MRPP 环 4000 的主节点，并且配置 E1/1 为主端口，E1/2 为副端口。其它交换机为 MRPP 环的副节点，分别配置了主端口和副端口。

为了避免环路，在使能整个 MRPP 环的各 MRPP 环时，应该暂时关闭其中主节点的一个端口，等到所有节点配置完成后，再打开该端口。

在关闭 MRPP 环时，需要确保该 MRPP 环没有环路。

SWITCH A 配置任务序列：

```
Switch(Config)#mrpp enable
Switch(Config)#mrpp ring 4000
Switch(mrpp-ring-4000)#control-vlan 4000
Switch(mrpp-ring-4000)#fail-timer 18
Switch(mrpp-ring-4000)#hello-timer 5
Switch(mrpp-ring-4000)#node-mode master
Switch(mrpp-ring-4000)#enable
Switch(mrpp-ring-4000)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-if-Ethernet1/1)#mrpp ring 4000 primary-port
Switch(config-if-Ethernet1/1)#interface ethernet 1/2
Switch(config-if-Ethernet1/2)#mrpp ring 4000 secondary-port
Switch(config-if-Ethernet1/2)#exit
Switch(Config)#
```

SWITCH B 配置任务序列：

```
Switch(Config)#mrpp enable
Switch(Config)#mrpp ring 4000
Switch(mrpp-ring-4000)#control-vlan 4000
Switch(mrpp-ring-4000)#enable
Switch(mrpp-ring-4000)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)#mrpp ring 4000 primary-port
Switch(config-If-Ethernet1/1)#interface ethernet 1/2
Switch(config-If-Ethernet1/2)#mrpp ring 4000 secondary-port
Switch(config-If-Ethernet1/2)#exit
Switch(Config)#
```

SWITCH C 配置任务序列:

```
Switch(Config)#mrpp enable
Switch(Config)#mrpp ring 4000
Switch(mrpp-ring-4000)#control-vlan 4000
Switch(mrpp-ring-4000)#enable
Switch(mrpp-ring-4000)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)#mrpp ring 4000 primary-port
Switch(config-If-Ethernet1/1)#interface ethernet 1/2
Switch(config-If-Ethernet1/2)#mrpp ring 4000 secondary-port
Switch(config-If-Ethernet1/2)#exit
Switch(Config)#
```

SWITCH D 配置任务序列:

```
Switch(Config)#mrpp enable
Switch(Config)#mrpp ring 4000
Switch(mrpp-ring-4000)#control-vlan 4000
Switch(mrpp-ring-4000)#enable
Switch(mrpp-ring-4000)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)#mrpp ring 4000 primary-port
Switch(config-If-Ethernet1/1)#interface ethernet 1/2
Switch(config-If-Ethernet1/2)#mrpp ring 4000 secondary-port
Switch(config-If-Ethernet1/2)#exit
Switch(Config)#
```


7.4.4 MRPP 排错帮助

MRPP 协议的正常运行非常依赖于对 MRPP 环上各个交换机的正常配置，否则极有可能形成环路和广播风暴：

- ☞ 在配置 MRPP 环路时。最好先断开该环路，等待各个交换机配置完毕后再打开该环路。
- ☞ 在关闭 MRPP 环路上一个使能的交换机的 MRPP 环的时候，应确保该 MRPP 环所在的环路已经断开。
- ☞ 在 MRPP 环上出现广播风暴时，首先断开该环路，并确定环上的各个交换机 MRPP 环配置是否正确，如果配置正确，恢复该环路，然后观察环路是否正常。
- ☞ MRPP 环网的收敛时间和端口 up/down 的响应方式有关。若使用查询方式，在简单环网内，MRPP 收敛时间为几百毫秒；若使用中断方式，收敛时间在 50 毫秒以内。
- ☞ 一般情况下，端口扫描配置为查询方式即可。中断方式仅适用于对性能要求较高的环境。相比于中断，查询方式的安全性更高。具体参考端口配置命令 `port-scan-mode {interrupt | poll}`。
- ☞ 如果在正常配置下，仍然形成了环路的广播风暴或者环路不通，请打开主节点的 MRPP 的调试功能，并且利用 `show mrpp statistics` 命令观察主节点和传输节点的状态和统计信息是否正常，并将结果发送给本公司技术服务中心。

7.5 ULPP

7.5.1 ULPP 简介

每个 ULPP 组包括两个上行链路端口，一个主端口(master)，另一个为副端口(slave)。这里的端口可以是物理端口，也可以是 port channel。ULPP 组成员端口有三种状态：Forwarding, Standby, Down。正常情况下，只有一个端口处于转发(Forwarding)状态，另一个端口阻塞，处于待命(Standby)状态。当主端口发生链路故障时，主端口变为 Down 状态，原来处于 Standby 状态的副端口切换到 Forwarding 状态。

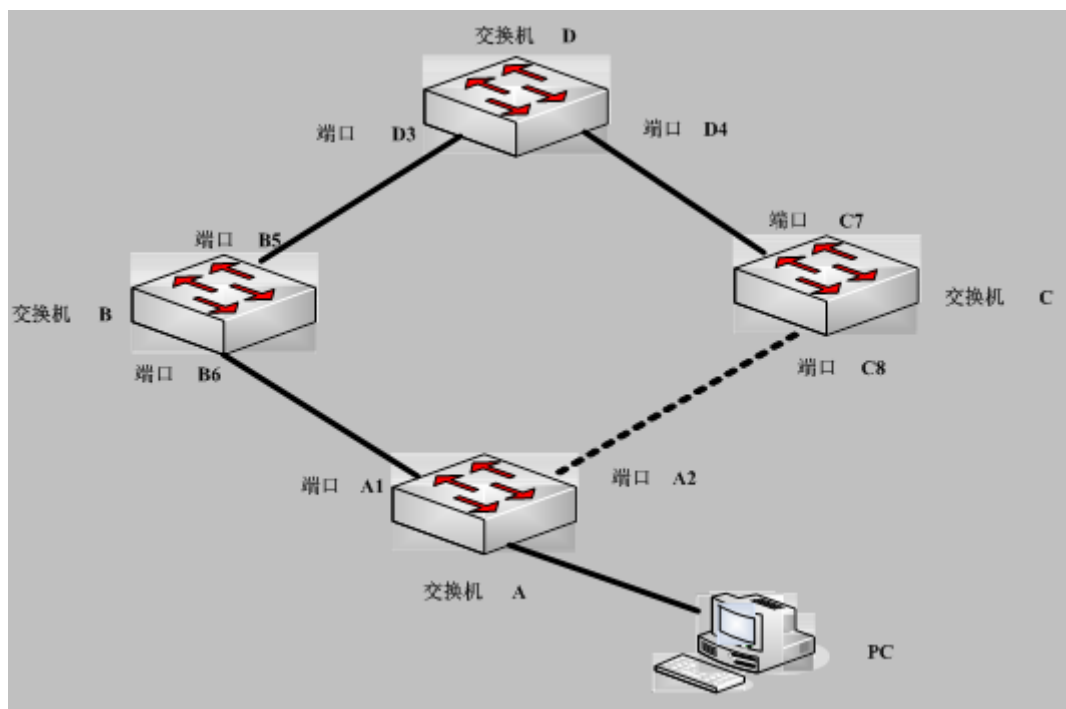


图 7-9 ULPP 使用场景

图 4-1 中使用了双上行组网，这是 ULPP 的典型应用场景。交换机 A 通过 B 和 C 双上行到 D，端口 A1 和 A2 为上行端口。交换机 A 上配置了 ULPP，其中端口 A1 设置为主端口，端口 A2 设置为副端口。当处于 Forwarding 状态的端口 A1 发生故障时，立即进行链路切换，端口 A2 进入 Forwarding 状态。此后，当主端口恢复时，如果没有配置抢占模式，则端口 A2 继续保持 Forwarding 状态，端口 A1 进入 Standby 状态。

通过开启抢占模式，使得主端口从故障恢复时可以对副端口进行抢占。为了避免异常故障导致的频繁链路切换，引入了抢占延时机制，主端口抢占副端口前需要等待一定的时间。为了保持流量的稳定，默认情况下主端口不进行抢占，而是进入 Standby 状态。

对 ULPP 配置时，需要通过 MSTP 实例的方式指定该 ULPP 组所保护的 VLAN，对保护 VLAN 之外的其它 VLAN，ULPP 不提供保护。

当发生链路切换时，网络中设备上原有的转发表项将不适用于新的拓扑。在图 4-1 所示拓扑中，交换机 A 上配置了 ULPP，作为主端口的端口 A1 处于 Forwarding 状态，此时交换机 D 会从端口 D3 上学习到 PC 的 MAC 地址。此后端口 A1 发生故障，流量切换到端口 A2 进行转发，此时如果有通过交换机 D 发往 PC 的数据，仍然会从端口 D3 进行转发，造成数据包的丢失。因此，链路切换时，配置了 ULPP 的设备需要通过切换变为 Forwarding 状态的端口发送 flush 报文，更新网络中其他设备的 MAC 地址表和 ARP 表。ULPP 分别使用两种 flush 报文对表项进行更新：MAC 地址更新报文和 ARP 删除报文。

为了充分利用带宽资源，ULPP 通过配置可以实现 VLAN 负载均衡。如图所示，交换机 A 上配置了两个 ULPP 组：组 1 中端口 A1 做为主端口，端口 A2 为副端口，组 2 中端口 A2 为主端口，端口 A1 为副端口，组 1 和组 2 保护的 VLAN 分别为 1-100 和 101-200。此时端口 A1 和端口 A2 同时处于转发状态，主端口和副端口互为备份，分别转发不同范围 VLAN 的报文。端口 A1 发生故障时，VLAN 1-200 的流量都会由端口 A2 进行转发。此后当端口

A1 恢复正常时，端口 2 继续转发 VLAN 101-200 的数据，VLAN 1-100 的数据切换到端口 A1 转发。

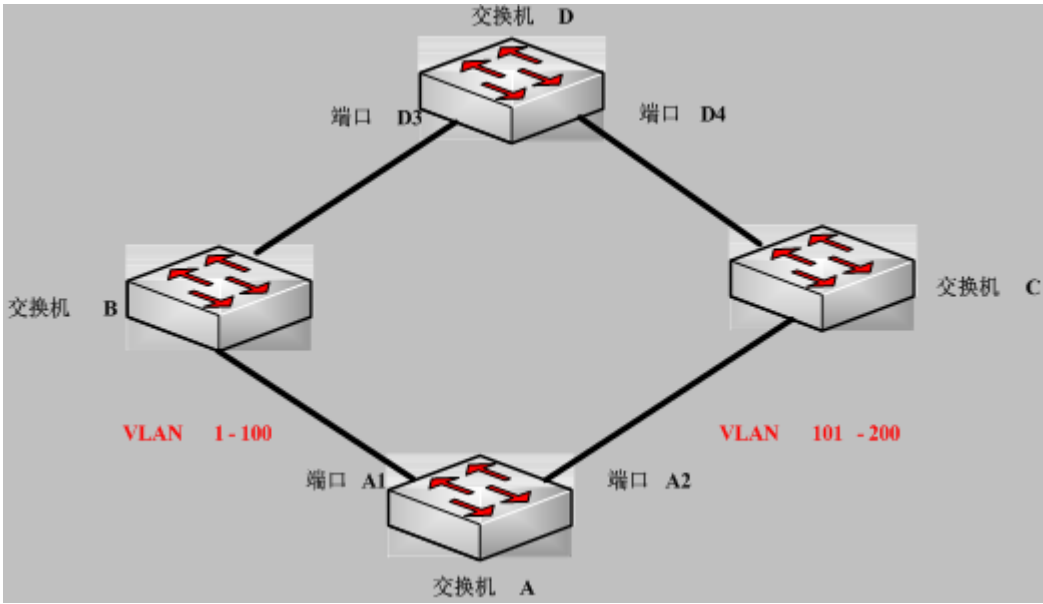


图 7-10 VLAN 负载均衡

7.5.2 ULPP 配置任务序列

- 1) 全局创建 ULPP 组
- 2) 配置 ULPP 组
- 3) 显示和调试 ULPP 相关信息

1) 全局创建 ULPP 组

命令	解释
全局配置模式	
ulpp group <integer> no ulpp group <integer>	全局配置和删除 ULPP 组。

2) 配置 ULPP 组

命令	解释
ULPP 组配置模式	
preemption mode no preemption mode	配置 ULPP 组的抢占模式，no 操作为删除抢占模式。
preemption delay <integer> no preemption delay	配置抢占时延，no 操作恢复默认值 30s。
control vlan <integer> no control vlan	配置发送控制 VLAN，no 操作恢复发送控制 VLAN 的默认值 1。

protect vlan-reference-instance <instance-list> no protect vlan-reference-instance <instance-list>	配置保护 VLAN，no 操作为删除保护 VLAN。
flush enable mac flush disable mac	使能或者关闭 MAC 地址更新的 flush 报文的发送功能。
flush enable arp flush disable arp	使能或者关闭 ARP 删除的 flush 报文的发送功能。
flush enable mac-vlan flush disable mac-vlan	使能或关闭根据 vlan 删除动态单播 mac 的 flush 报文发送功能
description <string> no description	配置或者删除 ULPP 组描述。
端口配置模式	
ulpp control vlan <vlan-list> no ulpp control vlan <vlan-list>	配置接收控制 VLAN，no 操作恢复接收控制 VLAN 的默认值 1。
ulpp flush enable mac ulpp flush disable mac	使能或者关闭 MAC 地址更新的 flush 报文的接收功能。
ulpp flush enable arp ulpp flush disable arp	使能或者关闭 ARP 删除的 flush 报文的接收功能。
ulpp flush enable mac-vlan ulpp flush disable mac-vlan	使能或关闭 mac-vlan 类型的 flush 报文接收功能
ulpp group <integer> master no ulpp group <integer> master	配置或者删除 ULPP 组的主端口。
ulpp group <integer> slave no ulpp group <integer> slave	配置或者删除 ULPP 组的副端口。

3) 显示和调试 ULPP 相关信息

命令	解释
特权配置模式	
show ulpp group [group-id]	显示已配置的 ULPP 组的配置信息。
show ulpp flush counter interface {ethernet <IFNAME> <IFNAME>}	显示 flush 报文统计信息。
show ulpp flush-receive-port	显示端口接收 flush 类型和 control vlan。
clear ulpp flush counter interface <name>	清除 flush 报文统计信息。

debug ulpp flush {send receive} interface <name> no debug ulpp flush {send receive} interface <name>	显示收发 flush 报文信息，no 操作为关闭显示收发 flush 报文信息。
debug ulpp flush content {send receive} interface <name> no debug ulpp flush content {send receive} interface <name>	显示接收到的 flush 报文内容，no 操作为关闭显示 flush 报文内容。
debug ulpp error no debug ulpp error	显示 ULPP 错误信息，no 操作为关闭显示 ULPP 错误信息。
debug ulpp event no debug ulpp event	显示 ULPP 事件信息，no 操作为关闭显示 ULPP 事件信息。

7.5.3 ULPP 典型案例

7.5.3.1 ULPP 典型案例 1

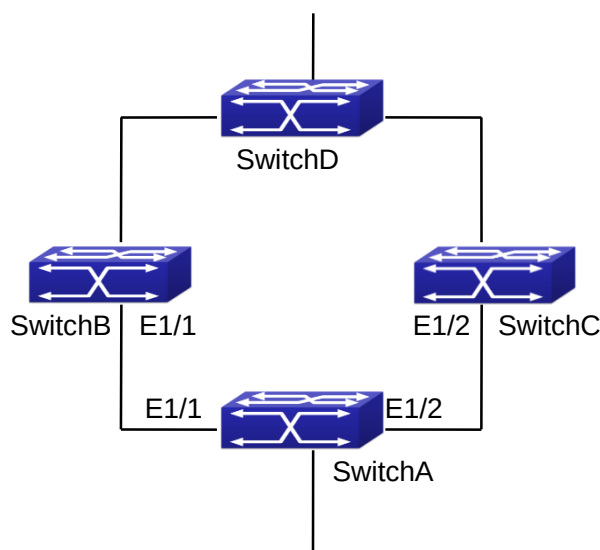


图 7-11 ULPP 典型配置案例 1

上面的拓扑是 ULPP 协议的典型应用环境。

SwitchA 有两条上行链路，分别连接 SwitchB 和 SwitchC，当不开启任何协议的时候，该拓扑成环。为了避免环路，可以在 SwitchA 上配置 ULPP 协议，配置 ULPP 组的主副端口，在主副端口都 up 的情况下，会将副端口置为 **standby** 状态，即不再转发数据报文，在主端口 down 掉的情况下，会将副端口置为 **forwarding** 状态，从而实现了链路的切换。SwitchB 和 SwitchC 上也可以开启接收 flush 报文的命令，用来配合 SwitchA 上的 ULPP 协议的运行，从而达到快速切换链路的目的，尽可能的减少了切换时延。

在配置 SwitchA 上的 ULPP 协议时，首先创建一个 ULPP 组，配置该组的保护 VLAN 为 vlan10，然后配置 interface Ethernet 1/1 为该 ULPP 组的主端口，interface Ethernet 1/2 为该 ULPP 组的副端口，控制 VLAN 为 10。在 SwitchB 和 SwitchC 上配置接收 ULPP 的 flush 报文。

SwitchA 配置任务序列：

```
Switch(Config)#vlan 10
Switch(Config-vlan10)#switchport interface ethernet 1/1; 1/2
Switch(Config-vlan10)#exit
Switch(Config)#spanning-tree mst configuration
Switch(Config-Mstp-Region)#instance 1 vlan 10
Switch(Config-Mstp-Region)#exit
Switch(Config)#ulpp group 1
Switch(ulpp-group-1)#protect vlan-reference-instance 1
Switch(ulpp-group-1)#control vlan 10
Switch(ulpp-group-1)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)# ulpp group 1 master
Switch(config-If-Ethernet1/1)#exit
Switch(Config)#interface Ethernet 1/2
Switch(config-If-Ethernet1/2)# ulpp group 1 slave
Switch(config-If-Ethernet1/2)#exit
```

SwitchB 配置任务序列：

```
Switch(Config)#vlan 10
Switch(Config-vlan10)#switchport interface ethernet 1/1
Switch(Config-vlan10)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)# ulpp flush enable mac
Switch(config-If-Ethernet1/1)# ulpp flush enable arp
Switch(config-If-Ethernet1/1)# ulpp control vlan 10
```

SwitchC 配置任务序列：

```
Switch(Config)#vlan 10
Switch(Config-vlan10)#switchport interface ethernet 1/2
Switch(Config-vlan10)#exit
Switch(Config)#interface ethernet 1/2
Switch(config-If-Ethernet1/2)# ulpp flush enable mac
Switch(config-If-Ethernet1/2)# ulpp flush enable arp
```

```
Switch(config-if-Ethernet1/2)# ulpp control vlan 10
```

7.5.3.2 ULPP 典型案例 2

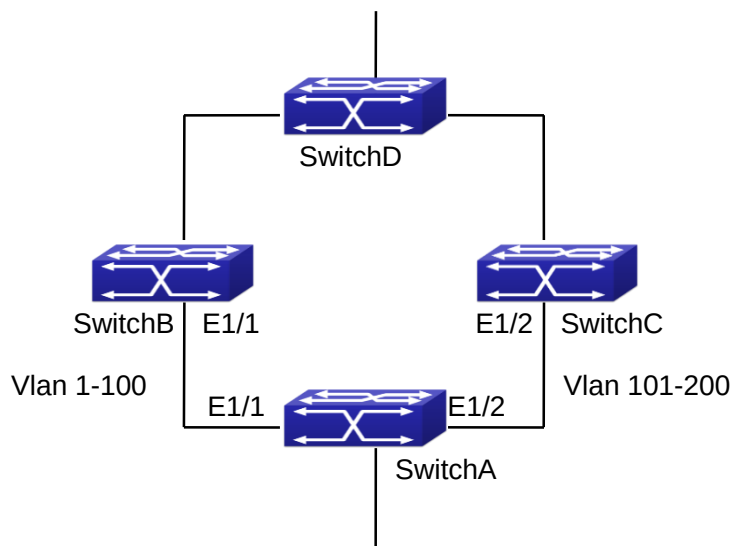


图 7-12 ULPP 典型配置案例 2

ULPP 可以通过配置来实现 VLAN 负载均衡。如图 4-4 所示，switchA 上配置了两个 ULPP 组：组 1 中端口 E1/1 做为主端口，端口 E1/2 为副端口，组 2 中端口 E1/2 为主端口，端口 E1/1 为副端口，组 1 和组 2 保护的 VLAN 分别为 1-100 和 101-200。此时端口 E1/1 和端口 E1/2 同时处于转发状态，主端口和副端口互为备份，分别转发不同范围 VLAN 的报文。端口 E1/1 发生故障时，VLAN 1-200 的流量都会由端口 E1/2 进行转发。此后当端口 E1/1 恢复正常时，端口 E1/2 继续转发 VLAN 101-200 的数据，VLAN 1-100 的数据切换到端口 E1/1 转发。

switchA 配置任务序列：

```
Switch(Config)#spanning-tree mst configuration
Switch(Config-Mstp-Region)#instance 1 vlan 1-100
Switch(Config-Mstp-Region)#instance 2 vlan 101-200
Switch(Config-Mstp-Region)#exit
Switch(Config)#ulpp group 1
Switch(ulpp-group-1)#protect vlan-reference-instance 1
Switch(ulpp-group-1)#preemption mode
Switch(ulpp-group-1)#exit
Switch(Config)#ulpp group 2
Switch(ulpp-group-2)#protect vlan-reference-instance 2
Switch(ulpp-group-2)#preemption mode
Switch(ulpp-group-2)#exit
```

```
Switch(Config)#interface ethernet 1/1
Switch(config-if-Ethernet1/1)#switchport mode trunk
Switch(config-if-Ethernet1/1)#ulpp group 1 master
Switch(config-if-Ethernet1/1)#ulpp group 2 slave
Switch(config-if-Ethernet1/1)#exit
Switch(Config)#interface Ethernet 1/2
Switch(config-if-Ethernet1/2)#switchport mode trunk
Switch(config-if-Ethernet1/2)# ulpp group 1 slave
Switch(config-if-Ethernet1/2)# ulpp group 2 master
Switch(config-if-Ethernet1/2)#exit
```

switchB 配置任务序列:

```
Switch(Config)#interface ethernet 1/1
Switch(config-if-Ethernet1/1)#switchport mode trunk
Switch(config-if-Ethernet1/1)# ulpp flush enable mac
Switch(config-if-Ethernet1/1)# ulpp flush enable arp
```

switchC 配置任务序列:

```
Switch(Config)#interface ethernet 1/2
Switch(config-if-Ethernet1/2)# switchport mode trunk
Switch(config-if-Ethernet1/2)# ulpp flush enable mac
Switch(config-if-Ethernet1/2)# ulpp flush enable arp
```

7.5.4 ULPP 排错帮助

- ☞ 目前对于超过双上行的多上行，可以允许配置，但是会出现环路，所以不建议在超过双上行的链路上配置 ULPP。
- ☞ 如果在正常配置下，仍然形成了环路的广播风暴或者环路不通，请打开 ULPP 的调试功能，复制 3 分钟内的 debug 信息以及配置信息，并将结果发送给本公司技术服务中心。

7.6 ULSM

7.6.1 ULSM 简介

ULSM (Uplink State Monitor) 用来进行端口状态同步。每个 ULSM 组由上行端口和下行端口组成，上行端口和下行端口都可以为多个。端口可以是物理端口，也可以是 port channel，但不能是 port channel 的成员端口，每个端口只能属于一个 ULSM 组。

上行端口是 ULSM 组中被监控的端口。当 ULSM 组中的所有上行端口都为 Down 或

ULSM 组不存在上行端口时，ULSM 组的状态为 Down。只要有一个上行端口为 Up，ULSM 组的状态就为 Up。

下行端口是 ULSM 组中的受控端口。下行端口的状态会随着 ULSM 组 Up/Down 状态而改变，并保持与 ULSM 组的状态一致。

ULSM 主要与 ULPP 配合使用，使得下游设备可以快速感知上游设备链路故障，并做出相应处理。在图 5-1 中所示环境中，交换机 A 上配置了 ULPP，此时通过端口 A1 转发流量，如果交换机 B 与交换机 D 间链路发生故障，由于交换机 A 无法感知上游链路的故障，继续从端口 A1 转发，从而造成流量的丢失。

可以通过在交换机 B 上配置 ULSM 解决上述问题。步骤为：将交换机 B 上的端口 B5 设置为 ULSM 组上行端口，端口 B6 设置为下行端口。当交换机 B 与交换机 D 间链路发生故障时，下行端口 B6 与 ULSM 组状态同步为 Down，从而引发配置了 ULPP 的交换机 A 发生链路切换，避免数据的丢失。

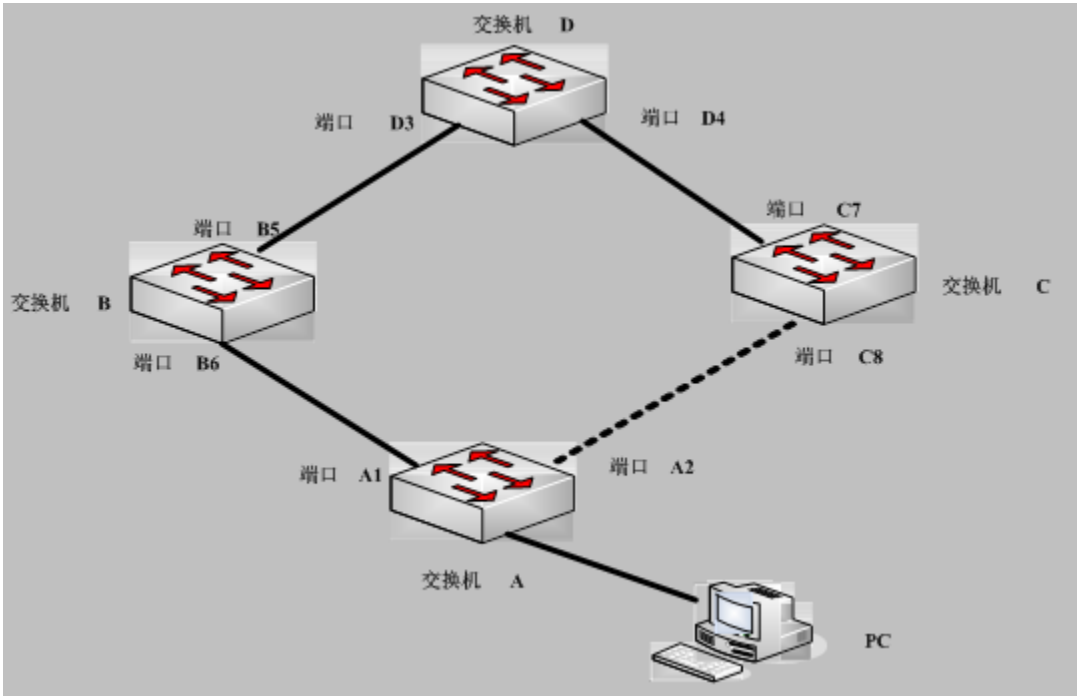


图 7-13 ULSM 使用场景

7.6.2 ULSM 配置任务序列

- 1) 全局创建 ULSM 组
- 2) 配置 ULSM 组
- 3) 显示和调试 ULSM 相关信息

1) 全局创建 ULSM 组

命令	解释
全局配置模式	

ulsm group <group-id> no ulsm group <group-id>	全局配置和删除 ULSM 组。

2) 配置 ULSM 组

命令	解释
端口配置模式	
ulsm group <group-id> {uplink downlink} no ulsm group <group-id> {uplink downlink}	配置 ULSM 组的上行/下行端口，no 命令删除上行/下行端口。

3) 显示和调试 ULPP 相关信息

命令	解释
特权配置模式	
show ulsm group [group-id]	显示已配置的 ULSM 组的配置信息。
debug ulsm event no debug ulsm event	显示 ULSM 事件信息，no 操作关闭显示 ULSM 事件信息。

7.6.3 ULSM 典型案例

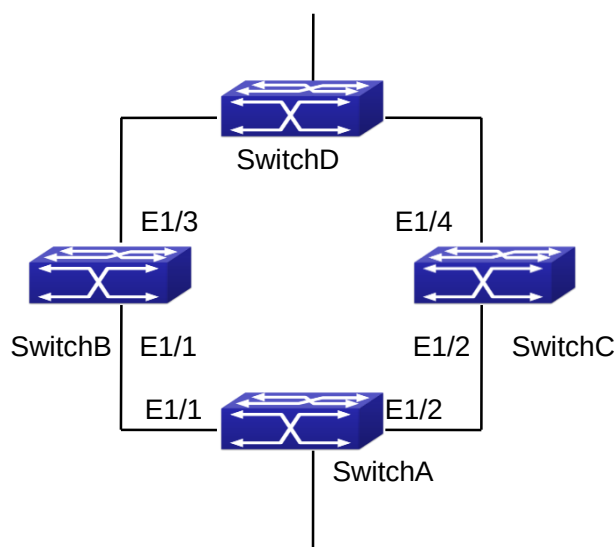


图 7-14 ULSM 典型配置案例

上面的拓扑是 ULSM 和 ULPP 协议配合使用的典型应用环境。

ULSM 本身是用来进行端口状态同步的，该功能单独使用并没有太大的意义，所以一般情况下是和 ULPP 协议结合使用的。在拓扑中，SwitchA 上开启 ULPP 协议，用来进行链路的切换，

SwitchB 和 SwitchC 上开启 ULSM 协议,用来监听上行的链路是否 down,如果 down 掉,则 ULSM 会对下行端口执行 down 操作,从而使下行链路 down 掉,此时 ULPP 协议就相应的执行链路切换操作。

SwitchA 配置任务序列:

```
Switch(Config)#spanning-tree mst configuration
Switch(Config-Mstp-Region)#instance 1 vlan 1
Switch(Config-Mstp-Region)#exit
Switch(Config)#ulpp group 1
Switch(ulpp-group-1)#protect vlan-reference-instance 1
Switch(ulpp-group-1)#exit
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)# ulpp group 1 master
Switch(config-If-Ethernet1/1)#exit
Switch(Config)#interface Ethernet 1/2
Switch(config-If-Ethernet1/2)# ulpp group 1 slave
Switch(config-If-Ethernet1/2)#exit
```

SwitchB 配置任务序列:

```
Switch(Config)#ulsm group 1
Switch(Config)#interface ethernet 1/1
Switch(config-If-Ethernet1/1)#ulsm group 1 downlink
Switch(config-If-Ethernet1/1)#exit
Switch(Config)#interface ethernet 1/3
Switch(config-If-Ethernet1/3)#ulsm group 1 uplink
Switch(config-If-Ethernet1/3)#exit
```

SwitchC 配置任务序列:

```
Switch(Config)#ulsm group 1
Switch(Config)#interface ethernet 1/2
Switch(config-If-Ethernet1/2)#ulsm group 1 downlink
Switch(config-If-Ethernet1/2)#exit
Switch(Config)#interface ethernet 1/4
Switch(config-If-Ethernet1/4)#ulsm group 1 uplink
Switch(config-If-Ethernet1/4)#exit
```

7.6.4 ULSM 排错帮助

- ✎ 如果在正常配置下，下行端口没有响应上行端口的 down 事件，请打开 ULSM 的调试功能，复制 3 分钟内的 debug 信息以及配置信息，并将结果发送给本公司技术服务中心。

7.7 HA

7.7.1 HA 简介

随着网络的发展，用户对于网络的可靠性提出了越来越高的要求，那么向用户提供“高可用性(High Availability)”数据通讯设备，逐渐成为各个数据通信设备商追求的目标，是其保持核心竞争力的重要保证。

HA 是一种系统设计和实现方式，它保证我们的设备能够向用户提供持续的无线和有线的服务。如果因为某种原因，我们提供的网络通信设备“Down 机”，那么这种不可访问性，对某些用户来说可能是无法接受的。

在数据通信中可以使用的 HA 技术很多，针对不同的问题，可能有不同的处理方式。传统高可用技术主要包括：

- 1) 物理设备：冗余电源、冗余风扇、冗余主控、板卡支持热插拔；
- 2) 链路层面：以太网链路聚合（手工聚合、LACP）；
- 3) 二层多路径：STP、MSTP、SmartLink；
- 4) 三层多路径：VRRP、ECMP、动态路由协议多路径；
- 5) 故障 检测：NQA、BFD、OAM、DLDP；
- 6) 不间断转发：GR、热补丁升级；
- 7) L4-L7 多路径：状态热备、非对称路径转发；
- 8) NSF/SSO：不间断转发的主备控制板倒换。

HA（High Availability，高可靠性）特性用于实现系统的高可靠性。支持 HA 特性的设备有两块单板，一块为主用板，工作在 Master 模式；另一块为备用板，工作在 Slave 模式。备用板通过同步功能，来保持与主用板当前配置的一致。

HA 主要体现在以下两个方面：

当主用板发生故障或者被拔出时，备用板将迅速自动取代主用板成为新的主用板（下面简称该过程为主备倒换），以保证设备的继续运行。

当升级设备的时候，可以先升级备用板，再手工执行主备倒换，主备倒换时，原主用板会使用配置的新的下次启动文件重新启动，从而达到升级原主用板的效果。相比整机重启的方式，大大减少了升级过程业务中断的时间。

当使用主备倒换方式升级设备的时候，要求新的软件版本和当前的软件版本能够实现平滑升级（即版本间的差异较小），否则，可能会导致备用板不能启动。这种升级方式的步骤如下：

- (1) 将最新的软件版本分别下载到主用板和备用板；

- (2) 分别指定主用板和备用板的下次启动文件为新版本的软件；
- (3) 手工重启备用板；
- (4) 手动进行主备倒换。

用户不能直接对备用板进行命令行操作，需要通过主用板的命令行界面进行配置，然后主用板会将配置同步给备用板。当备用板启动时，主用板会对备用板进行初始同步（即批量备份）。在这个过程中，系统将不会响应用户的输入。初始同步完成后，用户可以在主用板上正常执行命令、进行配置，此时，主用板和备用板将保持实时同步，即用户对主用板的配置将同步到备用板上，以保证主备板当前配置的一致。

7.7.2 HA 配置任务序列

1. 配置同步 running-config
2. 手动进行主备切换
3. 配置指定板卡复位
4. 配置 running-config 同步间隔时间

1. 配置同步 running-config

命令	解释
全局配置模式	
force runcfg-sync	手动配置主控与备份主控的配置同步。

2. 手动进行主备切换

命令	解释
全局配置模式	
force switchover	手动进行主备切换。

3. 配置指定板卡复位

命令	解释
全局配置模式	
reset slot <slot no>	复位指定板卡（包括备份主控）。

4. 配置 running-config 同步间隔时间

命令	解释
全局配置模式	
runcfg-sync [<interval>]	配置 running-config 同步间隔时间。

7.7.3 HA 典型案例

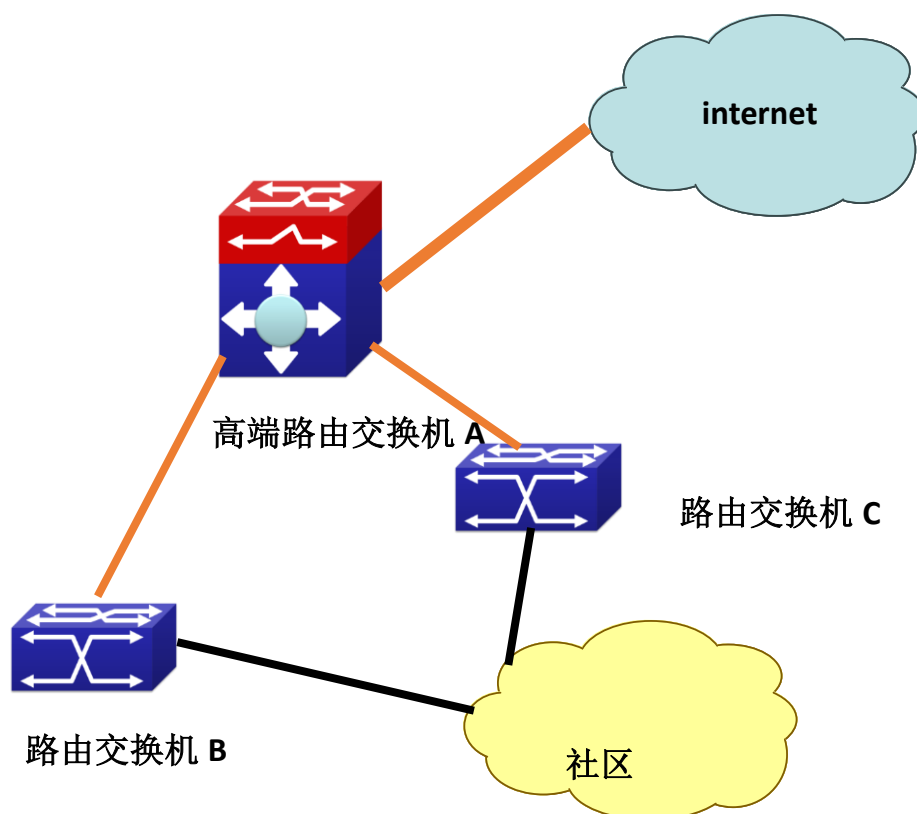


图 7-15 HA 典型应用

一种企业网的典型拓扑如图所示，高端路由交换机 A，通过两台普通路由交换机 B、C 向企业提供 internet 访问服务，尽管其提供 B 和 C 路由交换机作为负载分担和冗余备份链路，当其某一台发生故障的时候，企业依然可以正常的访问网络。但是当 A 节点发生故障的时候，整个企业与 internet 分离，无法提供 internet 访问服务。那么对于 A 设备，当其发生故障的时候，如何支持 NSF/SSO 技术，那么当高端路由交换机 A 的主控板发生故障的时候，其可以在不间断转发的前提下，倒换到备用主控，然后从故障中热恢复过来。在故障恢复的过程中整个数据转发层面没有任何的动荡，而企业内部用户根本无法感知到 A 节点的故障。

当然，在实际的部署中可以采用链路冗余等手段，向用户提供可靠的 internet 访问，但是其他技术都需要其他的冗余链路或者冗余设备，设置需要多态冗余网络设备。相比之下，支持 NSF/SSO 的设备，一台就可以满足用户的需求，这不失为一种选择。

第8章 维护及调试操作

8.1 维护和调试

当用户配置交换机时，需要查看各项配置是否配置正确，交换机是否符合期望，工作正常；或者当网络出现了故障，用户需要诊断故障，交换机为此提供了 ping、telnet、show、debug 等多种调试命令，帮助用户查看系统配置、运行状态，找到故障原因。

8.1.1 Ping

Ping 命令主要用于交换机向远端设备发 ICMP 请求包，检测交换机与远端设备之间是否可达。Ping 命令各选项及参数意义请参考命令手册 Ping 命令章节。

8.1.2 Ping6

Ping6 命令主要用于交换机向远端设备发 ICMPv6 请求包，检测交换机与远端设备之间是否可达。Ping6 命令各选项及参数意义请参考命令手册 Ping6 命令章节。

8.1.3 Traceroute

Traceroute 命令用于测试数据包从发送设备到目的地设备所经过的网关，检测网络是否可达，定位网络故障所在。

Traceroute 命令的执行过程是：首先向目的地址发送一个 TTL 为 1 的数据包，如果第一跳发送回一个 ICMP 错误消息以指明该数据包不能被发送（因为 TTL 超时），则继续向该地址发送 TTL 为 2 的数据包，同样第二跳也可能返回 TTL 超时，于是这个过程不断进行，直到数据包到达目的地。执行这些过程的目的是记录每一个 ICMP TTL 超时消息的源地址，以提供一个 IP 数据包到达目的地所经历的路径。

Traceroute 命令各选项及参数意义请参考命令手册 traceroute 命令章节。

8.1.4 Traceroute6

Traceroute6 功能用于测试数据包从发送设备到目的设备所经过的网关，检测网络是否可达，定位网络故障所在。IPv6 下 Traceroute6 的原理和 IPv4 下的 Traceroute 原理上是一样的，它利用 ICMPv6 及 IPv6 header 的跳限制字段。首先，Traceroute6 送出一个 HOPLIMIT 是 1 的 IPv6 datagram（包括源地址，目的地址和包发出的时间标签）到目的地，当路径上的第一个路由器（router）收到这个 datagram 时，它将 HOPLIMIT 减 1。此时，HOPLIMIT 变为 0 了，所以该路由器会将此 datagram 丢掉，并送回一个「ICMPv6 time exceeded」

消息(包括发 IPv6 包的源地址, IPv6 包的所有内容及路由器的 IPv6 地址), Traceroute6 收到这个消息后, 便知道这个路由器存在于这个路径上, 接着 Traceroute6 再送出另一个 HOPLIMIT 是 2 的 datagram, 发现第 2 个路由器……, Traceroute6 每次将送出的 datagram 的 HOPLIMIT 加 1 来发现另一个路由器, 这个重复的动作一直持续到某个 datagram 抵达目的地。

Traceroute6 命令各选项及参数意义请参考命令手册 traceroute6 命令章节。

8.1.5 Show

show 命令用来显示交换机的系统信息、端口信息、协议运行情况等。本部分介绍交换机的显示系统信息的 show 命令, 其它的 show 命令会在相关章节有介绍。

命令	解释
特权用户配置模式	
show debugging	显示调试开关的状态。
show flash	显示保存在 flash 中的文件及大小。
show history	显示用户最近输入的历史命令。
show history all-users [detail]	显示所有用户最近输入的历史命令, 可以使用 clear history all-users 命令清除系统保存的所有用户命令记录, 系统保存的最大命令记录数可以通过 history all-users max-length 命令设置。
show memory	显示指定内存区域的内容。
show running-config	显示当前运行状态下生效的交换机参数配置。
show running-config current-mode	显示当前模式下的配置。
show startup-config	显示当前运行状态下写在 Flash Memory 中的交换机参数配置, 通常也是交换机下次上电启动时所用的配置文件。
show switchport interface [ethernet <interface-list>]	显示交换机端口的 VLAN 端口模式和所属 VLAN 号及交换机的 Trunk 端口信息。
show tcp show tcp ipv6	显示当前与交换机建立的 TCP 连接情况。
show udp show udp ipv6	显示当前与交换机建立的 UDP 连接情况。
show telnet login	显示当前与交换机建立起 Telnet 连接的 Telnet 客户端的信息。

show tech-support	显示交换机运行的信息和各任务的状态，技术支持人员利用该命令，诊断交换机的运行是否正常。
show version	显示交换机版本信息。
show temperature	本型号交换机只显示主控的温度，不显示其他卡的温度。
show fan	本型号交换机不支持该命令。

8.1.6 Debug

交换机支持的每一项协议都有相应的 **debug** 命令，用户可以通过查看 **debug** 命令的显示信息诊断网络故障。在以后的章节中会陆续介绍到相应协议的 **debug** 命令。

8.2 Info-center

8.2.1 基本概念

信息中心是输出系统信息（以下称之为信息源）的一套机制。

用户需要针对输出方向进行配置，并设置对应的输出方向使能和匹配信息条件就可以输出期望的日志信息，配置过程简单灵活。

8.2.1.1 信息源

信息源分为八个等级（severity）如表：

等级编号	等级名称	说明
1	emergencies	致命错误
2	alerts	需立即纠正的错误
3	critical	关键错误
4	errors	需关注但不关键的错误
5	warnings	警告，可能存在某种差错
6	notifications	需注意的信息
7	informational	一般提示信息
8	debugging	调试信息

8.2.1.2 输出方向

目前支持 16 个输出方向，如表 1-2：

表 1-2

输出方向	说明
console	控制台
monitor	Telnet 终端
loghost <1-8>	日志主机
trapbuffer	告警缓冲
logbuffer	日志缓冲
logfile <1-4>	日志文件

几个输出方向的设置相互独立，但首先需要开启信息中心，设置才会生效。

8.2.1.3 信息源信息的格式

信息源信息的格式如下：

```
<priority>timestamp sysname module/level/digest:content
```

对应的中文格式为：

```
<优先级>时间戳 主机名 模块名/信息级别/信息摘要:信息文本
```

以上格式中的尖括号（< >）、空格、斜杠（/）、冒号（:）在配置了日志前缀（日志前缀可以参考后续的命令说明）时是必须的；其中尖括号只在输出方向为日志主机时有效。

输出到日志主机的日志格式的例子如下：

```
<188>Sep 28 15:33:46:235 2005 MyDevice SHELL/5/LOGIN: Console login from con0
```

不带前缀的日志格式的例子如下：

```
Console login from con0
```

下面对每一个字段做详细说明。

1) 优先级

优先级的计算按如下公式：**facility*8+severity-1**。

facility（工具名称）可选为**local0~local7**，默认为**local7**。值为**16~23**。

severity（信息级别）的取值范围为**1~8**，信息级别具体含义请参见表1-1。

优先级与时间戳之间没有任何字符。优先级只有在信息发送到日志主机上时才有效。

值得一提的是**facility**的由来。**facility**是信息向日志主机输出特有的一个属性，对于其他输出方向无意义。其值**local0~local7**来源于早期**syslog**协议为**UNIX**日志主机定义的日志类型。全部类型如下：

表1-3

facility	code
kernel messages	0

user-level messages	1
mail system	2
system daemons	3
security/authorization messages (note 1)	4
messages generated internally by syslogd	5
line printer subsystem	6
network news subsystem	7
UUCP subsystem	8
clock daemon (note 2)	9
security/authorization messages (note 1)	10
FTP daemon	11
NTP subsystem	12
log audit (note 1)	13
log alert (note 1)	14
clock daemon (note 2)	15
local use 0 (local0)	16
local use 1 (local1)	17
local use 2 (local2)	18
local use 3 (local3)	19
local use 4 (local4)	20
local use 5 (local5)	21
local use 6 (local6)	22
local use 7 (local7)	23

可以看到，为用户预留的就只有local0~local7，值为16~23。

2) 时间戳

时间戳记录了系统信息产生的时间，方便用户察看和定位系统事件。时间戳与主机名之间以一个空格隔开。

date 模式

mmm + space + dd + space + hh + colon + mm + colon + ss + colon + sss+
space+ yyyy

mmm = Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec

表示月份

dd = 1~31，表示每月第几天，其中如果值为 1~9 时，要在前面加 0 后加入 information text，比如 1 要转化为“01”，下同。

hh = 0 ~ 23, 表示小时
mm = 0 ~ 59 表示分钟
ss = 0 ~ 59 表示秒
sss = 0 ~ 999 表示毫秒
yyyy = 1970 ~ 9999 表示年

3) 主机名

主机名是本机的系统名。
主机名与模块名之间以一个空格隔开。

4) 模块名

该字段表示产生信息的功能模块的名称。
模块名与信息级别之间以一个斜杠 (/) 隔开。

5) 信息级别 (severity)

系统信息分为三类：日志、调试和告警。按照紧急程度的不同，每类信息都分为八个等级, 详见章节1.2信息源。系统信息的信息级别值越小，紧急程度越高。

系统进行信息过滤时采用的规则是：禁止信息级别大于所设置阈值的信息输出。

当设置信息级别的取值为 1 时，系统只会输出emergencies信息；

当设置信息级别的取值为 8 时，系统将输出所有级别的信息。

信息级别与信息摘要之间以一个斜杠 (/) 隔开。

6) 信息摘要

信息摘要是一个不超过 32 个字符的字符串，表示该信息的内容大意。

信息摘要与信息文本之间以一个冒号 (:) 隔开。

7) 信息文本

信息文本表示了该条系统信息的具体内容。

8.2.2 信息中心功能配置任务序列

- 1. 设置信息中心全局使能
- 2. 设置信息中心输出方向使能
- 3. 设置日志信息前缀开关
- 4. 设置日志信息匹配条件
- 5. 配置日志主机的IP和等级
- 6. 配置日志文件的条数和存取路径
- 7. 配置记录用户操作命令
- 8. 删除logbuffer或trapbuffer记录的日志
- 9. 查看支持存放文件的区域
- 10. 配置一键收集

1. 设置信息中心全局使能

命令	描述
全局配置模式	

info-center enable	设置信息中心全局使能
no info-center enable	设置信息中心全局去使能

2. 设置信息中心输出方向使能

命令	描述
全局配置模式	
info-center (console logbuffer monitor trapbuffer) output-enable info-center (logfile <1-4> loghost <1-8>) output-enable (member <1-4>)	设置信息中心输出方向使能
no info-center (console logbuffer monitor trapbuffer) output-enable no info-center (logfile <1-4> loghost <1-8>) output-enable (member <1-4>)	设置信息中心输出方向去使能

3. 设置日志信息前缀开关

命令	描述
全局配置模式	
info-center (console logbuffer monitor trapbuffer) prefix (on off) info-center (logfile <1-4> loghost <1-8>) prefix (on off) (member <1-4>)	设置日志信息携带前缀开关

4. 设置日志信息匹配条件

命令	描述
全局配置模式	
info-center (console logbuffer monitor trapbuffer) match level (emergencies alerts critical errors warnings notifications informational debugging) (exact) (keyword WORD) info-center (logfile <1-4> loghost <1-8>) match level (emergencies alerts critical errors warnings notifications informational debugging) (exact) (keyword WORD) (member <1-4>)	设置日志信息匹配条件
no info-center (console logbuffer monitor trapbuffer) match no info-center (logfile <1-4> loghost <1-8>) match	删除日志信息匹配条件

(member <1-4>)	

5. 配置日志主机的 IP 和等级

命令	描述
全局配置模式	
info-center loghost <1-8> config (A.B.C.D X:X::X:X) facility (local0 local1 local2 local3 local4 local5 local6 local7) (member <1-16>)	配置日志主机的 IP 和等级
no info-center loghost <1-8> config (member <1-4>)	删除日志主机的 IP 和等级

6. 配置日志文件的条数和存取路径

命令	描述
全局配置模式	
info-center logfile <1-4> config count <1-40960> (flash usb nandflash) WORD (member <1-4>)	配置日志文件的条数和存取路径
no info-center logfile <1-4> config (member <1-4>)	删除日志文件的条数和存取路径

7. 配置记录用户操作命令

命令	描述
全局配置模式	
info-center (logbuffer logfile <1-4> loghost <1-8>) record-cmd	配置记录用户操作命令
no info-center (logbuffer logfile <1-4> loghost <1-8>) record-cmd	删除记录用户操作命令

8. 删除 logbuffer 或 trapbuffer 记录的日志

命令	描述
全局配置模式	
info-center clear trapbuffer	删除 trapbuffer 记录的日志
info-center clear logbuffer (member <1-4>)	删除 logbuffer 记录的日志

9. 查看支持存放文件的区域

命令	描述
全局配置模式	

info-center list all disk	查看支持存放文件的区域

10. 配置一键收集

命令	描述
全局配置模式	
info-center save all (((flash usb nandflash) WORD))	配置一键收集

8.2.3 信息中心配置举例

8.2.3.1 console 输出日志

举例 1：配置 console 允许输出带前缀的级别为 debugging 及以上的日志。

配置步骤如下：

```
Switch(config)#info-center console output-enable
```

```
Switch(config)#info-center console prefix on
```

```
Switch(config)#info-center console match level debugging
```

8.2.3.2 monitor 输出日志

举例 2：配置 monitor 允许输出不带前缀的级别为 warnings 的日志。

配置步骤如下：

```
Switch(config)#info-center monitor output-enable
```

```
Switch(config)#info-center monitor prefix off
```

```
Switch(config)#info-center monitor match level warnings exact
```

8.2.3.3 logbuffer 输出日志

举例 3：配置 logbuffer 允许输出带前缀的级别为 debugging 及以上并且匹配正则表达式 DEVSM 的日志。

配置步骤如下：

```
Switch(config)#info-center logbuffer output-enable
```

```
Switch(config)#info-center logbuffer prefix on
```

```
Switch(config)#info-center logbuffer match level debugging keyword DEVSM
```

8.2.3.4 trapbuffer 输出日志

举例 4：配置 trapbuffer 允许输出带前缀的级别为 warnings 并且匹配正则表达式 DEVSM 的日志。

配置步骤如下：

```
Switch(config)#info-center trapbuffer output-enable
```

```
Switch(config)#info-center trapbuffer prefix on
```

```
Switch(config)#info-center trapbuffer match level warnings exact keyword DEVSM
```

8.2.3.5 loghost 输出日志

举例 5：配置 member 1 上的 loghost 1 允许输出带前缀的级别为 debugging 及以上的日志，并且 loghost 1 的 IP 为 192.168.1.11，facility 为 local0。

配置步骤如下：

```
Switch(config)#info-center loghost 1 output-enable member 1
```

```
Switch(config)#info-center loghost 1 prefix on member 1
```

```
Switch(config)#info-center loghost 1 match level debugging member 1
```

```
Switch(config)#info-center loghost 1 config 192.168.1.11 facility local0 member 1
```

8.2.3.6 logfile 输出日志

举例 6：配置 member 1 上的 logfile 1 允许输出带前缀的级别为 debugging 及以上的日志，并且 logfile 1 存储路径为 /mnt/flash/file1.log，最大支持 2000 条日志。

配置步骤如下：

```
Switch(config)#info-center logfile 1 output-enable member 1
```

```
Switch(config)#info-center logfile 1 prefix on member 1
```

```
Switch(config)#info-center logfile 1 match level debugging member 1
```

```
Switch(config)#info-center logfile 1 config count 2000 flash file1.log member 1
```

8.2.3.7 记录用户操作命令

举例 7：配置 logbuffer 允许记录用户操作命令。

配置步骤如下：

```
Switch(config)#info-center logbuffer output-enable
```

```
Switch(config)#info-center logbuffer record-cmd
```

8.2.3.8 删除 logbuffer 或 trapbuffer 记录的日志

举例 8：删除 logbuffer 或 trapbuffer 记录的日志。

配置步骤如下：

```
Switch(config)#info-center clear logbuffer member 1
```

```
Switch(config)#info-center clear trapbuffer
```

8.2.3.9 查看支持存放文件的区域

举例 9：查看支持存放文件的区域。

配置步骤如下：

```
Switch(config)#info-center list all disk
```

flash:

usb:

nandflash:

8.2.3.10 一键收集

举例 10：配置一键收集。

配置步骤如下：

```
Switch(config)#info-center list all disk
```

```
*****Now saving Master card(member 1)*****
```

```
Now saving infocenter all configuration, please wait..
```

```
Now saving infocenter logbuffer content, please wait..
```

```
Now saving infocenter trapbuffer content, please wait..
```

```
*****Master card(member 1) saving finished!*****
```

```
*****Now saving Slave card(member 2)*****
```

```
Now saving infocenter logbuffer content, please wait..
```

```
*****Slave card(member 2) saving finished!*****
```

8.2.4 信息中心功能排错帮助

- 配置日志信息的匹配级别对于之后产生的日志有效，之前已经保存的日志信息不变。
- disk 的容量是有限的，当 disk 中的日志达到配置的最大条数时，新的日志信息会替换旧的日志。
- 信息中心全局使能是默认开启的，使用 show info-center config 命令可以显示信息中心当前所有配置。

8.3 定时重启交换机

8.3.1 定时重启交换机简介

定时重启交换机是在不关闭电源的情况下，经过指定的时间以后，重新启动交换机。本功能一般用于版本升级。当升级完版本以后，可以不立刻重启交换机，而是指定经过一段时间以后再重启交换机。

8.3.2 定时重启交换机任务序列

1. 定时热启动交换机

命令	解释
特权用户配置模式	
reload after {[<HH:MM:SS>] [days <days>]}	定时热启动交换机。
reload cancel	取消定时热启动交换机。

8.4 CPU 收发报文调试

8.4.1 CPU 收发报文简介

CPU 收发报文调试命令用于对各种送往交换机 cpu 的报文的诊断调试，这部分命令的使用必须在厂商技术人员的指导下使用。

8.4.2 CPU 收发报文任务序列

命令	解释
全局配置模式	
cpu-rx-ratelimit total <packets> no cpu-rx-ratelimit total	设置 cpu 接收报文总限速值，本命令的 no 操作为恢复 cpu 接收报文总限速为默认值。
cpu-rx-ratelimit protocol {WORD} <1-2000> [slot <slotid>] no cpu-rx-ratelimit protocol {WORD} [slot <slotid>]	设置所有或指定 slot 的协议报文每秒允许送 CPU 的个数，本命令的 no 操作为恢复协议报文每秒允许送 CPU 的个数为默认值。

clear cpu-rx-stat protocol [<protocol-type> all] [slot <slotid>]	将所有或指定类型协议报文的收包统计清 0。
特权模式	
show cpu-rx protocol [<protocol-type> all] [slot <slotid>]	显示所有或指定板卡的指定类型报文的接收信息。
debug driver {receive send} [interface {<interface-name> all}] [protocol {<protocol-type> discard all}] [detail]	打开指定端口指定类型报文驱动接收或发送信息的显示。
no debug driver {receive send}	关闭驱动收发包信息的显示。

命令	解释
特权模式	
protocol filter {protocol-type} no Protocol filter {protocol-type}	开启/关闭针对具名协议报文的处理,目前该具名协议包括 {arp bgp dhcp dhcpv6 hsrp http igmp ip ldp mpls ospf pim rip snmp telnet vrrp}

8.5 COPP

8.5.1 COPP 功能简介

CPU 是整个设备的大脑，负责处理所有控制层面的信息，因此，需要采取适当的措施，对设备 CPU 提供保护，避免网络攻击。CPU 限速保留了之前的协议限速功能，而且 CPU 协议限速改为硬件限速。新增了 COPP 限速功能，COPP（control plane policing）功能可以满足对控制和管理层面的保护，确保路由功能的稳定性和报文的正常传输。首先配置 ACL 规则，支持多种 ACL 规则，通过 ACL 规则对需要限速或过滤的特定报文进行分类，然后通过配置 COPP 策略表匹配对应的 ACL 规则来过滤特定报文或限制报文上 cpu 的速率。COPP 策略表支持单桶和双桶模式以及多种操作动作。

8.5.2 COPP 功能任务序列

1. 配置 ACL 规则，以数字标准 IP 访问列表为例
2. 建立分类表
3. 建立 copp 策略表
4. 配置策略
5. 应用到端口

1. 配置ACL规则

命令	解释
全局配置模式	
access-list <num> {deny permit} {{<slpAddr> <sMask>} any-source {host-source <slpAddr>}} no access-list <num>	创建一条数字标准 IP 访问列表，如果已有此访问列表，则增加一条规则（rule）表项；本命令的 no 操作为删除一条数字标准 IP 访问列表。

2. 建立分类表

命令	解释
全局配置模式	
class-map <class-map-name> no class-map <class-map-name>	建立一个 class-map（分类表），并进入 class-map 模式；本命令的 no 操作为删除指定的 class-map。
match {access-group <acl-index-or-name> ip dscp <dscp-list> ip precedence <ip-precedence-list> ipv6 access-group <acl-index-or-name> ipv6 dscp <dscp-list> ipv6 flowlabel <flowlabel-list> vlan <vlan-list> cos <cos-list> vlan range <vlan-list>} no match {access-group ip dscp ip precedence ipv6 access-group ipv6 dscp ipv6 flowlabel vlan cos vlan range}	设置分类表中的匹配标准，可以根据 ACL 对数据流进行分类；本命令的 no 操作为删除指定的匹配标准。

3. 配置策略表

命令	解释
全局配置模式	
copp-policy-map <policy-map name>	建立一个 copp-policy-map（策略表），并进

no policy-map <policy-map-name>	入 copp-policy-map（策略表）模式；本命令的 no 操作为删除指定的 copp-policy-map。
class <class-map-name> no class <class-map-name>	建立一个 copp-policy-map（策略表）之后，可以将其对应于一个 class（策略分类表），并进入策略分类表模式，之后就可以对不同的数据流采取不同的策略；本命令的 no 操作为删除指定策略分类表。
pps 限速模式： policy packets-per-second <pps> normal-burst-packets <pps> { conform-action exceed-action } <ACTION> no policy bps 限速模式： 1、单桶模式： Policy <bits_per_second> <normal_burst_bytes> {{action{{policied-cos-to-cos-transmit{ policied-cos-to-dscp-transmit violate-a ction}}policied-cos-to-dscp-transmit{p olicied-cos-to-cos-transmit violate-acti on }} policied-dscp-exp-to-cos-transmit{poli cied-dscp-exp-to-dscp-transmit violate -action}}policied-dscp-exp-to-dscp-tra nsmit{policied-dscp-exp-to-cos-trans mit violate-action }}violate-action {drop transmit}} exceed-action ACTION }) 2、双桶模式： policy <bits_per_second> <normal_burst_bytes> [pir <peak_rate_bps>] <maximum_burst_bytes> [[action{{policied-cos-to-cos-transmit{ policied-cos-to-dscp-transmit violate-a ction}}policied-cos-to-dscp-transmit{p	pps限速模式： 支持单桶双色 Policy 命令，限速为 pps 模式，可以根据配置将报文分为不同颜色的报文，并且给不同颜色报文设置相应动作。no 命令删除模式配置。 bps限速模式： 为分类后的流量配置一个策略，支持三色 color 的非聚合 Policy 命令，根据配置参数解析出令牌桶的工作模式，是单速单桶、单速双桶还是双速双桶，并且给不同颜色报文设置相应动作。no 命令删除模式配置。

olicied-cos-to-cos-transmit violate-action } } policied-dscp-exp-to-cos-transmit{policied-dscp-exp-to-dscp-transmit violate-action}}policied-dscp-exp-to-dscp-transmit{policied-dscp-exp-to-cos-transmit violate-action } } exceed-action violate-action ACTION } } ACTION 定义: drop transmit set-internal-priority <intp_value> policied-intp-transmit no policy	
策略分类表配置模式	
drop no drop transmit no transmit	对分类后的流量可以选择丢弃、通过动作；本命令的 no 操作为取消分配的动作。

4. 应用策略表到端口

命令	解释
端口配置模式	
service-policy output <policy-map name> no service-policy output <policy-map-name>	在交换机端口出方向上应用一个策略表；本命令的 no 操作为删除应用到交换机端口的某个指定策略表。

8.5.3 COPP 功能配置举例

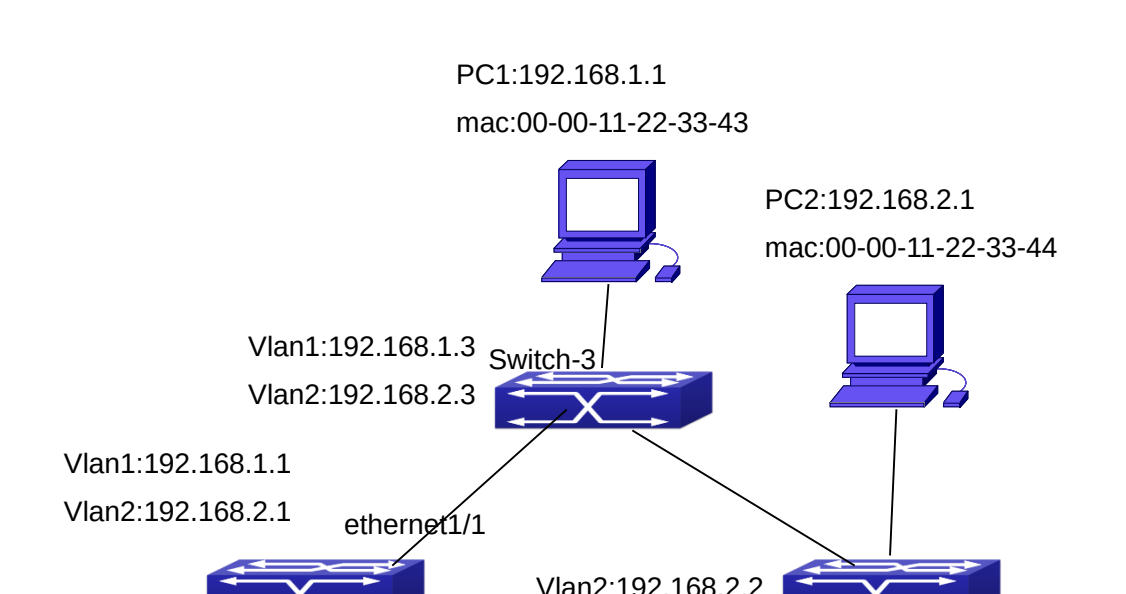


图 8-1 COPP 功能示意图

举例 1: 在 Switch-1 端口 ethernet1/1 上配置 COPP 策略表，将属于网段 192.168.1.0 内的报文上 cpu 的报文限制为 10pps，突发值设为 20pps，超过带宽的该网段内的报文一律丢弃。

配置步骤如下：

```
Switch#config
Switch(config)#access-list 1 permit 192.168.1.0 0.0.0.255
Switch(config)#class-map c1
Switch(config-classmap-c1)#match access-group 1
Switch(config-classmap)# exit
Switch(config)#copp-policy-map p1
Switch(config-copp-policymap-p1)#class c1
Switch(config-copp-policymap-p1-class-c1)#policy packets-per-second 10
normal-burst-packets 20 exceed-action drop
Switch(config-copp-policymap-p1-class-c1)#exit
Switch(config-copp-policymap-p1)# #exit
Switch(config)#interface ethernet 1/1
Switch(config-if-ethernet1/1)# service-policy output p1
```

举例 2: 在 Switch-1 端口 ethernet1/1 上配置 COPP 策略表，将属于源 mac 地址为 00-00-11-22-33-44 的报文上 cpu 的报文限制为 10pps，突发值设为 20pps，超过带宽的该源 mac 地址的报文设置内部优先级为 1 后转发。

配置步骤如下：

```
Switch#config
Switch(config)#access-list 1100 permit host-source-mac 00-00-11-22-33-44
any-destination-mac
Switch(config)#class-map c1
Switch(config-classmap-c1)#match access-group 1100
Switch(config)#copp-policy-map p1
Switch(config-copp-policymap-p1)#class c1
Switch(config-copp-policymap-p1-class-c1)# policy packets-per-second 10
```

```
normal-burst-packets 20 exceed-action set-internal-priority 1 transmit
Switch(config-copp-policymap-p1-class-c1)#exit
Switch(config-copp-policymap-p1)# #exit
Switch(config)#interface ethernet 1/1
Switch(config-if-ethernet1/1)# service-policy output p1
```

8.5.4 COPP 功能排错帮助

- ☞ 属于该网段的报文是否正确上了 cpu，该功能是对上 cpu 的报文限速，可以通过 debug driver receive 查看报文是否上了 cpu。
- ☞ ACL 匹配是否正确，ACL 规则必须是 permit。
- ☞ COPP 策略表是否配置正确，COPP 在端口应用必须是出方向，不支持入方向。
- ☞ 可以通过 show cpu-rx protocol all 命令查看报文统计，确定限速是否生效。

8.6 镜像

8.6.1 镜像介绍

镜像功能包括端口镜像功能，CPU 镜像功能，流镜像功能。

端口镜像功能是指交换机把某一个端口接收或发送的数据帧完全相同的复制给另一个端口；其中被复制的端口称为镜像源端口，复制的端口称为镜像目的端口。通常在镜像目的端口处连接一个协议分析仪（如 Sniffer）或者 RMON 监测仪，可以监视和管理网络，并且能诊断网络故障。

CPU 镜像功能是指交换机把 CPU 接收或发送的数据帧完全复制给一个端口。

流镜像功能是指交换机把端口的指定规则的接收的数据帧完全复制给一个端口，其中指定规则只有为 permit 时流镜像才能生效。

机架式交换机支持最多 4 个镜像目的端口，每个板卡上允许设置一个镜像 session 的源或者目的端口，对于盒式交换机目前能设置多个镜像 session。镜像源端口则没有使用上的限制，可以是 1 个也可以是多个，多个源端口可以在相同的 VLAN，也可以在不同 VLAN。目的端口和源端口可以在不同的 VLAN。

8.6.2 镜像配置任务序列

1. 指定镜像目的端口
2. 指定镜像源端口（CPU）
3. 指定流镜像源

1. 指定镜像目的端口

命令	解释
全局配置模式	
monitor session <session> destination interface <interface-number> no monitor session <session> destination interface <interface-number>	指定镜像目的端口；本命令的 no 操作为删除镜像目的端口。

2. 指定镜像源端口（CPU）

命令	解释
全局配置模式	
monitor session <session> source {interface <interface-list> cpu [slot <slotnum>]} {rx tx both} no monitor session <session> source {interface <interface-list> cpu [slot <slotnum>]}	指定镜像源端口；本命令的 no 操作为删除镜像源端口。

3. 指定流镜像源

命令	解释
全局配置模式	
monitor session <session> source {interface <interface-list>} access-group <num> {rx tx both} no monitor session <session> source {interface <interface-list>} access-group <num>	指定流镜像源端口及应用规则；本命令的 no 操作为删除流镜像源端口。

8.6.3 镜像举例

案例：

用户有如下的配置需求：为了在 1 端口监测 7 端口接收的数据帧和 9 端口发出的数据帧，同时还要监测 CPU 接收和发出的数据帧，端口 15 的符合规则 120（源 IP 为 1.2.3.4,目的 IP 为 5.6.7.8）的入口的数据帧，通过配置镜像来达到这个目的。

配置修改：

- 1: 配置 1 端口为该镜像的镜像目的；
- 2: 配置 7 端口的入口方向和 9 端口的出口方向为镜像源；
- 3: 配置 cpu 为镜像源；

- 4: 配置规则 120;
- 5: 配置端口 15 的入口绑定规则 120。

配置步骤如下:

```
Switch(config)#monitor session 4 destination interface ethernet 1/1
Switch(config)#monitor session 4 source interface ethernet 1/7 rx
Switch(config)#monitor session 4 source interface ethernet 1/9 tx
Switch(config)#monitor session 4 source cpu
Switch(config)#access-list 120 permit tcp 1.2.3.4 0.0.0.255 5.6.7.8 0.0.0.255
Switch(config)#monitor session 4 source interface ethernet 1/15 access-list 120 rx
```

8.6.4 镜像排错帮助

当配置镜像功能出现问题时, 请检查是否是如下原因:

- ☞ 镜像目的端口是端口聚合组成员; 如果是, 请修改端口聚合组。
- ☞ 镜像目的端口的吞吐量小于镜像源端口吞吐量的总和, 目的端口无法完全复制源端口的流量; 请减少源端口的个数或复制单向的流量, 或者选择吞吐量更大的端口作为目的端口。镜像目的端口不能划入Isolate vlan, 否则将影响跨VLAN镜像。

8.7 RSPAN

8.7.1 RSPAN 简介

端口镜像功能是指交换机把某一个端口接收或发送的数据帧完全相同的复制给另一个端口。其中被复制的端口称为镜像源端口, 复制到的端口称为镜像目的端口。镜像功能的出现, 给网管人员诊断网络故障带给了很大的方便。但美中不足的是, 它只局限于镜像源端口和镜像目的端口处于同一交换机的情况。

RSPAN (remote switched port analyzer, 远程交换端口分析), 即远程端口镜像, 突破了镜像源和目的端口必须处于同一交换机的限制, 使镜像源端口和镜像目的端口可以在不同的网络设备上, 从而方便网管人员对远程交换机设备进行管理。在远程镜像的 vlan 上不能进行业务流的传输。

实现了 RSPAN 功能的交换机分为三种:

1. 源交换机: 被监测的端口所在的交换机, 负责将需要镜像的流量在 Remote VLAN 上做二层转发, 转发给中间交换机或目的交换机。
2. 中间交换机: 网络中处于源交换机和目的交换机之间的交换机, 把镜像流量传输给下一个中间交换机或目的交换机。如果源交换机与目的交换机直接相连, 则不存在

中间交换机。

3. 目的交换机：远程镜像目的端口所在的交换机，将从 Remote VLAN 接收到的镜像流量通过镜像目的端口转发给监控设备。

在配置源交换机的 rspan 镜像时可以采用反射端口或直接镜像到目的端口的方式。在目的交换机端将 RSPAN vlan 上的数据发送到 RSPAN 目的端口。我们采用一般模式和高级模式两种，其中一般作为默认方法，适合一般用户。高级模式，适合高级用户。

- 1.高级模式特性：RSPAN vlan 的数据重定向到 RSPAN 目的端口需要 RSPAN 链路的中间和目的设备支持流的重定向的功能。
- 2.一般模式特性：RSPAN 目的端口划入 RSPAN vlan，这样 RSPAN 的数据直接广播到目的端口。目的端口局限于 RSPAN vlan，源端口不能配置广播抑制。需要小心配置各个 Trunk 端口，防止 RSPAN 数据向其他网络转发。由于采用一般方式，占用资源最少、配置简单。

注：默认采用一般模式。采用一般模式时，如果镜像的报文的 MAC 地址是保留地址则无法广播

机架式交换机支持最多 4 个镜像目的端口，每个板卡上允许设置一个镜像 session 的源或者目的端口，对于盒式交换机目前只能设置一个镜像 session。镜像源端口则没有使用上的限制，可以是 1 个也可以是多个，多个源端口可以在相同的 VLAN，也可以在不同 VLAN。目的端口和源端口可以在不同的 VLAN。

在使用 RSPAN 功能之前，要先创建专属的 RSPAN vlan，用于转发 RSPAN 数据报文不能在这个 vlan 上承载业务。由于默认情况下所有的端口都属于缺省 vlan，在标记 RSPAN vlan 时禁止将缺省 vlan、动态 vlan、私有 vlan、组播 vlan、配置了三层接口的 vlan 等特殊 vlan 配置为 RSPAN vlan。目的端口与 Monitor 相连，该端口必须为 access 端口且属于 RSPAN vlan。反射端口必须属于 RSPAN vlan，可以为 RSPAN vlan 的 access 口，或者为 trunk 口。端口被配置为 RSPAN 的反射端口之后自动失去与对端的连接并禁止在反射端口上做 loopback 相关的配置，同时其他的业务数据也将无法转发，所以反射端口要求必须是专用的端口。需要通过配置保证 Remote VLAN 从源交换机到目的交换机的二层互通性。

需要注意的问题：

- 1.在源交换机、中间交换机和目的交换机的 RSPAN vlan 上都不能起三层接口，否则镜像后的报文有可能被丢弃而无法到达目的端口。
- 2.在源交换机和中间交换机的 RSPAN 数据传输链路中，交换机输出方向的 Trunk 端口的 native vlan 不能设置为 RSPAN vlan，否则 RSPAN tag 在没有到达目的交换机时就会被剥离掉，从而导致 RSPAN 失败。
- 3.在源交换机采用反射端口的镜像方式时，不能把镜像的源端口以 access 或者是 trunk 的方式配置为 RSPAN VLAN 的成员端口。采用反射端口做跨板的 cpu tx 方向镜像时，反射端口必须配置为 trunk 端口允许 RSPAN VLAN 的数据通过，并且其 native vlan 不能配置为 RSPAN VLAN。
- 4.在使用远程镜像时需注意链路的带宽要满足业务流和镜像流量的需要。

名词解释：

RSPAN(remote switched port analyzer)：远程交换端口分析。

RSPAN vlan：专用于 RSPAN 的一个特殊 vlan。

RSPAN Tag：RSPAN 数据在 MTP 上被打上的一个 Vlan Tag。

反射端口：从 RSPAN 源端口到本地目的端口的本地镜像。但这个本地目的端口不直接与中间交换机相连，我们把这个端口叫做反射端口。

8.7.2 RSPAN 配置任务序列

1. 配置 rspan vlan
2. 指定镜像源端口（cpu）
3. 指定镜像目的端口
4. 配置反射端口
5. 配置镜像组的 remote vlan

1. 配置 RSPAN vlan

命令	解释
Vlan 配置模式	
remote-span no remote-span	将当前 vlan 配置为 rspan vlan；no 操作为删除 rspan vlan。

2. 指定镜像源端口(CPU)

命令	解释
全局配置模式	
monitor session <session> source {interface <interface-list> cpu [slot <slotnum>]} {rx tx both} no monitor session <session> source {interface <interface-list> cpu [slot <slotnum>]}	指定镜像源端口；本命令的 no 操作为删除镜像源端口。

3. 指定镜像目的端口

命令	解释
全局配置模式	
monitor session <session> destination interface <interface-number> no monitor session <session> destination interface <interface-number>	指定镜像目的端口；本命令的 no 操作为删除镜像目的端口。

4. 配置反射端口

命令	解释
全局配置模式	
monitor session <session> reflector-port <interface-number> no monitor session <session> reflector-port	将端口配置为反射端口，使用 no 命令删除反射端口。

5. 配置镜像组的 remote vlan

命令	解释
全局配置模式	
monitor session <session> remote vlan <vid> no monitor session <session> remote vlan	配置镜像组的 remote vlan，no 操作为删除镜像组的 remote vlan。

8.7.3 RSPAN 典型案例

在没有 RSPAN 之前，网络管理人员需要带着笔记本电脑跑到各个交换机处，将电脑连接到交换机上，才能检测网络的状况，非常不方便。

而有了 RSPAN 功能之后，管理员只需要在网管中心通过 Telnet 做一些简单的配置，就能检测到任何一台交换机（当然，这台交换机需要支持 RSPAN 功能）的运行状况。大大方便了网管人员的管理工作。下图是 RSPAN 功能的一种应用：

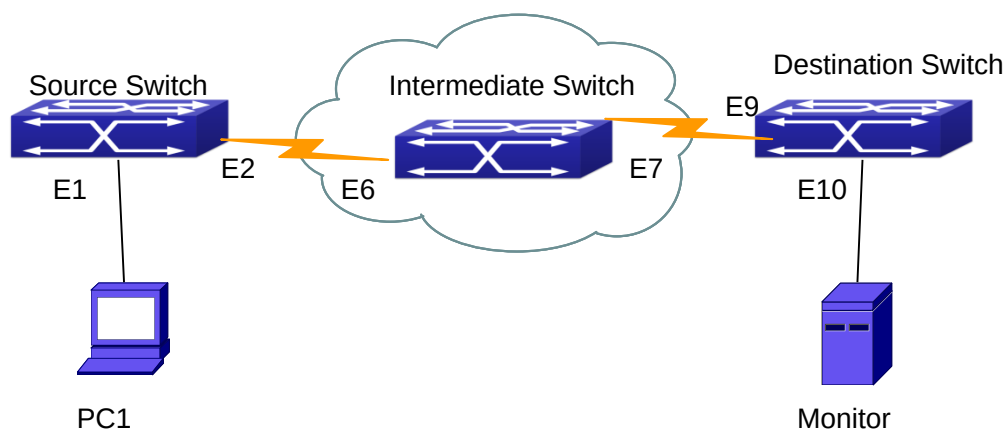


图 8-2 RSPAN 应用示意图

可以采用两种方案：方案一无反射端口，方案二有反射端口：方案一只能固定一个端口与中间交换机连接，方案二源与中间交换机连接端口不固定，比较灵活；

两种方案的比较：方案一虽说只能固定一个接口与中间交换机连接，但是它省去了一个反射端口，端口利用率高。方案二通过反射端口，可以将数据报文通过回环在 RSPAN vlan 中广播，灵活的与中间交换机通信。

以下是采用一般模式转发时各个交换机的配置：

方案一：

源交换机：

interface ethernet 1/1 为源端口。

interface ethernet 1/2 为目的端口，与中间交换机相连。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
```

```
Switch(Config-Vlan5)#remote-span
```

```
Switch(Config-Vlan5)#exit
```

```
Switch(config)#interface ethernet 1/2
```

```
Switch(Config-If-Ethernet1/2)#switchport mode trunk
```

```
Switch(Config-If-Ethernet1/2)#exit
```

```
Switch(config)#monitor session 1 source interface ethernet1/1 rx
```

```
Switch(config)#monitor session 1 destination interface ethernet1/2
```

```
Switch(config)#monitor session 1 remote vlan 5
```

中间交换机：

interface ethernet1/6 为源端口，与源交换机相连。

interface ethernet1/7 为目的端口，与目的交换机相连，该端口的 native vlan 不能设置为 RSPAN vlan，否则镜像后的数据可能无法在目的交换机上正确传输。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
```

```
Switch(Config-Vlan5)#remote-span
```

```
Switch(Config-Vlan5)#exit
```

```
Switch(config)#interface ethernet 1/6-7
```

```
Switch(Config-If-Port-Range)#switchport mode trunk
```

```
Switch(Config-If-Port-Range)#exit
```

目的交换机：

interface ethernet1/9 为源端口，与源交换机相连。

interface ethernet1/10 为目的端口，与 Monitor 相连，该端口必须为 access 端口且属于 RSPAN vlan。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
```

```
Switch(Config-Vlan5)#remote-span
```

```
Switch(Config-Vlan5)#exit
```

```
Switch(config)#interface ethernet 1/9
Switch(Config-If-Ethernet1/9)#switchport mode trunk
Switch(Config-If-Ethernet1/9)#exit
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#switchport access vlan 5
Switch(Config-If-Ethernet1/10)#exit
```

方案二:

源交换机:

interface ethernet 1/1 为源端口。

interface ethernet 1/2 为 trunk 端口, 与中间交换机相连, 该端口的 native vlan 不能为 RSPAN vlan。

interface ethernet 1/3 为反射端口, 也是本地镜像目的端口, 反射端口必须属于 RSPAN vlan, 可以为 RSPAN vlan 的 access 口, 或者为 trunk 口。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
Switch(Config-Vlan5)#remote-span
Switch(Config-Vlan5)#exit
Switch(config)#interface ethernet 1/2
Switch(Config-If-Ethernet1/2)#switchport mode trunk
Switch(Config-If-Ethernet1/2)#exit
Switch(config)#interface ethernet 1/3
Switch(Config-If-Ethernet1/3)#switchport mode trunk
Switch(Config-If-Ethernet1/3)#exit
Switch(config)#monitor session 1 source interface ethernet1/1 rx
Switch(config)#monitor session 1 reflector-port ethernet1/3
Switch(config)#monitor session 1 remote vlan 5
```

中间交换机:

interface ethernet1/6 为源端口, 与源交换机相连。

interface ethernet1/7 为目的端口, 与目的交换机相连, 该端口的 native vlan 不能设置为 RSPAN vlan, 否则镜像后的数据可能无法在目的交换机上正确传输。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
Switch(Config-Vlan5)#remote-span
Switch(Config-Vlan5)#exit
```

```
Switch(config)#interface ethernet 1/6-7
Switch(Config-If-Port-Range)#switchport mode trunk
Switch(Config-If-Port-Range)#exit
```

目的交换机：

interface ethernet1/9 为源端口，与源交换机相连。

interface ethernet1/10 为目的端口，与 Monitor 相连，该端口必须为 access 端口且属于 RSPAN vlan。

RSPAN vlan 为 5。

```
Switch(config)#vlan 5
Switch(Config-Vlan5)#remote-span
Switch(Config-Vlan5)#exit
Switch(config)#interface ethernet 1/9
Switch(Config-If-Ethernet1/9)#switchport mode trunk
Switch(Config-If-Ethernet1/9)#exit
Switch(config)#interface ethernet 1/10
Switch(Config-If-Ethernet1/10)#switchport access vlan 5
Switch(Config-If-Ethernet1/10)#exit
```

8.7.4 RSPAN 排错帮助

当配置 rspan 功能出现问题时，请检查是否是如下原因：

- ☞ 镜像目的端口是否是端口聚合组成员；如果是，请修改端口聚合组；
- ☞ 镜像目的端口的吞吐量小于镜像源端口吞吐量的总和，目的端口无法完全复制源端口的流量；请减少源端口的个数或复制单向的流量，或者选择吞吐量更大的端口作为目的端口。
- ☞ 在源交换机和中间交换机的RSPAN数据传输链路中，交换机输出方向的Trunk端口的 native vlan是否设置为RSPAN vlan；如果是，请修改Trunk端口的native vlan。
- ☞ 配置RSPAN功能后，出口镜像的报文会加上vlan tag，导致反射端口上出现abort错误帧，此时需要修改交换机默认MTU值。

8.8 sFlow

8.8.1 sFlow 介绍

sFlow（RFC 3176）是基于标准网络导出协议，是由 InMon 公司开发出来的用于监控

网络流量信息的协议。其主要操作是由被监视的交换机、路由器把被监控的数据通过采样、统计等操作发送到用于监控的用户端分析器，由分析器对收到的数据进行用户所要求的分析，从而达到监控网络的目的。

一个 sFlow 监控系统包括：sFlow 代理、中央数据收集器、和 sFlow 分析器。sFlow 代理利用采样技术从交换机设备抓取数据。sFlow 数据收集器用来格式化要转发采样的数据统计到 sFlow 分析器，sFlow 分析器对采样数据进行分析并根据分析结果作出相应的处理措施。这里我们的交换机实现的是 sFlow 系统中的代理和中央数据收集器部分。

我们当前实现了针对物理端口的数据采样和统计。

我们实现的采样数据类型针对 IPV4 报文类型、IPv6 报文类型进行处理。其它类型的扩展暂不支持。对于非 IPV4 和 IPv6 的报文，我们按照 RFC3176 的要求，采用统一的头（HEADER）模式，在分析其协议类型的基础上，复制报文的头信息。

InMon 公司发布的 sFlow 协议目前已经更新到了版本 5，由于 RFC3176 实现的是版本 4，而版本之间在结构和报文格式上可能存在一定的不兼容，由于版本 5 还没有成为正式的协议，为与当前的应用软件兼容，我们仍遵照 RFC3176 实现。

8.8.2 sFlow 配置任务序列

1.配置 sFlow Collector 地址

命令	解释
全局配置模式，端口配置模式	
sflow destination <collector-address> [<collector-port>] no sflow destination	配置 sFlow 分析软件所在主机的 IP 地址和端口号。对于端口来说，如果端口上有配置 IP 地址，则使用端口的配置，否则使用全局的配置。No 命令恢复为默认端口值，删除 IP 地址值。

2.配置 sFlow 代理地址

命令	解释
全局配置模式	
sflow agent-address <collector-address> no sflow agent-address	配置 sFlow 代理使用的源 IP 地址，no 命令删除该地址。

3.配置 sFlow 协议优先级

命令	解释
全局配置模式	
sflow priority <priority-value> no sflow priority	配置 sFlow 从硬件收报文时的优先级，no 命令恢复为默认值。

4.配置 sFlow 复制报文数据头长度

命令	解释
端口配置模式	
sflow header-len <length-vlaue> no sflow header-len	配置 sFlow 在数据采样中复制的报文数据头的长度，no 命令恢复为默认值。

5.配置 sFlow 报文最大允许数据头长度

命令	解释
端口配置模式	
sflow data-len <length-vlaue> no sflow data-len	配置 sFlow 在报文中允许的最大数据长度，no 命令恢复为默认值。

6.配置 sFlow 硬件采样速率

命令	解释
端口配置模式	
sflow rate {input <input-rate> output <output-rate> } no sflow rate [input output]	配置 sFlow 在硬件采样时的采样速率。No 命令删除采样速率值。

7.配置 sFlow 统计采样间隔

命令	解释
端口配置模式	
sflow counter-interval <interval-vlaue> no sflow counter-interval	配置 sFlow 进行统计采样的最大间隔。No 命令删除统计采样间隔值。

8. 配置 sFlow 使用的分析器

命令	解释
全局配置模式	
sflow analyzer sflowtrend no sflow analyzer sflowtrend	配置 sFlow 使用的分析器。no 命令删除 sflow 的分析器。

9. 配置全局 sflow 版本

命令	解释
全局配置模式	
sflow version<version> no sflow version	配置全局 sflow 版本，本命令的 no 操作恢复为缺省值。

8.8.3 sFlow 举例

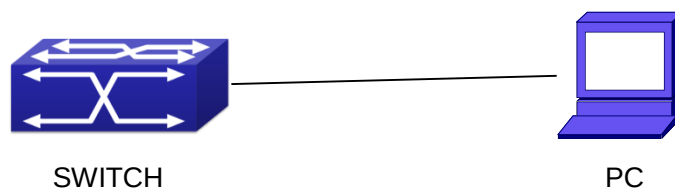


图 8-3 sFlow 配置拓扑图

如图所示, SWITCH 上的端口 1/1 以及 1/2 上启动 sFlow 采样, 假定 PC 上安装了 sFlow 分析器软件, PC 地址为 192.168.1.200, SWITCH 上与 PC 相连的三层接口的地址为 192.168.1.100, SWITCH 上配置了一个地址为 10.1.144.2 的 loopback 接口。以下是 sFlow 的配置

配置步骤如下:

```
switch#config
switch (config)#sflow agent-address 10.1.144.2
switch (config)#sflow destination 192.168.1.200
switch (config)#sflow priority 1
switch (config)#in e1/1
switch (Config-If-Ethernet1/1)#sflow rate input 10000
switch (Config-If-Ethernet1/1)#sflow rate output 10000
switch (Config-If-Ethernet1/1)#sflow counter-interval 20
switch (Config-If-Ethernet1/1)#exit
switch (config)#in e1/2
switch (Config-If-Ethernet1/2)#sflow rate input 20000
switch (Config-If-Ethernet1/2)#sflow rate output 20000
switch (Config-If-Ethernet1/2)#sflow counter-interval 40
```

8.8.4 sFlow 排错帮助

在配置、使用 sFlow 功能时, 可能会由于物理连接、配置错误等原因导致 sFlow 未能正常运行。因此, 用户应注意以下要点:

- ◇ 应该保证物理连接的正确无误;
- ◇ 保证全局配置模式或者接口配置模式下所配置的sFlow分析器地址是可达的;
- ◇ 如果要求流采样, 必须保证配置了接口下的采样速率;
- ◇ 如果要求统计采样, 必须保证配置了接口下的统计采样间隔。
- ◇ 如果使用检查尝试仍无法解决, 请联系本公司技术服务中心。

8.9 IPFIX

8.9.1 IPFIX 介绍

IPFIX 全称为 IP Flow Information Export，即 IP 流信息输出，它是由 IETF 根据 Cisco NetFlow Version9 制定的用于网络中流信息测量的标准协议，它使网络中流量统计信息的格式标准化。其主要操作是由被监控的交换机或者路由器根据用户的监控需求，把被监控的数据流进行分类、统计，生成不同的流记录，然后发送到用于监控的用户端的收集器，由收集器对收集到的流记录进行分析和存储，通过记录和分析网络中这些流量的特征，如流量持续时间、流量中报文平均长度等，我们可以了解到当前网络的应用情况，并根据这些信息对网络进行优化，安全检测，流量计费，从而达到监控网络流量的目的。该协议可以工作在任何厂家的网络设备和管理系统平台之上，而且 IPFIX 的数据输出格式是基于模板的，具有很强的可扩展性，如果流量监控的要求发生改变，网络管理员也不必升级网络设备软件或管理工具。

目前网络界基于流(Flow)的分析技术主要有 NetFlow、sFlow 和 IPFIX。NetFlow 是 Cisco 公司独有的技术，它既是一种流量分析协议，又是一种流交换技术，而 IETF 制定的 IPFIX 规范，则是基于 NetFlow V9 设计的，它使得网络中流量统计信息的格式趋于标准化。sFlow 是基于标准网络导出协议，是由 InMon 公司开发出来的用于监控网络流量信息的协议。它采用数据流随机采样技术把采样数据发送到用于监控的用户端分析器，由分析器对收到的数据进行用户所要求的分析，从而达到监控网络的目的。和 IPFIX 相比，sFlow 更像是一种简单的数据采样，更容易支持高速接口，它能提供给分析器更多的报文信息，但报文输出格式是固定的，不能扩展，sFlow 的实时性较强，具备突出的第 2-7 层信息的描述能力。而 IPFIX 可以通过用户的配置把不同的报文流进行分类统计，主要关注报文的头部信息，提供路由器的三层信息。另外，用户可以灵活的配置想要获取的报文内容，同时也可以设定数据输出的模板格式，具有很强的可扩展性。

基于上面的分析，sFlow 主要考虑用于那些对统计结果要求不是特别准确，用户需要关心报文内容的应用环境或者网络流量比较大的环境中。IPFIX 主要用于需要精确统计和对流量进行分类统计的应用环境中，比如需要对服务类型进行分类统计。

IPFIX 是在支持特定芯片的板卡上实现的，和 sFlow 模块不存在互斥关系。因此，交换机或者路由器可以同时支持这两个功能，用户可以根据实际需要来选择不同的流量统计方法。

8.9.2 IPFIX 基本配置

IPFIX 配置任务序列如下：

1. 配置匹配规则

- 4) 设置 L2 报文流记录匹配关键字
- 5) 设置 IPv4 报文流记录匹配关键字
- 6) 设置 IPv6 报文流记录匹配关键字
- 7) 设置流记录的非关键字
2. 配置采样规则
3. 配置输出规则
4. 配置监控规则
 - 1) 选择匹配关键字
 - 2) 选择输出地址
 - 3) 选择监控报文类型
 - 4) 设置监控相关参数
5. 将配置应用到端口

1. 配置匹配规则

命令	解释
全局配置模式	
ipfix record <name> no ipfix record <name>	创建新 record，并进入 record 配置模式；本命令的 no 操作为删除指定的 record。
match datalink vlan {id priority} no match datalink vlan {id priority} match datalink mac {destination-address source-address} no match datalink mac {destination-address source-address} match datalink ether-type no match datalink ether-type	设置 L2 报文的流记录关键字，当需要多个关键字时，可进行多次配置。 L2 报文关键字有： vlan-id vlan-priority dst-mac-address src-mac-address ether-type
select {ipv4 ipv6} no select {ipv4 ipv6}	选择流记录匹配关键字的类型。（该命令不配置时，match ip/match ipv4-mask/match ipv6-prefix 命令的配置不生效）

<pre>match ip {protocol tos destination-port source-port} no match ip {protocol tos destination-port source-port }</pre>	设置 IP 报文（对 IPv4 及 IPv6 报文均生效）的流记录关键字，当需要多个关键字时，可进行多次配置。 该命令设置的报文关键字有： protocol（对 IPv6 而言，是匹配 next-header 字段） tos destination-port source-port
<pre>match ipv4-mask destination <mask-length> source <mask-length> no match ipv4-mask</pre>	设置匹配 IPv4 报文的源/目的地址掩码长度。（需要与 select ipv4 命令配合使用）
<pre>match ipv6-prefix destination <prefix-length> source <prefix-length> no match ipv6-prefix</pre>	设置匹配 IPv6 报文的源/目的地址前缀长度。（需要与 select ipv6 命令配合使用）
<pre>match ipv6 flow-label no match ipv6 flow-label</pre>	设置 IPv6 报文的流关键字为 flow-label。
<pre>collect counter {bytes packets} no collect counter {bytes packets} collect timestamp sys-uptime{first last} no collect timestamp sys-uptime{first last}</pre>	设置流记录的非关键字，该非关键字用于流信息中提供一些额外信息，但不会产生新流；当需要多个非关键字时，可进行多次配置。 非关键字有： bytes packets sys-uptime first sys-uptime last
<pre>description no description</pre>	配置描述信息。

2. 配置采样规则

命令	解释
全局配置模式	
<pre>ipfix sampler <name> no ipfix sampler <name></pre>	创建新 sampler，并进入 sampler 配置模式；本命令的 no 操作为删除指定的 sampler。
<pre>rate <number> no rate</pre>	设置采样率为 1/（N+1），每隔 N 个报文采一个。（不区分报文类型）

description	配置描述信息。
no description	

3. 配置输出规则

命令	解释
全局配置模式	
ipfix exporter no ipfix exporter	创建新 exporter，并进入 exporter 配置模式；本命令的 no 操作为删除指定的 exporter。
ipv4 destination <ipv4-address> [source <ipv4-address>] no ipv4 destination ipv6 destination <ipv6-address> [source <ipv6-address>] no ipv6 destination	配置流记录输出的目的地址及源地址，每个 exporter 只能配置一个 IPv4 源/目的地址，或者是一个 IPv6 源/目的地址。
transport {udp tcp sctp} [destination-port <port>] no transport	选择传输协议及传输端口，目前仅支持 UDP 协议。
udp template {timeout-rate <seconds> refresh-rate <packets>} no udp template	配置 UDP 协议下模板重传参数，可以选择以时间或者以发送报文数为间隔单位。
description no description	配置描述信息。

4. 配置监控规则

命令	解释
全局配置模式	
ipfix monitor <name> no ipfix monitor <name>	创建新 monitor，进入 monitor 配置模式；本命令的 no 操作为删除指定的 monitor。
record {<name> default-set [ipv4] [ipv6] [l2] [ipv4-ipv6] [ipv4-l2] [ipv6-l2]} no record	选择监控报文的关键字，可以配置使用步骤 1 所创建的 record 或者是预设的基本流关键字。
exporter <name> no exporter <name>	选择流记录输出地址，对应于步骤 3 所创建的 exporter。

set packet-type {ipv4 ipv6 l2} no set packet-type {ipv4 ipv6 l2}	配置需要监控的报文类型，若需要监控多种报文，可进行多次配置。
deal {non-discard discard all} no deal	配置是否需要监控被标记为丢弃的报文。
cache {entries <entries> timeout {active <active- time> inactive <inactive- time>} type {normal tcp-end-detect}} no cache {entries <entries> timeout {active <active- time> inactive <inactive- time>} type {normal tcp-end-detect}}	配置流记录的输出参数，其中包括 cache 中最大流记录存储数，记录输出方式，老化时间及活动时间。
description no description	配置描述信息。

5. 将配置应用到端口

命令	解释
特权模式	
ipfix apply monitor <monitor-name> [sampler <sampler-name>] {input output} no ipfix apply monitor <monitor-name> [sampler <sampler-name>] {input output}	将 ipfix monitor 及 sampler 功能应用于端口，对于每个端口，输入和输出方向都只能设置一个 ipfix monitor。

8.9.3 IPFIX 举例

案例：

IPFIX 的一个应用是基于使用的计费。IPFIX 记录可以精确到目的 IP、协议类型和端口号等字段，它可以为应用报告提供更详细的测量结果。如下图所示，交换机需要在端口 1/1 上监控某个用户在差分服务网络中的单一流量，需要监控如下信息：

- IPv4 源地址: 4 字节
- IPv4 目的地址: 4 字节
- TOS(DSCP+ECN): 1 个字节

还需要报告通信量结果，这样，要使用流字节数（4 字节）作为输出记录的非关键字。最后将该记录信息输出到 server 服务器。

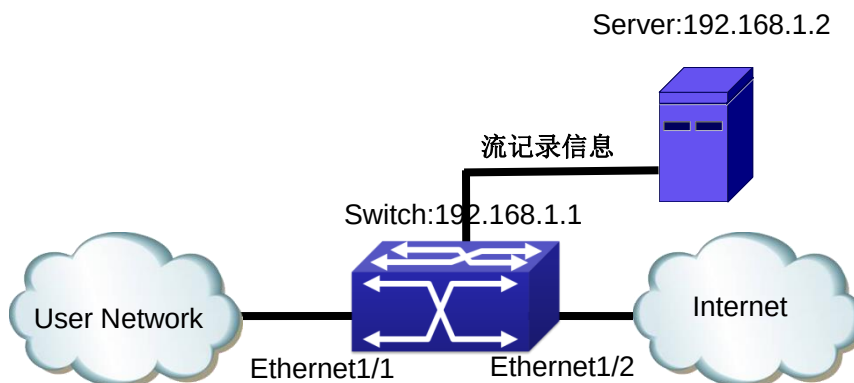


图 8-4 IPFIX 配置

为实现此应用，需按以下方式进行配置：

Switch 的配置(此时无须进行采样规则的配置)：

(1)匹配规则：匹配报文的 IPv4 源地址、IPv4 目的地址和 TOS 字段，流的字节数作为非关键字。

```
Switch(config)#ipfix record my-record
```

```
Switch(config-ipfix-record)#select ipv4
```

```
Switch(config-ipfix-record)#match ipv4-mask destination 32 source 32
```

```
Switch(config-ipfix-record)#match ip tos
```

```
Switch(config-ipfix-record)#collect counter bytes
```

```
Switch(config-ipfix-record)#exit
```

(2)输出规则：输出目的地址为 192.168.1.2，源地址为 192.168.1.1。

```
Switch(config)#ipfix exporter my-exporter
```

```
Switch(config-ipfix-exporter)#ipv4 destination 192.168.1.2 source 192.168.1.1
```

```
Switch(config-ipfix-exporter)#exit
```

(3)监控规则：监控 ipv4 报文，选择关键字及输出地址为步骤 1 和 2 的配置，其他参数使用默认设置即可。

```
Switch(config)#ipfix monitor my-monitor
```

```
Switch(config-ipfix-monitor)#set packet-type ipv4
```

```
Switch(config-ipfix-monitor)#record my-record
```

```
Switch(config-ipfix-monitor)#exporter my-exporter
```

```
Switch(config-ipfix-monitor)#exit
```

(4)将配置应用到端口 1/1。

```
Switch(config)#interface ethernet1/1
```

```
Switch(config-if-ethernet1/1)#ipfix apply monitor my-monitor input
Switch(config-if-ethernet1/1)#ipfix apply monitor my-monitor onput
Switch(config-if-ethernet1/1)#exit
```

8.9.4 IPFIX 排错帮助

- 当在使用 IPFIX 过程中遇到问题，请检查是否是如下原因：
- ❑ 交换机是否正确配置了监控规则，请确保关键字及监控报文类型配置的正确。
 - ❑ 请确保交换机与输出目的地址之间的连通性以及所使用的流收集工具(或计费软件)支持 IPFIX 功能。
 - ❑ 交换机 IPFIX 功能是否正常，可通过 debug ipfix monitor 及 debug ipfix exporter 观察交换机能否正确处理及发送相关 IPFIX 报文。

8.10 ERSPAN

8.10.1 ERSPAN 简介

ERSPAN(Encapsulated Remote Switched Port Analyzer, 密封式远程交换端口分析)，突破了镜像源和目的端口必须处于同一交换机的限制,使镜像源端口和镜像目的端口可以在不同的网络设备上，从而方便网管人员对远程交换机设备进行管理。与传统的 RSPAN 相比配置更简单，使被镜像到的端口流量在指定隧道中传输。

- 需要注意的问题：
1. ERSPAN 镜像的镜像源只能端口镜像，目前不支持 CPU 镜像和流镜像。
 2. 在源交换机和目的交换机的 ERSPAN 数据传输链路中，必须有一条隧道。
 3. 在使用远程镜像时需注意链路的带宽要满足业务流和镜像流量的需要。

名词解释：
ERSPAN(Encapsulated Remote Switched Port Analyzer): 密封式远程交换端口分析。

8.10.2 ERSPAN 配置任务序列

1. 指定镜像源端口

命令	解释
全局配置模式	

monitor session <session> source {interface <interface-list> {rx tx both}} no monitor session <session> source {interface <interface-list>}	指定镜像源端口；本命令的 no 操作为删除镜像源端口。

2. 指定镜像目的隧道

命令	解释
全局配置模式	
monitor session <session> destination tunnel <tunnel-number> no monitor session <session> destination tunnel <tunnel-number>	指定镜像目的隧道；本命令的 no 操作为删除镜像目的隧道。

8.10.3 ERSPAN 典型案例

在没有 ERSPAN 之前，网络管理人员需要带着笔记本电脑跑到各个交换机处，将电脑连接到交换机上，才能检测网络的状况，非常不方便。

而有了 ERSPAN 功能之后，管理员只需要在源交换机上配置好隧道，并制定镜像目的隧道，中间交换机配置 OSPF，目的交换机按照普通端口镜像模式配置，就可以实现远程端口分析。

以下是一个应用环境举例：在一个三层网络中，Device A 通过端口 Ethernet 1/1 连接市场部，并通过端口 Ethernet1/2 与 Device B 的端口 Ethernet1/2 相连；Device C 通过端口 Ethernet1/2 连接 Server，并通过端口 Ethernet1/1 与 Device B 的端口 Ethernet1/1 相连。通过配置远程端口镜像，使 Server 可以通过 GRE 隧道穿越三层网络监控所有进、出市场部的报文。组网图如下所示：

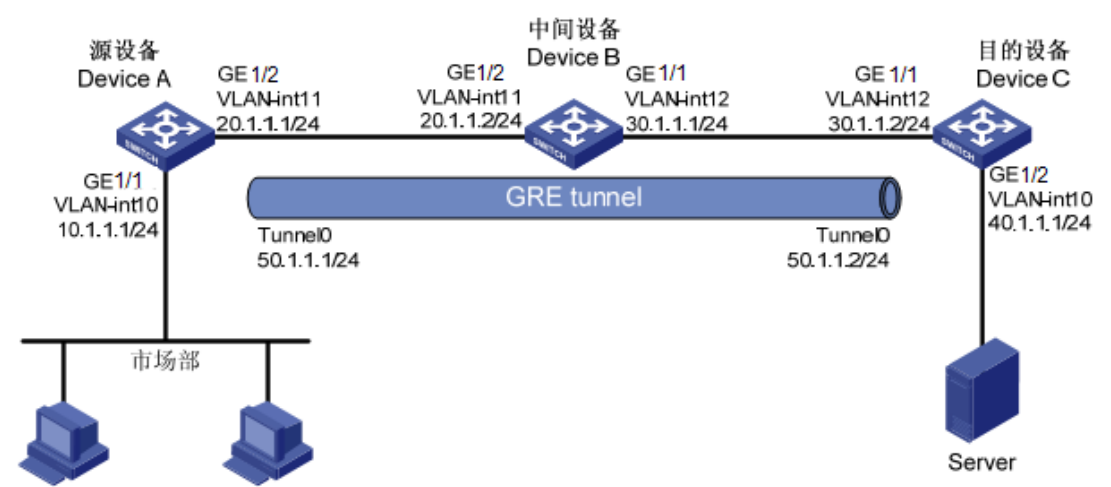


图 8-5 ERSPAN 应用示意图

在配置三层远程端口镜像之前，需完成源设备（Device A）到目的设备（Device C）

的 GRE 隧道的配置，并保证 GRE 隧道的正常转发。

三层远程端口镜像的配置需要分别在源设备和目的设备上进行。分别在源设备（Device A）和目的设备（Device C）上配置源端口和目的端口，不同的是：在 Device A 上，需要将源端口指定为待监控的端口，目的端口指定为 Tunnel 接口；在 Device C 上，需要将源端口指定为 Tunnel 接口对应的物理端口，目的端口指定为连接数据监测设备的端口。

(1) 配置 IP 地址

请按照组网图所示配置 VLAN，并将相应端口加入 VLAN，配置各接口的 IP 地址和子网掩码，具体配置过程略。

(2) 配置源交换机 Device A

创建接口 Tunnel1，并为其配置 IP 地址和掩码。

```
SwitchA(config)#interface tunnel 1
```

```
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA (config-if-tunnel1)#ip address 50.1.1.1 255.255.255.0
```

配置 Tunnel1 接口采用 GRE 隧道模式，并为该接口指定源地址和目的地址。

```
SwitchA(config-if-tunnel1)# tunnel source 10.1.1.1
```

```
SwitchA(config-if-tunnel1)# tunnel destination 40.1.1.1
```

```
SwitchA(config-if-tunnel1)# exit
```

配置 OSPF 协议。

```
SwitchA (config)#router ospf
```

```
SwitchA (config-router)#network 0.0.0.0/0 area 0
```

```
SwitchA (config-router)#exit
```

配置本地镜像组 4 的源端口为 Ethernet1/1，目的端口为 Tunnel1。

```
SwitchA(config)#monitor session 4 destination tunnel 1
```

```
SwitchA(config)#monitor session 4 source interface ethernet 1/1 both
```

(3) 配置中间交换机 Device B

配置 OSPF 协议。

```
SwitchB (config)#router ospf
```

```
SwitchB (config-router)#network 0.0.0.0/0 area 0
```

```
SwitchB (config-router)#exit
```

(4) 配置目的交换机 Device C

创建接口 Tunnel1，并为其配置 IP 地址和掩码。

```
SwitchC(config)#interface tunnel 1
```

```
SwitchC (config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchC (config-if-tunnel1)#ip address 50.1.1.2 255.255.255.0
```

配置 Tunnel1 接口采用 GRE 隧道模式，并为该接口指定源地址和目的地址。

```
SwitchC (config-if-tunnel1)# tunnel source 40.1.1.1
```

```
SwitchC (config-if-tunnel1)# tunnel destination 10.1.1.1
```

```
SwitchC (config-if-tunnel1)# exit
```

配置 OSPF 协议。

```
SwitchC (config)#router ospf
```

```
SwitchC (config-router)#network 0.0.0.0/0 area 0
```

```
SwitchC (config-router)#exit
```

配置本地镜像组 1 的源端口为 Ethernet1/1，目的端口为 Ethernet1/2。

```
SwitchC (config)#monitor session 1 destination interface ethernet 1/2
```

```
SwitchC (config)#monitor session 1 source interface ethernet 1/1 rx
```

8.10.4 ERSPAN 排错帮助

当配置 ERSPAN 功能出现问题时，请检查是否是如下原因：

- ☞ 确认 GRE 隧道的配置，并保证 GRE 隧道的正常转发；
- ☞ 镜像目的端口的吞吐量小于镜像源端口吞吐量的总和，目的端口无法完全复制源端口的流量；请减少源端口的个数或复制单向的流量，或者选择吞吐量更大的端口作为目的端口

第9章 网络时间及域名管理操作

9.1 SNTP

9.1.1 SNTP 简介

NTP (the Network Time Protocol)协议广泛用于维持全球 Internet 中的计算机的时钟同步。NTP 协议除了可以估算封包在网络上的往返延迟外,还可独立地估算计算机时钟偏差,从而实现在网络上的高精度度计算机校时。在大多数地方,根据同步源的特性和网络路径,NTP 提供 1~50ms 的精度。

SNTP (Simple Network Time Protocol) 是 NTP 协议的简化版本,除去了 NTP 复杂的算法。SNTP 用于那些不需要完整 NTP 实现复杂性的主机,它是 NTP 的一个子集。通常让局域网上的若干台主机通过因特网与其他的 NTP 主机同步时钟,接着再向局域网内其他客户端提供时间同步服务。下图描画了一个 NTP/SNTP 的应用网络结构,其中 SNTP 主要工作在二级服务器和各种终端之间,主要是由于这些场景时间精度要求不高,而 SNTP 本身可以提供的时间精度(1—50ms)一般可以满足这些服务的应用。

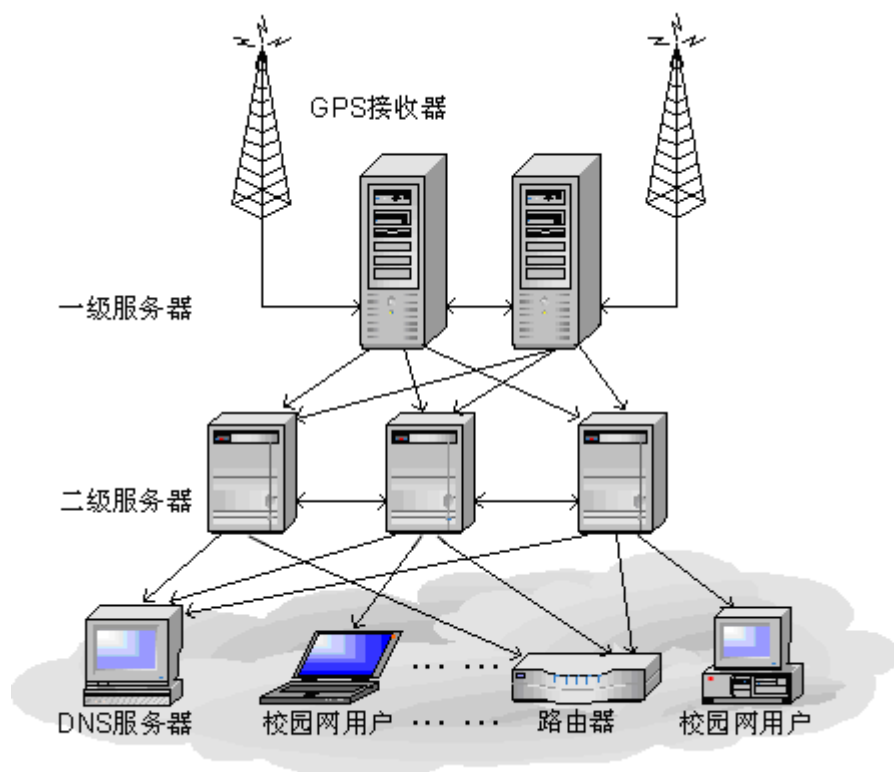


图 9-1 NTP/SNTP 工作场景

交换机实现了 SNTP 的客户端，支持 RFC2030 描述的 SNTP 客户端单播模式（unicast）的协议行为，不支持 SNTP 客户端组播模式（multicast）和任播模式（anycast），也不支持 SNTP 服务器端功能。

9.1.2 SNTP 典型配置举例

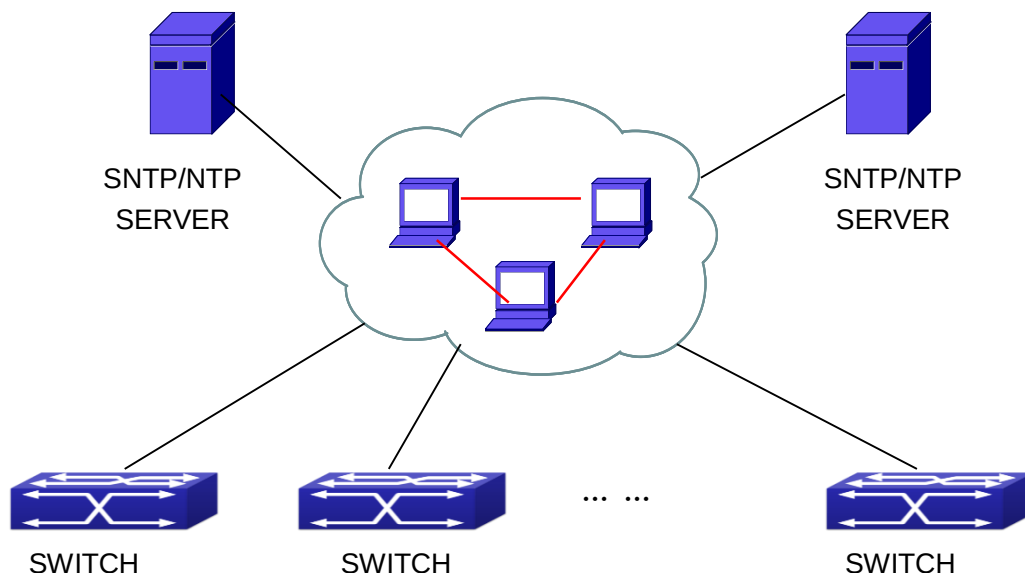


图 9-2 SNTP 典型配置

在自治系统域内的所有交换机需要进行时间同步，时间同步是通过两台冗余的 SNTP/NTP 服务器实现。为达到时间同步，当前网络必须正确配置，任意一台交换机和两台 SNTP/NTP 服务器间有可达的路由。设两个 SNTP/NTP Server 的 IP 地址分别配置为 10.1.1.1 和 20.1.1.1，并且已经开启 SNTP/NTP 服务器端功能（如 ntp master）则任意一台交换机配置如下：

```
Switch#config
Switch (config)#sntp server 10.1.1.1
```

9.2 NTP

9.2.1 NTP 功能简介

NTP（Network Time Protocol，网络时间协议）是一个跨越广域网或局域网的复杂的同步时间协议，它通常可获得毫秒级的精度，用来在分布式时间服务器和客户端之间进行时

间同步。RFC-1305 对 NTP 协议自动机在事件、状态、转变功能和行为方面给出了明确的说明。

NTP 的目标是对网络内所有具有时钟的设备进行时钟同步，使网络内所有设备的时钟基本保持一致，从而使设备能够提供基于统一时间的多种应用。

对于运行 NTP 的本地系统，既可以接受来自其他时钟源的同步，又可以作为时钟源同步其他时钟，并且可以通过交换 NTP 报文互相同步。

9.2.2 NTP 功能配置任务序列

1. 启动 NTP 功能
2. 配置 NTP 时间服务器的功能
3. 配置 NTP 广播或组播 server 地址的最大个数
4. 配置时钟区域
5. 配置 NTP 访问控制列表
6. 配置 NTP 验证
7. 配置某接口为 NTP 广播/组播客户端接口
8. 配置禁止某接口接收 NTP 报文
9. 设置 ntp client 请求报文发包间隔
10. 信息显示
11. 调试

1. 启动 NTP 功能

命令	解释
全局配置模式	
ntp enable ntp disable	全局启动或关闭 NTP 功能。

2. 配置 NTP 时间服务器的功能

命令	解释
全局配置模式	
ntp server {<ip-address> <ipv6-address>} [version <version_no>] [key <key-id>] no ntp server {<ip-address> <ipv6-address>}	启用作为时钟源的指定时间服务器。

3. 配置 NTP 广播或组播 server 地址的最大个数

命令	解释
全局配置模式	

ntp broadcast server count <number>	设置 NTP 广播或组播 server 地址的最大个数
no ntp broadcast server count	

4. 配置时钟区域

命令	解释
全局配置模式	
clock timezone WORD {add subtract} <0-23> [<0-59>] no clock timezone WORD	此命令在全局配置时区，no 操作命令为删除设置的时区。

5. 配置 NTP 访问控制列表

命令	解释
全局配置模式	
ntp access-group server <acl> no ntp access-group server <acl>	设置 NTP 服务的访问控制列表。

6. 配置 NTP 验证

命令	解释
全局配置模式	
ntp authenticate no ntp authenticate	启用 NTP 身份验证功能。
ntp authentication-key <key-id> md5 <value> no ntp authentication-key <key-id>	定义 NTP 身份验证的验证密钥。
ntp trusted-key <key-id> no ntp trusted-key <key-id>	配置可信赖密钥。

7. 配置某接口为 NTP 广播/组播客户端接口

命令	解释
vlan 配置模式	
ntp broadcast client no ntp broadcast client	配置指定端口来接收 NTP 广播包。
ntp multicast client no ntp multicast client	配置指定端口来接收 NTP 组播包。
ntp ipv6 multicast client no ntp ipv6 multicast client	配置指定端口来接收 IPv6 的 NTP 组播包。

8. 配置禁止某接口接收 NTP 报文

命令	解释
vlan 配置模式	
ntp disable no ntp disable	关闭接口上的 NTP 功能。

9. 设置 ntp client 请求报文发包间隔

命令	解释
全局配置模式	
ntp syn-interval <1-3600> no ntp syn-interval	设置 ntp client 请求报文发包间隔 1s-3600s。no 命令恢复默认值 64s。

10. 信息显示

命令	解释
特权模式	
show ntp status	显示时间同步的具体状况。
show ntp session [<ip-address> <ipv6-address>]	显示当前所有的或某个具体的 ntp 会话具体信息。

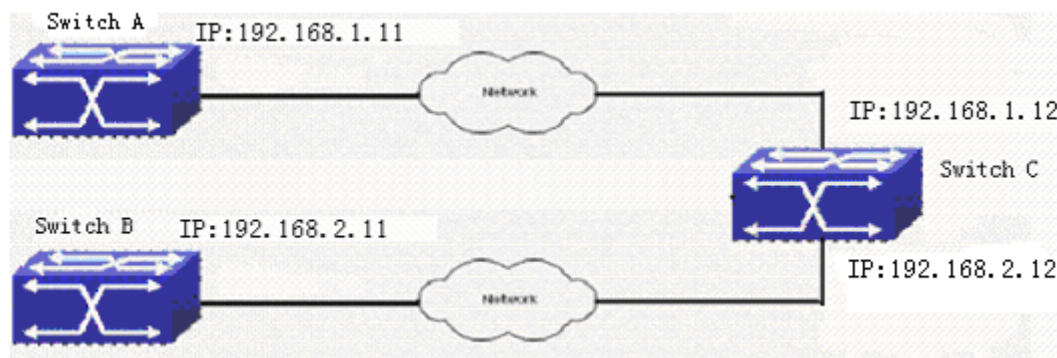
11. 调试

命令	解释
特权模式	
debug ntp authentication no debug ntp authentication	打开 NTP 认证信息的调试开关。
debug ntp packets [send receive] no debug ntp packets [send receive]	打开 NTP 数据包信息的调试开关。
debug ntp adjust no debug ntp adjust	打开时间更新信息的调试开关。
debug ntp sync no debug ntp sync	打开时间同步信息的调试开关。
debug ntp events no debug ntp events	打开 NTP 事件信息的调试开关。

9.2.3 NTP 功能典型案例

在网络中一台客户端交换机需要与网络中的时间服务器同步时钟，网络中有两台时间服

务器，一个作为主用，另一个作为备用，具体连接和配置如下（其中 SwitchA 与 SwitchB 为支持 NTP server 的交换机/路由器）：



Switch C 配置如下：(Switch A 与 Switch B 依据各厂家的实现可能配置命令有所不同，这里暂不说明，我们的交换机目前不支持 NTP Server 的功能)

Switch C:

```
Switch(config)#ntp enable
```

```
Switch(config)#interface vlan 1
```

```
Switch(Config-if-Vlan1)# ip address 192.168.1.12 255.255.255.0
```

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

```
Switch(Config-if-Vlan1)# ip address 192.168.2.12 255.255.255.0
```

```
Switch(config)#ntp server 192.168.1.11
```

```
Switch(config)#ntp server 192.168.2.11
```

9.2.4 NTP 功能排错帮助

在操作过程中，如果有操作错误，系统将会给出具体的错误提示信息。

NTP 功能默认是关闭的，可以使用显示命令看目前的配置情况。在确认配置正确的情况下，可以使用 debug 各个相关功能的调试命令，查看过程中的具体信息，以及功能是否正确，也可以使用相关的 show 命令查看 NTP 目前的运行信息，如果有什么问题可以将结果发送给技术服务中心。

9.3 DNSv4/v6

9.3.1 DNS 简介

DNS（域名系统，Domain Name System）是一种用于 TCP/IP 应用程序的分布式数据库，提供域名与 IPv4/IPv6 地址之间的转换。通过域名系统，用户进行某些应用时，可以直接使用便于记忆的、有意义的域名，而由网络中的域名解析服务器解析为正确的 IPv4/IPv6 地址。

域名解析分为静态域名解析和动态域名解析，二者在实际应用中相辅相成。在解析域名时，可以首先采用静态域名解析的方法，如果静态域名解析不成功，再采用动态域名解析的方法。可以将一些常用的域名放入静态域名解析表中，这样可以大大提高域名解析效率。静态域名解析是手工建立域名和 IPv4/IPv6 地址之间的对应关系。当用户使用域名进行某些应用时系统查找静态域名解析表，从中获取指定域名对应的 IPv4/IPv6 地址。动态域名解析是通过对域名服务器的查询完成的。当用户使用域名进行某些应用时，作为 DNS 客户端的交换机向域名服务器发出查询请求，域名服务器从自己的数据库中找出该域名对应的 IPv4/IPv6 地址并回送给交换机。如果判断不属于本域范围之内，就将请求交给上一级的域名解析服务器，直到完成解析为止。

DNS 分为“Client”和“Server”，“Client”扮演发问的角色，也就是问“Server”一个“Domain Name”，而“Server”必须要回答此“Domain Name”的真正 IPv4/IPv6 地址。而本地的 DNS 先会查自己的资料库，如果自己的资料库没有，则会往该 DNS 上所设置的 DNS 询问，依此得到答案之后，将收到的答案存起来，并回答客户。

DNS 服务器会根据不同的授权区(Zone)，记录所属该网域下的各名称资料，这个资料包括网域下的次网域名称及主机名称。在每一个名称服务器中都有一个快取缓存区(Cache)，这个快取缓存区的主要目的是将该名称服务器所查询出来的名称及相对的 IPv4/IPv6 地址记录在快取缓存区中，这样当下一次有另外一个客户端到该服务器上去查询相同的名称时，服务器就不用再到别的主机上去寻找，而直接可以从缓存区中找到该项名称记录资料，传回给客户端，加速客户端对名称查询的速度。例如：

DNS 客户端向指定的 DNS 服务器查询网际网络上某台主机名称，当 DNS 服务器在该资料记录找不到用户所指定的名称时，会转向该服务器的快取缓存区找寻是否有该资料，当快取缓存区也找不到时，会向最接近的名称服务器去要求帮忙找寻该名称的 IPv4/IPv6 地址，在另一台服务器上也有相同动作的查询，当查询到后会回复原本要求查询的服务器，该 DNS 服务器在接收到另一台 DNS 服务器查询的结果后，先将所查询到的主机名称及对应的地址记录到快取缓存区中，最后再将所查询到的结果回复给客户端。

9.3.2 DNSv4/v6 配置任务序列

1. 启动/关闭 DNS 功能
2. 配置/删除 DNS 服务器
3. 配置/删除域名后缀
4. 删除动态缓存中指定地址的域名表项
5. 执行 DNS 动态域名解析
6. 启动/关闭 DNS SERVER 功能
7. 配置交换机缓存客户信息的最大数量
8. 配置交换机缓存客户信息的最大超时时间
9. DNS 功能的维护与诊断

1. 启动/关闭 DNS 功能

命令	解释
全局配置模式	
ip domain-lookup no ip domain-lookup	启动/关闭 DNS 动态查询功能。

2. 配置/删除 DNS 服务器

命令	解释
全局配置模式	
dns-server {<ip-address> <ipv6-address>} [priority <value>] no dns-server {<ip-address> <ipv6-address>}	配置 DNS 服务器，no 命令删除 DNS 服务器。

3. 配置/删除域名后缀

命令	解释
全局配置模式	
ip domain-list <WORD> no ip domain-list <WORD>	配置/删除域名后缀。

4. 删除动态缓存中指定地址的域名表项

命令	解释
特权模式	
clear dynamic-host {<ip-address> <ipv6-address>}all }	删除动态缓存中指定地址的域名表项。

5. 执行 DNS 动态域名解析

命令	解释
全局配置模式	
dns lookup {ipv4 ipv6} <hostname>	执行 DNS 动态域名解析。

6. 启动/关闭 DNS SERVER 功能

命令	解释
全局配置模式	
[no] ip dns server	启动/关闭 DNS SERVER 功能。

7. 配置交换机缓存客户信息的最大数量

命令	解释

全局配置模式	
[no] ip dns server queue maximum <1-5000>	配置交换机缓存客户信息的最大数量。

8. 配置交换机缓存客户信息的最大超时时间

命令	解释
全局配置模式	
[no] ip dns server queue timeout <1-100>	配置交换机缓存客户信息的最大超时时间。

9. DNS 功能的维护与诊断

命令	解释
特权模式或者配置模式	
show dns name-server	显示配置的 DNS 服务器信息。
show dns domain-list	显示配置的 DNS 域名后缀信息。
show dns hosts	显示交换机中解析到的动态域名信息。
show dns config	显示交换机中配置的 DNS 全局信息。
show dns client	显示交换机中维护的 DNS Client 信息。
[no] debug dns {all packet [send recv]] events relay}	打开/关闭 DNS 功能的 DEBUG 显示。

9.3.3 DNS 典型案例

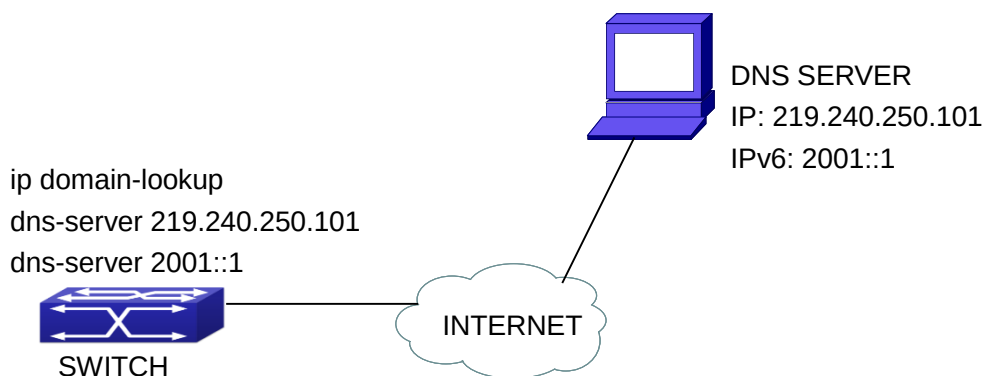


图 9-3 DNS CLIENT 典型环境

如上图所示，交换机通过网络连接到 DNS 服务器上，交换机如果想访问新浪网，它不用知道新浪网的 IPv4/IPv6 地址，只要记住新浪网的域名 `www.sina.com.cn` 就可以了，DNS 服务器会解析出该域名对应的 IPv4/IPv6 地址并告知交换机，从而交换机可以正常访问新浪

网。交换机做为 DNS 客户端，基本操作过程如下：首先在交换机上启动 DNS 动态域名解析功能，配置 DNS 服务器地址，然后在各种应用工具中（例如 ping 等）就可以进行动态域名解析功能，获得域名对应的 IPv4/IPv6 地址。

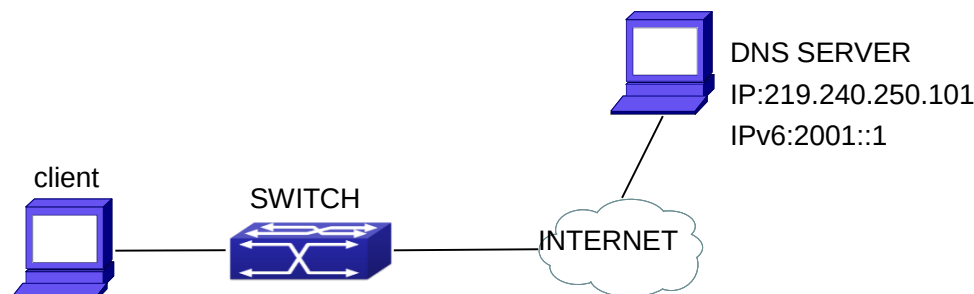


图 9-4 DNS SERVER 典型环境

如上图，是 DNS SERVER 的应用场景。在某些应用场景下，客户端 PC 不用知道真正的 DNS SERVER，它指向的 DNS SERVER 是交换机。交换机担当 DNS SERVER 的功能。交换机作为 DNS 服务器，基本操作过程：首先在交换机上启动全局 DNS SERVER 功能，配置真正的 DNS 服务器 IP 地址。交换机全局启动了 DNS SERVER 功能后，当交换机收到客户端 PC 的 DNS 请求后，会查询自己本地的 cache，如果本地有客户端需要的域名，就会直接答复客户端，如果没有则交换机会 relay 该客户端的请求到真正的 DNS 服务器，交换机收到 DNS SERVER 的 reply 后，回复给发送该请求的客户端，同时记录该域名的 IP 地址对应关系，提高下次查询的速度。

Switch 作为 DNS CLIENT 的配置：

```
Switch(config)#ip domain-lookup
Switch(config)#dns-server 219.240.250.101
Switch(config)#dns-server 2001::1
Switch#ping host www.sina.com.cn
Switch#traceroute host www.sina.com.cn
Switch#telnet host www.sina.com.cn
```

Switch 作为 DNS SERVER 的配置：

```
Switch(config)#ip domain-lookup
Switch(config)#dns-server 219.240.250.101
Switch(config)#dns-server 2001::1
Switch(config)#ip dns server
```

9.3.4 DNS 排错帮助

在配置、使用 DNS 功能时，可能会由于物理连接、配置错误等原因导致 DNS 功能异常。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后，使用 DNS CLIENT 功能时确保开启了 DNS 动态查询功能（使用 ip domain-lookup 命令）。使用 DNS SERVER 功能时需要开启 DNS SERVER 功能（使用 ip dns server 命令）；
- ☞ 另外，确保配置了 DNS 服务器地址(使用 dns-server 命令)，并且交换机能够 Ping 通 DNS 服务器；
- ☞ 如果尝试使用以上检查仍无法解决 DNS 的问题，那么请使用 debug dns all 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给技术服务中心。

9.4 夏令时

9.4.1 夏令时介绍

夏令时，又称“日光节约时制”或“夏时制”，是一种为节约能源而人为规定地方时间的制度，在这一制度实行期间所采用的统一时间称为“夏令时间”。一般在天亮早的夏季人为将时间提前一小时，可以使人早起早睡，减少照明量，以充分利用光照资源，从而节约照明用电。各个采纳夏令时的国家具体规定不同。目前全世界有近 110 个国家每年要实行夏令时。（各时区多数位于其理想边界之西，导致实际上全年实施夏令时）。

绝大多数实行夏令时的国家，都是规定夏令时比标准时晚一个小时。例如，在夏令时的实施期间，标准时间的上午 10 点就成了夏令时的上午 11 点。

9.4.2 夏令时配置任务序列

1. 设置夏令时绝对或循环时间段

命令	解释
全局配置模式	
clock summer-time <word> absolute <HH:MM> <YYYY.MM.DD> <HH:MM> <YYYY.MM.DD> [<offset>] no clock summer-time	设置夏令时绝对时间段，只在设置年份的指定时间开始和结束夏令时。
clock summer-time <word> recurring <HH:MM> <MM.DD> <HH:MM> <MM.DD> [<offset>] no clock summer-time	设置夏令时循环时间段，每年时间到达夏令时起始点时开始夏令时，到达夏令时结束点时结束夏令时。
clock summer-time <word> recurring <HH:MM>	设置夏令时循环时间段，每年

<code><week> <day> <month> < HH:MM > <week> <day> <month> [<offset>] no clock summer-time</code>	时间到达夏令时起始点时开始夏令时，到达夏令时结束点时结束夏令时。

9.4.3 夏令时举例

案例 1:

用户有如下的配置需求：在 2012 年 4 月 1 日 23:00 到 2012 年的 10 月 1 日 00:00 期间实行夏令时，时钟偏移量为 1 小时，并将该夏令时配置命名为 2012。

配置步骤如下：

```
Switch(config)# clock summer-time 2012 absolute 23:00 2012.4.1 00:00 2012.10.1
```

案例 2:

用户有如下的配置需求：每年 4 月第一个星期天 23:00 到这一年的 10 月最后一个星期日 00:00 期间实行夏令时，时钟偏移量为 2 小时，并将该夏令时配置命名为 time_travel。

配置步骤如下：

```
Switch(config)# clock summer-time time_travel recurring 23:00 first sat apr 00:00 last sun  
oct 120
```

9.4.4 夏令时排错帮助

当配置夏令时功能出现问题时，请检查是否是如下原因：

- ☞ 命令模式是否在全局模式。
- ☞ 系统时钟是否正确。

第10章 vpn 操作

10.1 MPLS

10.1.1 MPLS 概述

MPLS (Multiprotocol Label Switching, 多协议标签交换) 起源于 IPv4, 最初是为了提高转发速度而提出的, 其核心技术可扩展到多种网络协议, 包括 IPv6 (Internet Protocol version 6)、IPX (Internet Packet Exchange)、Appletalk、DECnet、CLNP (Connectionless Network Protocol) 等。MPLS 中的 “Multiprotocol” 指的就是支持多种网络协议。MPLS 技术集二层的快速交换和三层的路由转发于一体, 可以满足各种新应用对网络的要求。

10.1.1.1 MPLS 简介

转发等价类

MPLS 作为一种分类转发技术, 将具有相同转发处理方式的分组归为一类, 称为 FEC (Forwarding Equivalence Class, 转发等价类)。相同 FEC 的分组在 MPLS 网络中将获得完全相同的处理。FEC 是一组三层报文, 它们在同样的路径上、按照相同的转发待遇、以相同的模式被转发。转发决定可以分为两步:

- ☞ 分析分组头并将分组分成 FEC;
- ☞ 将 FEC 映射到下一跳。

传统 IP 转发网络中, 每个路由器对相同分组都要进行 FEC 分类和选择下一跳。FEC 可以包含一个或多个 FEC 单元, 每个 FEC 单元是一组可以映射到相同 LSP 的三层报文。

目前, 定义了两种类型的 FEC 单元:

- ☞ 地址前缀型(Address Prefix): 使用地址前缀作为 FEC 单元, 其长度为从 0 到完整的地址。前缀型 FEC 单元对应于一个目标子网;
- ☞ 主机地址型(Host Address): 使用终端地址作为一个 FEC 单元, 对应于一个终端地址。

FEC 的划分方式非常灵活, 可以是源地址、目的地址、源端口、目的端口、协议类型、VPN 等的任意组合。例如, 在传统的采用最长匹配算法的 IP 转发中, 到同一个目的地址的所有报文就是一个 FEC。

标签

MPLS 网络中, 在边缘 LSR 处每一特定 FEC 都被编码为一个短而定长的值, 称为标签, 标记加在分组前成为标签分组, 再转发到下一跳。标签中, 除了表示 FEC 的标记, 还可以(但不是必须)包含 COS 字段。此时, 标签就成为代表 FEC、优先级和服务类型的综合体。LSR 可以将到达不同端口的分组分配到不同的 FEC, 这就形成了建立 VPN 的基础。当 LSR 创建一个新的 FEC, LSR 就创建一个对应的标签, 并把标签分发给其所有 peer。LSR 维护输入和输出两个方向的标签。假如要进行负载分担, 对应一个 FEC 可能会有多个标签, 但是一个标签只能代表一个 FEC。

标签由报文的头部所携带, 不包含拓扑信息, 只具有局部意义。标签的长度为 4 个字

节，封装结构如图 1-1 所示：



标签的封装结构

标签共有 4 个域：

- ☞ Label: 标签值字段，长度为 20bits，用于转发的指针；
- ☞ Exp: 3bits，用于 QoS；
- ☞ S: 1bit，MPLS 支持标签的分层结构，即多重标签。值为 1 时表明为最底层标签；
- ☞ TTL: 8bits，和 IP 分组中的 TTL（Time To Live，生存时间）意义相同。

标签与 ATM 的 VPI/VCI 以及 Frame Relay 的 DLCI 类似，是一种连接标识符。如果链路层协议具有标签域，如 ATM 的 VPI/VCI 或 Frame Relay 的 DLCI，则标签封装在这些域中；如果不支持，则标签封装在链路层和 IP 层之间的一个垫层中。这样，标签能够被任意的链路层所支持。

标签空间

LSR 可以基于输入端口为一个 FEC 分配不同的标签。这样，从不同端口输出的分组就可以独立转发，这是实现 VPN 的基础。为了提高标签利用率，MPLS 提出了标签空间的概念。标签空间是一个标签前缀，在分配标签时，对于不同标签空间的转发等价类，可以分配相同的标签，这样就相当于扩展了标签的范围。标签空间只对标签的分配有意义，对于标签转发无意义。

标签交换

在非边缘 LSR，不再需要分析分组头，而是用标签作为指针，指向下一跳的输出端口和一个新的标签，标签分组用新标签替代旧标签后经指定的输出端口转发。标签交换不仅可以简化转发、提高速度，还可以实现 VPN，QOS，流量工程等应用。

标签交换路由器

LSR（Label Switching Router，标签交换路由器）是 MPLS 网络中的基本元素，所有 LSR 都支持 MPLS 技术。

LSR 是一种可以基于分组中的标签值来转发分组的设备。连接 IP 路由网和 MPLS 交换网的 LSR，被称为边缘 LSR，它对 IP 报文添加标签，然后按照 LSP 转发数据；或者删除 MPLS 分组标签，按照 IP 路由转发数据。每个 LSR 必须分配一个全局唯一的标识符(LSR ID)，通常取 LSR 一个接口的 IP 地址。假设，LSR Ru 和 Rd 对标签 L 和 FEC F 之间的映射关系达成一致，分组可以利用标签 L 从 Ru 转发到 Rd，则把 Ru 称为上游 LSR，Rd 称为下游 LSR。也就是说，分组总是从上游 LSR 向下游 LSR 转发。

标签交换路径

一个转发等价类在 MPLS 网络中经过的路径称为标签交换路径 LSP（Label Switched Path）。在一条 LSP 上，沿数据传送的方向，相邻的 LSR 分别称为上游 LSR 和下游 LSR。如图 1-2 中，R2 为 R1 的下游 LSR，相应的，R1 为 R2 的上游 LSR。

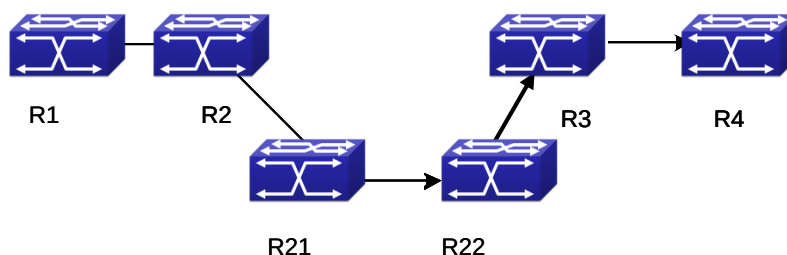


图 10-1 标签交换路径 LSP

LSP 在功能上与 ATM 和帧中继 (Frame Relay) 的虚电路相同, 是从 MPLS 网络的入口到出口的一个单向路径。LSP 中的每个节点由 LSR 组成。

下游 LSR 向上游 LSR 分发标签, 连成一串的标签和 LSR 序列就构成 LSP。LSP 是把 IP 层路由信息映射到链路层的一条交换路径。LSP 是一条单向分组转发路径, 分组总是从上游 LSR 转发到下游 LSR。需要反方向转发的分组, 必须建立另外一条完全独立的 LSP。LDP 总是把 FEC 和 LSP 关联起来, FEC 和 LSP 的关联关系被称为报文映射到 LSP。

1. 映射报文到 LSP 需要遵循的规则:

- 如果只有一个 LSP, 它包含一个终端地址型的 FEC 单元和报文的地址相同, 则报文被映射到这个 LSP;
- 如果存在多个 LSP 满足条件 1, 则报文被映射到这些 LSP 中的一个;
- 如果只有一个 LSP, 报文可以匹配其包含的地址前缀型 FEC 单元, 则报文被映射到这个 LSP;
- 如果存在多个 LSP 满足条件 3, 则根据最长匹配原则选择 LSP;
- 如果能够确定一个分组必须通过某个特定出口 LSR, 并且存在一条 LSP, 它绑定的 FEC 的前缀型成员就是出口 LSR 的地址, 则报文被映射到这条 LSP。

2. 附加规则:

- 如果报文目的地址没有匹配任何 LSP, 报文可以在与其出口节点(Egress Router)地址相同的包含地址前缀型 FEC 单元的 LSP 上发送;
- 如果报文匹配了两个 LSP, 一个 LSP 包含终端地址型的 FEC 单元, 一个 LSP 包含地址前缀型 FEC 单元, 报文总是映射到前者;
- 如果报文没有匹配包含终端地址型的 FEC 单元的 LSP, 即使终端地址型的 FEC 单元与其出口节点地址相同, 也不能在这个 LSP 上发送。
- LSP 的创建是基于连接的, 这种连接是根据拓扑信息, 而不是数据流的要求建立的。也就是说, 不管是否存在通过该路由转发的数据, LSP 都将被建立。

标签合并

LSR 把多个入标签映射到相同的 FEC, 这些入标签对应的出标签和出接口都将是相同的。这样当携带不同标签的分组到达 LSR 时, 输出分组将携带相同的标签, 这种情况被称为标签合并。标签合并可以减少 MPLS 域的标签数量, 但可能丢失分组入接口信息。

如果 LSR 不支持标签合并, 对于多个标签请求, 无论它们的 FEC 是否相同, 都将向下游 LSR 发起新的标签请求。如果 LSR 支持标签合并, 对于 FEC 的相同标签请求, LSR 只向下游 LSR 发起一次标签请求。

标签分发协议

标签分发协议 LDP (Label Distribution Protocol) 是 MPLS 的控制协议, 它相当于传统网络中的信令协议, 负责 FEC 的分类、标签的分配以及 LSP 的建立和维护等一系列操作。

MPLS 可以使用多种标签发布协议，包括专为标签发布而制定的协议，例如：LDP、CR-LDP（Constraint-Based Routing using LDP，基于约束路由的 LDP）；也包括现有协议扩展后支持标签发布的协议，例如：BGP（Border Gateway Protocol，边界网关协议）、RSVP（Resource Reservation Protocol，资源预留协议）。同时，还可以手工配置静态 LSP。

LSP 隧道技术

MPLS 支持 LSP 隧道技术。一条 LSP 的上游 LSR 和下游 LSR，尽管它们之间的路径可能并不在路由协议所提供的路径上，但是 MPLS 允许在它们之间建立一条新的 LSP，这样，上游 LSR 和下游 LSR 分别就是这条 LSP 的起点和终点。这时，上游 LSR 和下游 LSR 间的 LSP 就是 LSP 隧道，它避免了采用传统的网络层封装隧道。如图 1-2 中 LSP<R2→R21→R22→R3>就是 R2、R3 间的一条隧道。

如果隧道经由的路由与从路由协议取得的路由一致，这种隧道就称为逐跳路由隧道（Hop-by-Hop Routed Tunnel）；否则称为显式路由隧道（Explicitly Routed Tunnel）。

多层标签栈

如果分组在超过一层的 LSP 隧道中传送，就会有多层标签，形成标签栈（Label Stack）。在每一隧道的入口和出口处，进行标签的入栈（PUSH）和出栈（POP）操作。

标签栈按照“后进先出”（Last-In-First-Out）方式组织标签，MPLS 从栈顶开始处理标签。

MPLS 对标签栈的深度没有限制。若一个分组的标签栈深度为 m ，则位于栈底的标签为 1 级标签，位于栈顶的标签为 m 级标签。未压入标签的分组可看作标签栈为空（即标签栈深度为零）的分组。

10.1.1.2 MPLS 网络简介

MPLS 网络结构

如图 1-3 所示，MPLS 网络的基本构成单元是 LSR，由 LSR 构成的网络称为 MPLS 域。

位于 MPLS 域边缘、连接其它用户网络的 LSR 称为 LER（LER，Label Edge Router，边缘 LSR），区域内部的 LSR 称为核心 LSR。核心 LSR 可以是支持 MPLS 的路由器，也可以是由 ATM 交换机等升级而成的 ATM-LSR。域内部的 LSR 之间使用 MPLS 通信，MPLS 域的边缘由 LER 与传统 IP 技术进行适配。

分组在入口 LER 被压入上标签后，沿着由一系列 LSR 构成的 LSP 传送，其中，入口 LER 被称为 Ingress，出口 LER 被称为 Egress，中间的节点则称为 Transit。

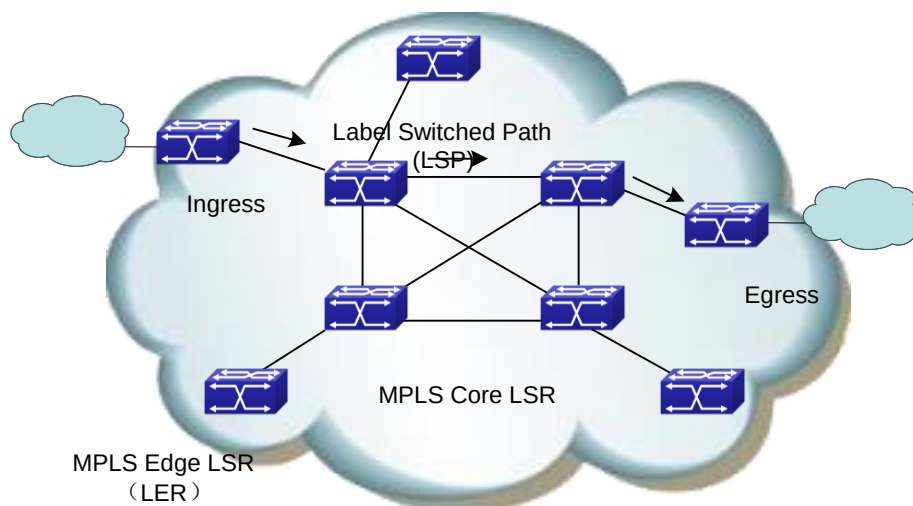


图 10-2 MPLS 网络结构

结合图 1-3 简要介绍 MPLS 的基本工作过程：

首先，LDP 和传统路由协议（如 OSPF、ISIS 等）一起，在各个 LSR 中为有业务需求的 FEC 建立路由表和标签信息表（Label Information Base，LIB）；

入口 LER 接收分组，完成第三层功能，判定分组所属的 FEC，并给分组加上标签，形成 MPLS 标签分组；

接下来，在 LSR 构成的网络中，LSR 根据分组上的标签以及标签转发表（Label Forwarding Information Base，LFIB）进行转发，不对标签分组进行任何第三层处理；

最后，在 MPLS 出口 LER 去掉分组中的标签，继续进行后面的 IP 转发。

由此可以看出，MPLS 并不是一种业务或者应用，它实际上是一种隧道技术，也是一种将标签交换转发和网络层路由技术集于一身的路由与交换技术平台。这个平台不仅支持多种高层协议与业务，而且，在一定程度上可以保证信息传输的安全性。

10.1.1.3 MPLS 与路由协议简介

LDP 通过逐跳方式建立 LSP 时，利用沿途各 LSR 路由转发表中的信息来确定下一跳，而路由转发表中的信息一般是通过 IGP、BGP 等路由协议收集的。LDP 并不直接和各种路由协议关联，只是间接使用路由信息。

另一方面，通过对 BGP、RSVP 等已有协议进行扩展，也可以支持 MPLS 标签的分发。

在 MPLS 的应用中，也可能需要对某些路由协议进行扩展。例如，基于 MPLS 的 VPN 应用需要对 BGP 进行扩展，使 BGP 能够传播 VPN（Virtual Private Network，虚拟专用网）的路由信息；基于 MPLS 的 TE（Traffic Engineering，流量工程）需要对 OSPF 或 IS-IS 协议进行扩展，以携带链路状态信息。

10.1.1.4 MPLS 应用简介

最初，MPLS 技术结合了二层交换技术和三层路由技术，提高了路由查找速度。但是，随着 ASIC（Application-Specific Integrated Circuit，专用集成电路）技术的发展，路由查找速度已经不成为阻碍网络发展的瓶颈。这使得 MPLS 在提高转发速度方面不具备明显的优势。

但由于 MPLS 结合了 IP 网络强大的三层路由功能和传统二层网络高效的转发机制，在转发平面采用面向连接方式，与现有二层网络转发方式非常相似，这些特点使得 MPLS 能够很容易地实现 IP 与 ATM、帧中继等二层网络的无缝融合，并为 QoS (Quality of Service, 服务质量)、TE、VPN 等应用提供更好的解决方案。

基于 MPLS 的 VPN

传统的 VPN 一般是通过 GRE、L2TP、PPTP 等隧道协议来实现私有网络间数据流在公网上的传送，LSP 本身就是公网上的隧道，因此，用 MPLS 来实现 VPN 有天然的优势。

基于 MPLS 的 VPN 就是通过 LSP 将私有网络的不同分支联结起来，形成一个统一的网络。基于 MPLS 的 VPN 还支持对不同 VPN 间的互通控制。

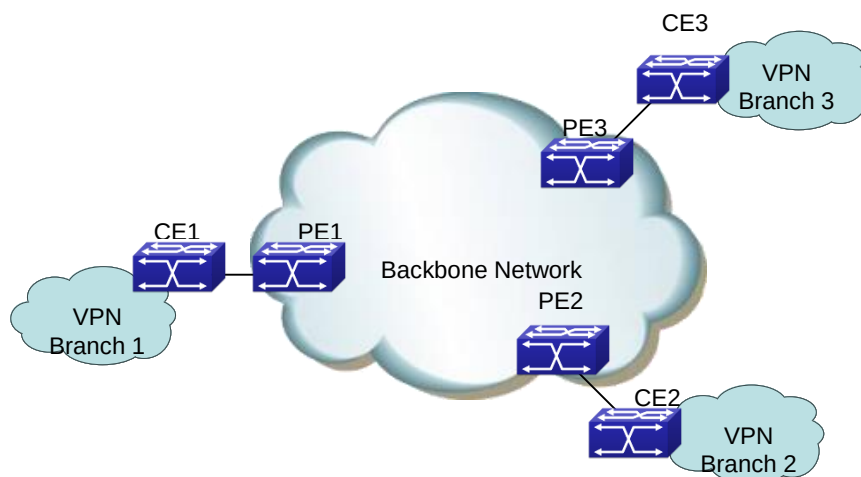


图 10-3 基于 MPLS 的 VPN

图 1-4 是基于 MPLS 的 VPN 的基本结构：CE (Customer Edge, 用户边缘设备) 可以是路由器，也可以是交换机或主机；PE (Provider Edge, 服务商边缘路由器) 位于骨干网络。

PE 负责对 VPN 用户进行管理、建立各 PE 间 LSP 连接、同一 VPN 用户各分支间路由分派。PE 间的路由分派通常是用 LDP 或扩展的 BGP 协议实现。

基于 MPLS 的 VPN 支持不同分支间 IP 地址复用，并支持不同 VPN 间互通。与传统的路由相比，VPN 路由中需要增加分支和 VPN 的标识信息，这就需要对 BGP 协议进行扩展，以携带 VPN 路由信息。

基于 MPLS 的流量工程

基于 MPLS 的 TE 和差分服务 Diff-serv 特性，在保证网络高利用率的同时，可以根据不同数据流的优先级实现差别服务，从而为语音、视频等数据流提供有带宽保证的低延时、低丢包率的服务。

由于全网实施流量工程的难度比较大，因此，在实际的组网方案中往往通过差分服务模型来实施 QoS。

Diff-Serv 的基本机制是在网络边缘，根据业务的服务质量要求将该业务映射到一定的业务类别中，利用 IP 分组中的 DS 字段（由 ToS 域而来）唯一的标记该类业务，然后，骨干网络中的各节点根据该字段对各种业务采取预先设定的服务策略，保证相应的服务质量。

Diff-Serv 的这种对服务质量的分类和标签机制和 MPLS 的标签分配十分相似，事实上，

基于 MPLS 的 Diff-Serv 就是通过将 DS 的分配与 MPLS 的标签分配过程结合来实现的。

10.1.1.5 MPLS PHP

在 MPLS 网络中, 核心 LSR 根据分组上的标签进行转发; 在 Egress 节点 (出口 LER) 去掉分组中的标签, 继续进行 IP 转发。

实际上, 在比较简单的 MPLS 应用中, Egress 节点只需要进行 IP 转发, 标签已经没有使用价值。这种情况下, 可以利用倒数第二跳弹出特性 PHP (Penultimate Hop Popping), 在倒数第二个节点处将标签弹出, Egress 节点就不再进行标签操作了。

10.2 LDP

10.2.1 LDP 简介

LDP 协议是在 MPLS 标签交换环境中用于标签分配的协议。LDP 协议只运行在具有标签交换能力的网络上。LDP 与传统路由算法相结合, 通过在 TCP 连接上传送各种消息, 分配标签、发布<标签, FEC>映射, 建立维护标签转发表和标签交换路径(LSP)。在 MPLS VPN 环境下, LDP 协议用于分配公网标签。

LDP 协议本身不产生路由, 它从系统获得路由信息, 为路由分配标签, 并把标签通告给其上游路由器。同时, 对于具有下游的 FEC, LDP 协议从其从下游接收标签作为出标签, 从而建立标签交换的通路, 即建立一个把入标签交换为出标签的表项。如果下游分配的标签为 3, 则 LDP 产生一个弹出标签的表项。

LDP 协议定义在 RFC3036, 最新的标准为 RFC5036, 它通过对等体间的 TCP 连接交换各自的标签和路由的对应关系, 其发现邻居的方式包括基本方式(自动发现)和扩展方式(指定)两种。对等体的自动发现使用发往所有路由器 (224.0.0.2) 的 UDP 组播报文, 其使用的 TCP 和 UDP 报文的端口号都是 646。

其主要运行过程如下:

- ☞ 邻居发现与维护: 在全局和指定接口上启动 LDP 协议后, LDP 协议在指定接口上向外发送组播 Hello 报文 (除非配置了不采用组播报文发现邻居) 向网络通告自己的存在。Hello 报文中携带了自己的传输地址, 即用于建立 TCP 连接的地址。在收到其它 LSR 发出的 Hello 报文后, 创建邻居关系; 邻居关系依靠周期性发送 Hello 报文来维护。
- ☞ 会话建立与维护: LDP 会话是建立在 TCP 连接之上的; 首先比较对端 Hello 报文中携带的传输地址和本端的传输地址, 地址比较大的成为 ACTIVE, 地址比较小的成为 PASSIVE, ACTIVE 的路由器主动发起连接请求以建立 TCP 连接 (这是为了避免类似 BGP 邻居的连接冲突问题)。TCP 连接建立后, 连接双方发送初始化报文以协商会话参数, 协商一致建立会话关系。会话关系后互相把本地的接口地址列表以及标签信息发送给自己的邻居。没有信息时通过发送 KEEPALIVE 消息以保持连接。

- ☞ LSP 的建立与维护：每两个 LSR 对等体之间需要一个会话关系来交换标签信息，LSR 之间通过交换 FEC 和标签绑定消息来建立 LSP。
- ☞ 会话的撤销：若长时间收不到对端的消息，LDP 会中断连接并发出 notification 消息通知对端关闭会话连接。

值得注意的是，LDP 协议不会为默认路由分配标签，另外，除非特别指定，LDP 协议也不会为 BGP 路由分配标签。

10.2.1.1 LDP 基本概念

LDP 对等体

LDP 分配标签给 FEC 的时候，需要把此标签及其意义在 MPLS 网络内传播，从而建立 LSP。采用 LDP 交换标签信息的 LSR 被称为 LDP 对等体(LSP Peer)。LDP 对等体通过它们之间的 LDP 会话获得对方的标签映射消息以及其他消息。

LDP 会话

交换 LDP Discovery Hello 报文之后，LSR 之间就会建立 LDP 会话。LDP 会话用于在 LSR 之间交换标签映射、释放等各种消息。

LDP 会话的建立可以分为两步：

- ☞ 建立传输连接
- ☞ 初始化会话

LDP 会话可以分为两种类型：

- ☞ 本地 LDP 会话（Local LDP Session）：建立会话的两个 LSR 之间是直连的；
- ☞ 远端 LDP 会话（Remote LDP Session）：建立会话的两个 LSR 之间是非直连的。

LDP 消息类型

LDP 协议主要使用四类消息：

- ☞ 发现（Discovery）消息：用于通告和维护网络中 LSR 的存在；
- ☞ 会话（Session）消息：用于建立、维护和终止 LDP 对等体之间的会话；
- ☞ 通告（Advertisement）消息：用于创建、改变和删除标签到 FEC 的映射；
- ☞ 通知（Notification）消息：用于提供建议性的消息和差错通知。

为保证 LDP 消息的可靠发送，除了发现阶段使用 UDP 传输外，LDP 的 Session 消息、Advertisement 消息和 Notification 消息都使用 TCP 传输。

10.2.1.2 LDP 报文格式简介

LDP PDU

LDP PDU 包括 LDP 报文头和若干个 LDP 消息。LDP 报文头格式如下：

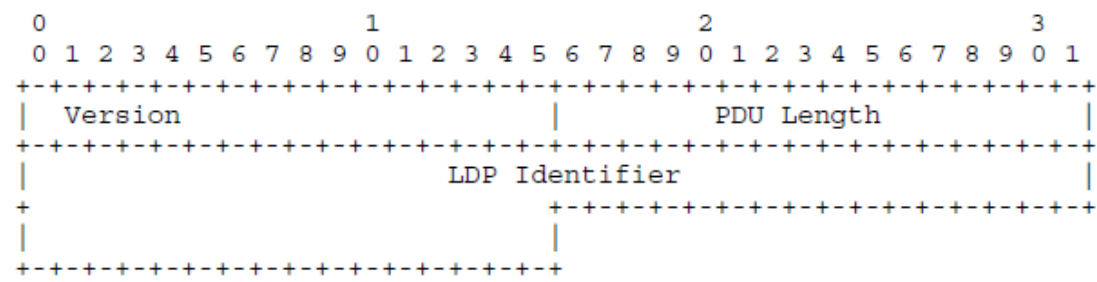


图 10-4 LDP 的 TLV 格式

- Version: 版本号，1 字节。当前 LDP 版本号为 1;
- PDU Length: LDP 报文的总长度，2 字节。单位是字节;
- LDP ID: LDP 标识符，6 字节。前 4 字节是全局唯一的 LSR ID，后 2 字节是标签空间标识符。对于全局标签空间，这个字段为 0。

TLV 编码

LDP 采用 TLV(Type-Length-Value)方式封装 LDP 消息中的参数。LDP TLV 格式如下:

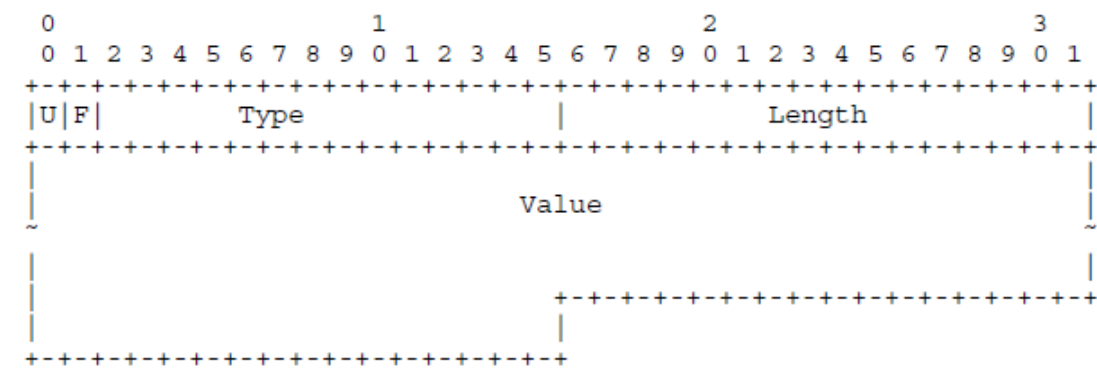


图 10-5 LDP 的 TLV 格式

- U bit: 未知标志，1bit。如果 U 标志置 0，LSR 必须向发送报文的源 LSR 发送通知，并忽视整个消息；如果 U 标志置 1，忽视此 TLV 参数，正常解析其它 TLV 参数;
- F bit: 转发未知 TLV 标志，1。此标志只适用于未知 TLV 且 U bit 置 1 的 LDP 消息。如果 F 标志置 0，不再转发未知 TLV 参数；如果 F 标志置 1，转发未知 TLV 参数;
- Type: 类型，14bits;
- Length: 长度，1 字节。TLV 数值字段的长度;
- Value: 数值字段。其长度由参数 Length 定义;
- TLV 的数值字段还可以包含 TLV 参数，也就是说 TLV 可以嵌套。TLV 的第一个字节病不要求字节对齐。

目前定义的 TLV 类型:

图 10-6 LDP 报文中 TLV 类型

TLV	Type
FEC	0x0100
Address List	0x0101

Hop Count	0x0103
Path Vector	0x0104
Generic Label	0x0200
ATM Label	0x0201
Frame Relay Label	0x0202
Status	0x0300
Extended Status	0x0301
Returned PDU	0x0302
Returned Message	0x0303
Common Hello Parameters	0x0400
IPv4 Transport Address	0x0401
Configuration Sequence Number	0x0402
IPv6 Transport Address	0x0403
Common Session Parameters	0x0500
ATM Session Parameters	0x0501
Frame Relay Session Parameters	0x0502
Label Request Message ID	0x0600
Vendor-Private	0x3E00- 0x3EFF
Experimental	0x3F00- 0x3FFF

10.2.1.3 LDP 标签管理

在 MPLS 体系中，由下游 LSR 决定将标签分配给特定 FEC，再通知上游 LSR。即，标签由下游指定，标签的分配按从下游到上游的方向。

标签分发模式（Label Advertisement Mode）

在 MPLS 域中，分组从上游 LSR 经过标签交换打上下游 LSR 的标签转发至下游 LSR。下游 LSR 为 FEC 分配的标签只在本地和其上游 LSR 之间有意义，下游 LSR 分配的标签必需传播到上游 LSR。对于标签从下游 LSR 到上游 LSR 的传播方式问题，MPLS 定义了两种标签分发模式：

- ☞ 下游按需模式 DoD（Downstream On Demand）：对于一个特定的 FEC，LSR 从上游获得标签请求消息之后才进行标签分配与分发；
- ☞ 下游自主模式 DU（Downstream Unsolicited）：对于一个特定的 FEC，LSR 无须从上游获得标签请求消息即进行标签分配与分发，会主动发送标签映射消息，通知上游 LSR。

这两种模式可以混合适用，LSR 的每个接口可以独立配置为这两种标签分发模式之一。在初始化时，上游 LSR 和下游 LSR 必须交换标签分发模式信息，就标签分发模式达成一致，否则 LSP 无法正常建立。

如下图 2-3 为 LDP 标签分发示意图。

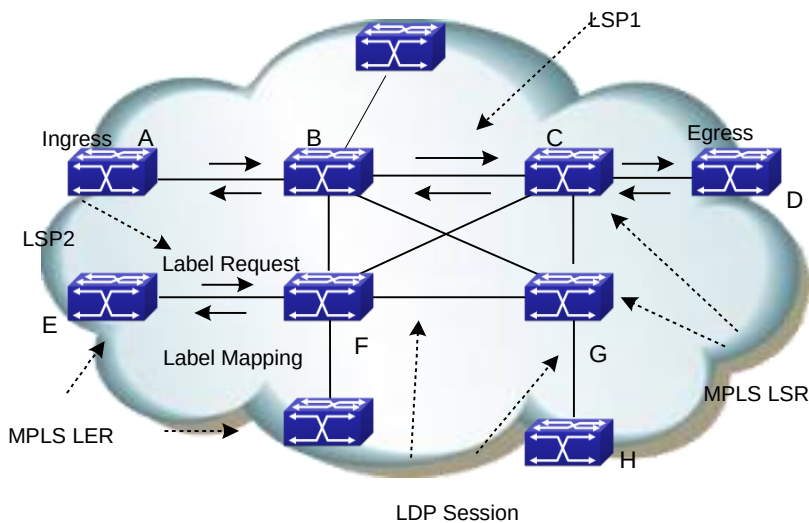


图 10-7 标签分发过程

例如，图 2-3 中的 LSP1 上，LSR B 为 LSR C 的上游 LSR，LSR C 则是 LSR B 的下游 LSR。

标签的分发过程有两种模式，主要区别在于标签映射的发布是上游请求（DoD）还是下游主动发布（DU）。

下面分别详细描述这两种模式的标签分发过程：

(1) DoD（downstream-on-demand）模式分发过程

上游 LSR 向下游 LSR 发送标签请求消息（Label Request Message），其中包含 FEC 的描述信息。下游 LSR 为此 FEC 分配标签，并将映射的标签通过标签映射消息（Label Mapping Message）反馈给上游 LSR。

下游 LSR 何时反馈标签映射消息，取决于该 LSR 采用的标签分配控制方式。

- 1) 采用 Ordered 方式时，只有收到它的下游返回的标签映射消息后，才向其上游发送标签映射消息；
- 2) 采用 Independent 方式时，不管有没有收到它的下游返回的标签映射消息，都立即向其上游发送标签映射消息。

上游 LSR 一般是根据其路由表中的信息来选择下游 LSR。在图 2-3 中，LSP1 沿途的 LSR 都采用 Ordered 方式，LSP2 上的 LSR F 则采用 Independent 方式。

(2) DU（downstream unsolicited）模式分发过程

下游 LSR 在 LDP 会话建立成功后，主动向其上游 LSR 发布标签映射消息。上游 LSR 保存标签映射信息，并根据保有模式来处理收到的标签映射信息。

标签控制模式（Label Distribution Control Mode）

在 MPLS 域中，LSR 之间通过交换标签建立一条从输入节点（ingress router）到输出节点（egress router）的标签转发路径 LSP。LSP 是沿着 MPLS 域中 IGP 建立的路由转发路径建立的，只有一条完整的路径才对分组转发有意义。建立 LSP 的过程，也就是 LSR 分发标签的过程；控制 LSP 的建立，就是控制 LSR 的标签分发。对于下游 LSR 何时向上游 LSR 分发标签的问题，MPLS 定义了两种 LSP 控制模式：

- ☞ 有序模式(Ordered)：对于 LSR 上某个 FEC 的标签映射，只有当该 LSR 已经具有此 FEC 下一跳的标签映射消息、或者该 LSR 就是此 FEC 的出口节点时，该 LSR

才可以向上游发送此 FEC 的标签映射。一个流的标签分发从这个 FEC 流所属的出口节点开始，由下游向上游逐级绑定，这样可以保证整个网络内标记与流的映射完整一致。有序方式提供了更好的环路预防能力。

- ☞ 独立模式(Independent): LSR 不必等待 FEC 下一跳的标签就可以把 FEC 标签分发给它的 peer。LSR 可以在任意时间向与它连接的 LSR 通告标签映射。这种方式可能导致在收到下游标签之前就向上游发布了标签。独立方式可以提高 LSP 的建立和汇聚速度。

LSR 作为出口节点的条件:

- ☞ FEC 引用了 LSR 本身的地址;
- ☞ FEC 的下一跳路由器位于标签交换网之外;
- ☞ FEC 的单元经过路由区域, 例如另一个 OSPF SUMMARY 域, 或者 OSPF、BGP 的另一个自治系统。

标签保有模式 (Label Retention Mode)

标签保有(retention)模式指 LSR 对收到的、但目前暂时用不到的标签到 FEC 映射的处理方式。在下游主动标签分发模式下, 上游 LSR 可能接收到下游 LSR 通知的大量<FEC、标签>映射组, 只有当映射组中的 FEC 是上游 LSR 本地 FEC 的下一跳时, 这个映射组才对标签转发有意义。对于如何处理暂时没有转发意义的映射组, MPLS 定义了两种标签保有模式:

- ☞ 保守模式(Conservative): 对于从邻居 LSR 收到的标签映射, 只有当邻居 LSR 是自己的下一跳时才保留。保守模式的优点是: 只创建和维护对数据转发有意义的标签, 在标签空间有限的情况下(ATM 交换), 非常重要。
- ☞ 自由模式(Liberal): 对于从邻居 LSR 收到的标签映射, 无论邻居 LSR 是不是自己的下一跳都保留。自由模式的优点是: 处理路由变化的开销非常小; 缺点是: 通告和维护了许多暂时无效的标签。

使用自由标签保持方式, LSR 能够迅速适应路由变化; 而使用保守标签保持方式, LSR 可以分配和保存较少的标签数量。保守标签保持方式通常与 DoD 方式一起, 用于对于标签空间有限的 LSR。

标签交换中的几个基本概念

- ☞ NHLFE (Next Hop Label Forwarding Entry): 下一跳标签转发项。用于描述对标签执行的操作, 包括入栈 (Push) 操作和交换 (Swap) 操作。
- ☞ FTN (FEC to NHLFE map): 在 Ingress 节点将转发等价类 FEC 映射到 NHLFE 的过程。
- ☞ ILM (Incoming Label Map): 入标签映射。对于接收的标签分组, LSR 将标签映射到 NHLFE 的过程。

标签交换的过程

入口 LER (Ingress) 将进入网络的分组划分成转发等价类 FEC。属于相同 FEC 的分组在 MPLS 域中将经过相同的路径, 即使用同一条 LSP。LSR 对到来的 FEC 分组分配一个标签, 然后从相应的接口转发出去。

标签交换的具体操作描述如下:

- ☞ LSP 沿途的所有 LSR 都先建立 ILM, 入标签是根据 ILM 表项进行映射的;
- ☞ 对于接收到的标签分组, LSR 将标签映射到 NHLFE;
- ☞ LSR 只需根据标签从标签信息表中找到相应的 NHLFE, 用新的标签替换原来的标签, 然后对标签分组继续转发。

10.2.1.4 LDP 会话

按照先后顺序，LDP 会话建立过程主要包括以下四个阶段：

- ☞ 发现阶段
- ☞ 会话建立与维护
- ☞ LSP 建立
- ☞ 会话撤销

发现阶段

在这一阶段，希望建立会话的 LSR 向相邻 LSR 周期性地发送 Hello 消息，通知相邻节点自己的存在。在基本发现机制下，通过这一过程，LSR 可以自动发现它的 LDP 对等体，而无需进行手工配置。LDP 有两种发现机制：

- ☞ 基本发现机制

基本发现机制用于发现本地的 LDP 对等体，即通过链路层直接相连的 LSR，建立本地 LDP 会话。这种方式下，LSR 周期性以 UDP 报文形式从接口发送 LDP 链路 Hello 消息(LDP Link Hello)，发往标识“子网内所有路由器”的组播地址。

LDP 链路 Hello 消息带有接口的 LDP 标识符及其他相关信息，如果 LSR 在某个接口收到了 LDP 链路 Hello 消息，则表明在该接口（链路层）存在 LDP 对等体。

- ☞ 扩展发现机制

扩展发现机制用于发现远端的 LDP 对等体，即不通过链路层直接相连的 LSR，建立远端 LDP 会话。这种方式下，LSR 周期性以 UDP 报文形式向指定的 IP 地址发送 LDP 目标 Hello 消息（LDP Targeted Hello）。

LDP 目标 Hello 消息带有 LSR 的 LDP 标识符及其他相关信息，如果 LSR 收到 LDP 目标 Hello 消息，则表明在网络层存在 LDP 对等体。

会话建立与维护

在第一阶段发现 LDP 对等体之后，LSR 开始建立会话。这一过程又可分为两步：

- ☞ 建立传输层连接，即，在 LSR 之间建立 TCP 连接；
- ☞ 随后对 LSR 之间的会话进行初始化，协商会话中涉及的各种参数，如 LDP 版本、标签分发方式、定时器值、标签空间等。协商成功之后，就在 LSR 之间建立了会话连接

会话建立后，通过不断地发送 Keepalive 消息来维护这个会话。

LSP 建立

LSP 的建立过程实际就是将 FEC 和标签进行映射，并将这种映射通告 LSP 上相邻 LSR。这个过程是通过 LDP 实现的，以 DoD 模式为例，主要步骤如下：

- (1) 当网络的路由改变时，如果有一个边缘节点发现自己的路由表中出现了新的目的地址，并且这一地址不属于任何现有的 FEC，则该边缘节点需要为这一目的地址建立一个新的 FEC。边缘 LSR 决定该 FEC 将要使用的路由，向其下游 LSR 发起标签请求消息，并指明是要为哪个 FEC 申请标签；
- (2) 收到标签请求消息的下游 LSR 记录这一请求消息，根据本地的路由表找出对应该 FEC 的下一跳，继续向下游 LSR 发出标签请求消息；
- (3) 当标签请求消息到达目的节点或 MPLS 网络的出口节点时，如果此节点尚有可供分配的标签，并且判定上述标签请求消息合法，则该节点为 FEC 分配标签，并向上游发出标签映射消息，标签映射消息中包含分配的标签等信息；
- (4) 收到标签映射消息的 LSR 检查本地存储的标签请求消息状态。对于某一 FEC 的标

签映射消息，如果数据库中记录了相应的标签请求消息，LSR 将为该 FEC 进行标签分配，并在其标签转发表中增加相应的条目，然后向上游 LSR 发送标签映射消息；

- (5) 当入口 LSR 收到标签映射消息时，它也需要在标签转发表中增加相应的条目。这时，就完成了 LSP 的建立。

会话撤销

LDP 通过检测 Hello 消息来维持邻接关系。同时 LDP 也通过检测 Keepalive 消息来维持会话的连接，如果一定时间内没有收到 Keepalive 消息，则 LDP 会话就会关闭连接。

每个 LDP 会话可以包含一个或者多个 Hello 邻接。LDP 依靠周期性的 Hellos 报文维持 Hello 邻接关系。如果长时间没有接收到 LDP Discovery Hellos 报文，LDP 就关闭 Hello 邻接。当关闭 LDP 会话中的最后一个 Hello 邻接时，LDP 就发送通知消息，并关闭传输连接。

10.2.1.5 LDP 环路检测

在 MPLS 域中建立 LSP 也要防止产生环路，LDP 环路检测机制可以检测 LSP 环路的出现，并避免发生环路。

如果对 MPLS 域进行环路检测，则必须在所有 LSR 上都配置环路检测。但在建立 LDP 会话时，并不要求双方的环路检测配置一致。

LDP 环路检测有两种方式：

最大跳数

指标签消息(包括标签映射和标签请求)经过的 LSR 的数量。当 LSR 传播包含跳数参数的标签信息时，首先把跳数加 1。当跳数达到配置的最大数值时，即认为出现环路，LSP 建立失败。如果跳数为 0，就表示跳数未知，这类消息，保持跳数为 0 不变。最大跳数的默认值为 255。

路径向量

在传递标签映射（或者标签请求）的消息中记录路径信息，每经过一跳，LSR 就会检查自己的 LSR ID 是否在此记录中。在以下条件之一时认为出现环路，LSP 建立失败：

- ☞ 路径向量记录表中已有本 LSR 的记录；
- ☞ 路径的跳数超过设置的最大值。

如果记录中没有自己的 LSR ID，就会将其添加到该记录中。路径向量的最大值与最大跳数相等。

10.2.2 LDP 配置

LDP 配置任务序列：

1. 全局启动 MPLS（必选）
2. 启动 LDP 协议（必选）
 - (1) 启动 LDP 模块/关闭 LDP 模块
 - (2) 在接口上启动 label-switching/关闭 label-switching
 - (3) 在接口上启动 LDP 模块/关闭 LDP 模块
3. 配置 LDP 协议参数（可选）
 - (1) 配置 LDP 标签管理模式
 - 1) 配置 LDP 标签保有模式

- 2) 配置 LDP 标签分发模式
- 3) 配置 LDP 控制模式
- (2) 配置 LDP 环路检测
 - 1) 配置 LDP 是否进行环路检测
 - 2) 配置 LDP 环路检测时允许的跳数
- (3) 配置 LDP 指定对等体
- (4) 配置 LDP 协议其它参数
 - 1) 配置 LDP 各计时器老化时间或时间间隔
 - 2) 配置 LDP 路由器 ID
 - 3) 配置 LDP 用于 TCP 连接的接口地址
 - 4) 配置 LDP 是否通过组播 Hello 发现对等体
 - 5) 配置 LDP 是否引入 BGP 路由
 - 6) 配置 LDP 是否使能标签合并能力
 - 7) 设置 LDP 是否传播标签释放消息
 - 8) 设置 LDP 在申请标签被拒绝后是否重试
 - 9) 配置 LDP 接收指定目标的 Hello
- 4. 清除 LDP 的连接或邻接关系

1. 全局启动 MPLS（必选）

命令	解释
全局配置模式	
mpls enable no mpls enable	启动 mpls 协议；本命令的 no 操作关闭 MPLS 协议。

2. 启动 LDP 协议（必选）

在 DCNOS 中运行 LDP 协议的基本配置很简单，通常只需打开 LDP 开关、并在准备运行 LDP 的接口配置启动 LDP 即可，但应当注意的是，配置启动 LDP 的接口应当使能标签交换。

命令	解释
全局配置模式	
router ldp no router ldp	打开/关闭 LDP 协议；缺省情况下未使能 LDP。
接口配置模式	
label-switching no label-switching	使能标签交换功能；缺省情况下未使能标签交换。
ldp {enable disable}	接口上启动/关闭 LDP 协议；接口缺省情况下未使能 LDP。

3. 配置 LDP 协议参数（可选）

- (1) 配置 LDP 标签管理模式
 - 1) 配置 LDP 标签保有模式

- 2) 配置 LDP 标签分发模式
- 3) 配置 LDP 控制模式

命令	解释
路由器配置模式	
label-retention-mode {conservative liberal}	配置全局标签保有模式，可选择保守模式或自由模式，缺省情况下是自由模式 liberal 。
advertisement-mode {downstream-on-demand downstream-unsolicited}	配置全局标签分发模式，可选择下游主动或下游按需。该模式与另两种模式有搭配关系，标签分发模式改变会同时改变标签保有模式和全局标签路径控制模式。缺省情况下标签分发模式是下游主动模式 downstream-unsolicited 。
control-mode {ordered independent}	配置全局标签路径控制模式，可选择有序模式和独立模式。缺省情况下是独立模式 independent 。
接口配置模式	
ldp label-retention-mode {conservative liberal}	配置所在接口的标签保有模式。缺省情况与全局模式相同。如果配置与全局模式不符，以接口上的配置为准。
ldp advertisement-mode {downstream-on-demand downstream-unsolicited}	配置所在接口的标签分发模式。缺省情况与全局模式相同。如果配置与全局模式不符，以接口上的配置为准。

(2) 配置 LDP 环路检测

- 1) 配置 LDP 是否进行环路检测
- 2) 配置 LDP 环路检测时允许的跳数

命令	解释
路由器配置模式	
loop-detection no loop-detection	设定 LDP 进行环路检测。其 NO 形式取消进行环路检测。
loop-detection-count <count> no loop-detection-count <count>	设定 LDP 进行环路检测时允许的跳数，缺省为 255，其 no 形式恢复为缺省值。

(3) 配置 LDP 指定对等体

命令	解释
路由器配置模式	
targeted-peer <ip-addr> no targeted-peer <ip-addr>	配置 LDP 指定目的地的远端邻居或远端对等体。

(4) 配置 LDP 协议其它参数

- 1) 配置 LDP 各计时器老化时间或时间间隔
- 2) 配置 LDP 路由器 ID

- 3) 配置 LDP 用于 TCP 连接的接口地址
- 4) 配置 LDP 是否通过组播 Hello 发现对等体
- 5) 配置 LDP 是否引入 BGP 路由
- 6) 配置 LDP 是否使能标签合并能力
- 7) 配置 LDP 是否传播标签释放消息
- 8) 配置 LDP 在申请标签被拒绝后是否重试
- 9) 配置 LDP 接收指定目标的 Hello

命令	解释
路由器配置模式	
keepalive-interval <interval> no keepalive-interval <interval>	配置 LDP 保活报文发送的时间间隔, 缺省为 10 秒, 其 no 形式恢复为缺省值。
keepalive-timeout <time-val> no keepalive-timeout <time-val>	配置 LDP 会话的连接保持时间, 缺省为 30 秒, 其 no 形式恢复为缺省值。
Hello-interval <Hello-interval> no Hello-interval <Hello-interval>	配置 LDP 发送组播 HELLO 的时间间隔, 缺省为 5 秒, 其 no 形式恢复为缺省值。
hold-time <hold-time > no hold-time <hold-time >	配置 LDP 组播对等体的保持时间, 缺省为 15 秒, 其 no 形式恢复为缺省值。
targeted-peer-Hello-interval <Hello-interval> no targeted-peer-Hello-interval <Hello-interval>	配置 LDP 发送指定目的地的 HELLO 的时间间隔, 缺省为 15 秒, 其 no 形式恢复为缺省值。
targeted-peer-hold-time <hold-time> no targeted-peer-hold-time <hold-time>	配置 LDP 指定目的地对等体的保持时间, 缺省为 45 秒, 其 no 形式恢复为缺省值。
接口配置模式	
ldp keepalive-interval <interval> no ldp keepalive-interval <interval>	针对指定接口配置 LDP 保活报文发送的时间间隔; 其 no 形式恢复为缺省值。
ldp keepalive-timeout <time-val> no ldp keepalive-timeout <time-val>	针对指定接口配置 LDP 会话的连接保持时间; 其 no 形式恢复为缺省值。
ldp Hello-interval <Hello-interval> no ldp Hello-interval <Hello-interval>	针对指定接口配置 LDP 发送组播 HELLO 的时间间隔; 其 no 形式恢复为缺省值。
ldp hold-time <hold-time> no ldp hold-time <hold-time>	针对指定接口配置 LDP 组播对等体的保持时间; 其 no 形式恢复为缺省值。
ldp targeted-peer-Hello-interval <Hello-interval> no ldp targeted-peer-Hello-interval <Hello-interval>	针对指定接口配置 LDP 发送指定目的地的 HELLO 的时间间隔; 其 no 形式恢复为缺省值。

ldp targeted-peer-hold-time <hold-time> no ldp targeted-peer-hold-time <hold-time>	针对指定接口配置 LDP 指定目的地对等体的保持时间; 其 no 形式恢复为缺省值。
路由器配置模式	
router-id <ip-addr> no router-id <ip-addr>	配置 LDP 使用的路由器 ID,缺省为自动获得,其 no 形式取消手动配置的路由器 ID,并自动获得一个满足条件的接口的 IP 地址作为路由器 ID。
transport-address <ip-addr> no transport-address <ip-addr>	配置 LDP 用于进行 TCP 连接的 IP 地址,需要注意的是,该地址必须是主 VRF 上一个环回接口的地址。其 no 形式取消手动的配置而让 LDP 自动选择进行 TCP 连接的地址。
multicast-Hellos no multicast-Hellos	配置 LDP 使用组播 HELLO 来发现对等体,其 no 形式配置不使用组播 HELLO 来发现对等体。缺省使用组播 HELLO。
import-bgp-routes no import-bgp-routes	配置 LDP 引入 BGP 路由,其 no 形式取消引入 BGP 路由。缺省不引入 BGP 路由。
global-merge-capability {merge-capable non-merge-capable} no global-merge-capability {merge-capable non-merge-capable}	配置使能或不使能全局标签合并能力,缺省为使能,其 no 形式恢复为缺省值。
propagate-release no propagate-release	配置标签释放向对等体传播,其 no 形式配置不传播标签的释放,缺省不进行传播。
request-retry no request-retry	配置标签请求失败后进行 5 次重试,缺省不进行重试,其 no 形式取消重试的配置。
request-retry-timeout <time-val> no request-retry-timeout <time-val>	配置进行重试的时间间隔,缺省为 5 秒,其 no 形式恢复为缺省值。
targeted-peer-Hello-receipt no targeted-peer-Hello-receipt	配置 LDP 接收指定目的地的 HELLO,即使本端并未配置该对等体,缺省不接收这样的 HELLO。其 no 形式恢复为不接收。注意如果配置了指定的 LDP 对等体,那么一定要配置 targeted-peer-Hello-receipt

4. 清除 LDP 的连接或邻接关系 (可选)

命令	解释
特权模式	
clear ldp adjacency {<ip-addr> *}	清除指定的 LDP 邻接关系，“*”表示所有邻接关系。
clear ldp session {<ip-addr> *}	清除指定的 LDP 会话，“*”表示所有会话。

10.2.3 LDP 典型案例

LDP 的一些设计是为了适应不同网络环境进行的，在典型的以太网环境的配置十分简单，另外由于硬件系统的发展，特别是三层交换机的普遍应用，纯粹的 MPLS 网络已经没有什么明显的价值，而更多的应用在 MPLS VPN 环境下。

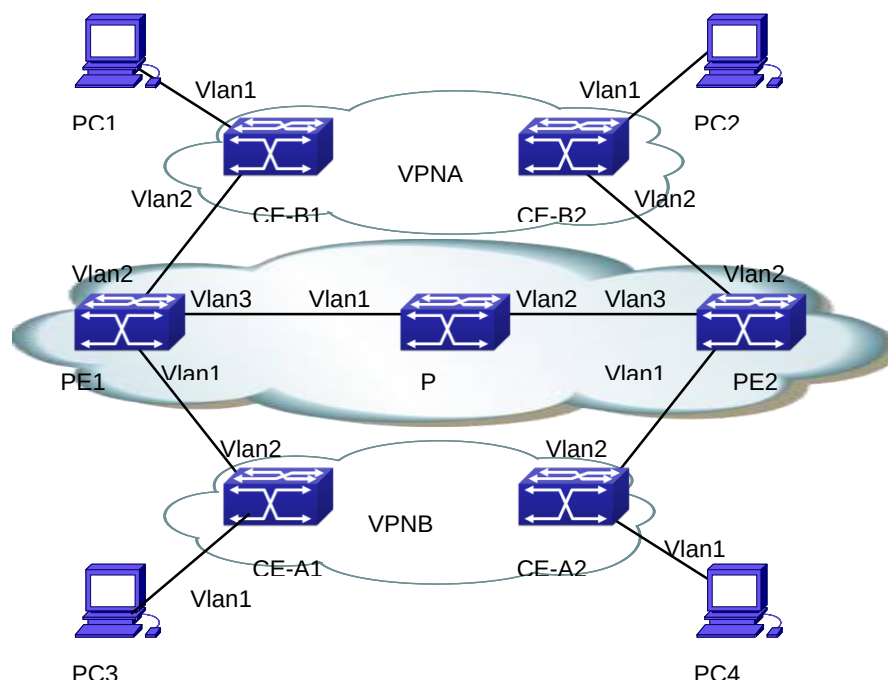


图 10-8 MPLS VPN 典型案例

如图 2-4 是一个典型的 MPLS VPN 案例，其中 PE1、P、PE2 组成了公网区域，也就是使用 MPLS 交换的区域，CE-A1 与 CE-A2 组成了 VPN-A，CE-B1 与 CE-B2 组成了 VPN-B。两个 VPN 都通过公网的标签交换通信，在公网区域，需要配置 LDP 协议来进行标签进行标签的分发、分配。为了保证路由的可达性，我们使用 OSPF 协议公布他们各自的路由。

PE1 的 LDP 配置如下：

```

PE1#config
PE1(config)#mpls enable
PE1(config)# router ldp
PE1(config-router)#exit
PE1(config)#interface vlan 3
PE1(config-if-Vlan3)#ip address 202.200.1.2 255.255.255.0

```

```
PE1(config-if-Vlan3)#ldp enable
PE1(config-if-Vlan3)#label-switching
PE1(config-if-Vlan3)#exit
PE1(config)#router ospf
PE1(config-router)#network 200.200.1.1/32 area 0
PE1(config-router)#network 202.200.1.0/24 area 0
PE1(config-router)#exit
```

P 交换机的 LDP 配置如下：

```
P#config
P(config)#mpls enable
P(config)# router ldp
P(config-router)#exit
P(config)#interface vlan 1
P(config-if-Vlan1)#ip address 202.200.1.1 255.255.255.0
P(config-if-Vlan1)#ldp enable
P(config-if-Vlan1)#label-switching
P(config-if-Vlan1)#exit
P(config)#interface vlan 2
P(config-if-Vlan2)#ip address 202.200.2.1 255.255.255.0
P(config-if-Vlan2)#ldp enable
P(config-if-Vlan2)#label-switching
P(config-if-Vlan2)#exit
P(config)#router ospf
P(config-router)#network 202.200.1.0/24 area 0
P(config-router)#network 202.200.2.0/24 area 0
P(config-router)#exit
```

PE1 的 LDP 配置如下：

```
PE2#config
PE2(config)#mpls enable
PE2(config)# router ldp
PE2(config-router)#exit
PE2(config)#interface vlan 3
PE1(config-if-Vlan3)#ip address 202.200.2.2 255.255.255.0
PE2(config-if-Vlan3)#ldp enable
PE2(config-if-Vlan3)#label-switching
PE2(config-if-Vlan3)#exit
PE2(config)#router ospf
PE2(config-router)#network 200.200.1.2/32 area 0
PE2(config-router)#network 202.200.2.0/24 area 0
PE2(config-router)#exit
```

关于 BGP 的配置，详见 BGP VPN 典型案例。

10.2.4 LDP 排错帮助

在配置、使用 LDP 协议时，可能会由于物理连接、配置错误等原因导致 LDP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证系统全局启动了 LDP 并且在活动接口上正确启动了 LDP，注意接口使能 LDP 的前提是全局使能了 MPLS；
- ☞ 其次，在连接正确后，使用 `show ldp interface` 命令查看是否相应的接口上 LDP 已经正常启动，如果确信接口上正确配置了启动 LDP，但显示却没有看到，可能是接口没有处于 UP 状态或没有配置接口的标签交换（命令 `label-switching`）；
- ☞ 然后，确认相邻的接口处在同一网段，并通过 `show ldp adjacency` 查看 LDP 是否可以正常发现对等体并建立邻接关系。如果没有发现对等体或者没有成功建立邻接关系，可能是接口没有处在同一网段或者本端和远端邻居之一配置了不发送组播 HELLO。另外在建立 TCP 连接时，缺省使用的地址是 LSR ID，因此，必须确认将 LSR ID 的路由发布给对端。
- ☞ 接着，通过 `show ldp session` 查看 LDP 会话是否建立并处于 operational 状态，只有处于 operational 状态的 LDP 会话才能交换消息。如果 LDP 会话不能建立，使用 `show ldp` 命令查看正在双方使用的用于 TCP 连接的地址并查看路由表，确认对端路由可达。
- ☞ 最后，在上述过程都正确的情况下，使用 `show ldp fec` 命令查看 LDP 引入了哪些路由及这些路由的相关信息，也可以通过 `show mpls ftn` 和 `show mpls ilm` 命令查看其建立的表项。
- ☞ 另外，如果是配置了指定目的地的 LDP 对等体，则要确保对端也配置了以本机为目的地的 LDP 对等体或者配置了可以接收指定目的地的 HELLO。并且双方指定的地址要路由可达。

10.3 MPLS VPN

10.3.1 BGP/MPLS VPN 简介

10.3.1.1 BGP/MPLS VPN 网络结构

BGP/MPLS VPN 是提供商 VPN 解决方案中一种基于 PE 的 L3VPN 技术，它使用 BGP 在服务提供商骨干网上发布 VPN 路由，使用 MPLS 在服务提供商骨干网上转发 VPN 报文。

BGP/MPLS VPN 组网方式灵活、可扩展性好，并能够方便地支持 MPLS QoS 和 MPLS TE，因此得到越来越多的应用。

BGP/MPLS VPN 模型由三部分组成：CE、PE 和 P。

- ☞ P 路由器（Provide Router）：供应商路由器。位于 MPLS 域的内部。可以基于标签交换快速转发 MPLS 数据流。P 路由器接收 MPLS 报文，交换标签后，输出 MPLS 报文。
- ☞ PE 路由器（Provide Edge Router）：供应商边界路由器。位于 MPLS 域的边界，

用于转换 IP 报文和 MPLS 报文。PE 路由器接收 IP 报文，压入 MPLS 标签后，输出 MPLS 报文；并且接收 MPLS 报文，弹出标签之后，输出 IP 报文。PE 路由器上，与其它 P 路由器或者 PE 路由器连接的端口被称为“公网端口”，配置公网 IP 地址；与 CE 路由器连接的端口被称为“私网端口”，配置私网 IP 地址。

- ☞ CE 路由器（Customer Edge Router）：用户边界路由器。位于用户 IP 域边界，直接和 PE 路由器连接，用于汇聚用户数据，并把用户 IP 域的路由信息转发到 PE 路由器。

图 3-1 是一个 BGP/MPLS VPN 组网方案的示意图：

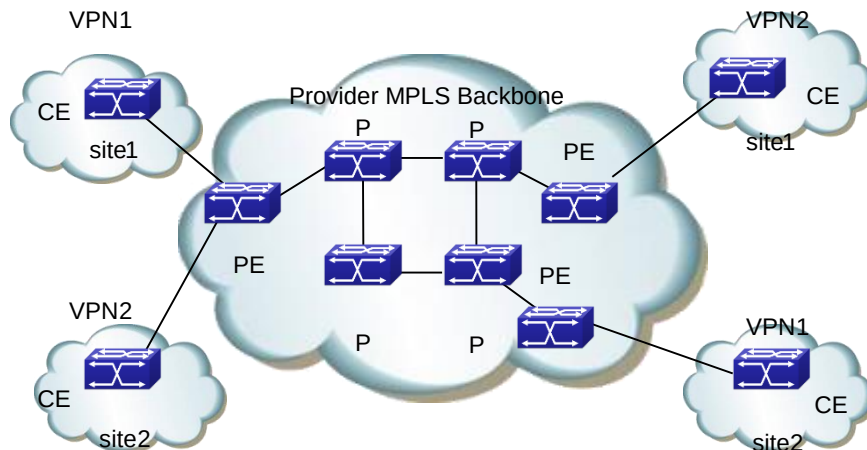


图 10-9 BGP/MPLS VPN 组网

CE 和 PE 的划分主要是根据 SP 与用户的管理范围，CE 和 PE 是两者管理范围的边界。

CE 设备通常是一台路由器，当 CE 与直接相连的 PE 建立邻接关系后，CE 把本站点的 VPN 路由发布给 PE，并从 PE 学到远端 VPN 的路由。CE 与 PE 之间使用 BGP/IGP 交换路由信息，也可以使用静态路由。

PE 从 CE 学到 CE 本地的 VPN 路由信息后，通过 BGP 与其它 PE 交换 VPN 路由信息。PE 路由器只维护与它直接相连的 VPN 的路由信息，不维护服务提供商网络中的所有 VPN 路由。

P 路由器只维护到 PE 的路由，不需要了解任何 VPN 路由信息。

当在 MPLS 骨干网上传输 VPN 流量时，入口 PE 做为 Ingress LSR（Label Switch Router），出口 PE 做为 Egress LSR，P 路由器则做为 Transit LSR。

10.3.1.2 BGP/MPLS VPN 基本概念

Site

在介绍 VPN 时经常会提到“site”，site（站点）的含义可以从下述几个方面理解：

- ☞ site 是指相互之间具备 IP 连通性的一组 IP 系统，并且，这组 IP 系统的 IP 连通性不需通过服务提供商网络实现；
- ☞ site 的划分是根据设备的拓扑关系，而不是地理位置，尽管在大多数情况下一个 site 中的设备地理位置相邻；
- ☞ 一个 site 中的设备可以属于多个 VPN，换言之，一个 site 可以属于多个 VPN；

- ☞ site 通过 CE 连接到服务提供商网络，一个 site 可以包含多个 CE，但一个 CE 只属于一个 site。

对于多个连接到同一服务提供商网络的 sites，通过制定策略，可以将它们划分为不同的集合（set），只有属于相同集合的 sites 之间才能通过服务提供商网络互访，这种集合就是 VPN。

VRF

VRF 即 VPN Routing & Forwarding Instance，VPN 路由转发实例，由 VPN IP 路由表和 VPN IP 转发表(转发表包含了 MPLS 封装信息)组成，是实现 MPLS VPN 分组转发的核心表项。一个 VPN 有自己独立的一个 VRF，不同 VPN 的 VRF 地址空间可以重叠。

在 MPLS VPN 网络中一个 PE/P 路由器通常包含有多个独立运作的 VRF。**地址空间重叠**

VPN 是一种私有网络，不同的 VPN 独立管理自己使用的地址范围，也称为地址空间（Address Space）。

不同 VPN 的地址空间可能会在一定范围内重合，比如，VPN1 和 VPN2 都使用了 10.110.10.0/24 网段的地址，这就发生了地址空间重叠（Overlapping Address Spaces）。

VPN 实例

在 MPLS VPN 中，不同 VPN 之间的路由隔离通过 VPN 实例（VPN-instance）实现。

PE 为每个直接相连的 Site 建立并维护专门的 VPN 实例。VPN 实例中包含对应 Site 的 VPN 成员关系和路由规则。如果一个 Site 中的用户同时属于多个 VPN，则该 Site 的 VPN 实例中将包括所有这些 VPN 的信息。

为保证 VPN 数据的独立性和安全性，PE 上每个 VPN 实例都有相对独立的路由表和 LFIB（Label Forwarding Information Base，标签转发表）。

具体来说，VPN 实例中的信息包括：标签转发表、IP 路由表、与 VPN 实例绑定的接口、以及 VPN 实例的管理信息（包括 RD、路由过滤策略、成员接口列表等）。

VPN-IPv4 地址

传统 BGP 无法正确处理地址空间重叠的 VPN 的路由。假设 VPN1 和 VPN2 都使用了 10.110.10.0/24 网段的地址，并各自发布了一条去往此网段的路由，BGP 将只会选择其中一条路由，从而导致去往另一个 VPN 的路由丢失。

PE 路由器之间使用 MP-BGP 来发布 VPN 路由，并使用 VPN-IPv4 地址族来解决上述问题。

VPN-IPv4 地址共有 12 个字节，包括 8 字节的 RD（Route Distinguisher）和 4 字节的 IPv4 地址前缀，如图 3-2 所示：

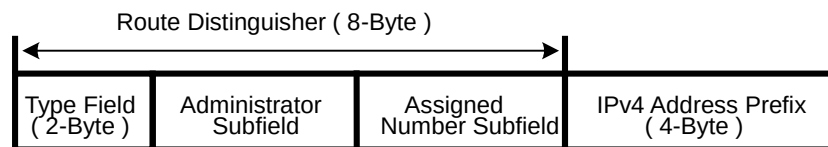


图 10-10 VPN-IPv4 地址结构

PE 从 CE 接收到普通 IPv4 路由后，需要将这些私网 VPN 路由发布给对端 PE。私网路由的独立性是通过为这些路由附加 RD 实现的。

SP 可以独立地分配 RD，但必须保证 RD 的全局唯一性。这样，即使来自不同服务提供商的 VPN 使用了同样的 IPv4 地址空间，PE 路由器也可以向各 VPN 发布不同的路由。

建议为 PE 上每个 VPN 实例配置专门的 RD, 以保证到达同一 CE 的路由都使用相同的 RD。RD 为 0 的 VPN-IPv4 地址相当于全局唯一的 IPv4 地址。

RD 的作用是添加到一个特定的 IPv4 前缀, 使之成为全局唯一的 VPN IPv4 前缀。

RD 或者是与自治系统号 (ASN) 相关的, 在这种情况下, RD 是由一个自治系统号和一个任意的数组成; 或者是与 IP 地址相关的, 在这种情况下, RD 是由一个 IP 地址和一个任意的数组成。

RD 有两种格式, 通过 2 字节的 Type 字段区分:

- ☞ Type 为 0 时, Administrator 子字段占 2 字节, Assigned Number 子字段占 4 字节, 格式为: 16bits 自治系统号:32bits 用户自定义数字。例如: 100:1
- ☞ Type 为 1 时, Administrator 子字段占 4 字节, Assigned Number 子字段占 2 字节, 格式为: 32bits IPv4 地址:16bits 用户自定义数字。例如: 172.1.1.1:1

为保证 RD 的全局唯一性, 建议不要将 Administrator 子字段的值设置为私有 AS 号或私有 IP 地址。

VPN Target 属性

BGP/MPLS VPN 使用 32 位的 BGP 扩展团体属性——VPN Target (也称为 Route Target) 来控制 VPN 路由信息的发布。

PE 路由器上的 VPN 实例有两类 VPN Target 属性:

- ☞ Export Target 属性: 本地 PE 在把从与自己直接相连的 Site 学到的 VPN-IPv4 路由发布给其它 PE 前, 为这些路由设置 Export Target 属性;
- ☞ Import Target 属性: PE 在接收到其它 PE 路由器发布的 VPN-IPv4 路由时, 检查其 Export Target 属性, 只有当此属性与 PE 上 VPN 实例的 Import Target 属性匹配时, 才把路由加入到相应的 VPN 路由表中。

也就是说, VPN Target 属性定义了, 一条 VPN-IPv4 路由可以为哪些 Site 所接收, PE 路由器可以接收哪些 Site 发送来的路由。

与 RD 类似, VPN Target 也有两种格式:

- ☞ 16bits 自治系统号:32bits 用户自定义数字, 例如: 100:1。
- ☞ 32bits IPv4 地址:16bits 用户自定义数字, 例如: 172.1.1.1:1。

MP-BGP

MP-BGP (Multiprotocol extensions for BGP-4) 在 PE 路由器之间传播 VPN 组成信息和路由。MP-BGP 向下兼容, 既可以支持传统的 IPv4 地址族, 又可以支持其他地址族 (比如 VPN-IPv4 地址族)。使用 MP-BGP 确保 VPN 的私网路由只在 VPN 内发布, 并实现 MPLS VPN 成员间的通信。

路由策略 (Routing Policy)

在通过入口、出口扩展团体来控制 VPN 路由发布的基础上, 如果需要更精确地控制 VPN 路由的引入和发布, 可以使用入方向或出方向路由策略。

入方向路由策略根据路由的 VPN Target 属性进一步过滤可引入到 VPN 实例的路由, 它可以拒绝接收引入列表中的团体选定的路由, 而出方向路由策略则可以拒绝发布输出列表中的团体选定的路由。

VPN 实例创建完成后, 可以选择是否需要配置入方向或出方向路由策略。

隧道策略 (Tunneling Policy)

隧道策略用于选择给特定 VPN 实例的报文使用的隧道。

隧道策略是可选配的, VPN 实例创建完成后, 就可以配置隧道策略。缺省情况下, 选择 LSP 做为隧道, 不进行负载分担 (负载分担条数为 1)。另外, 隧道策略只在同一 AS 域

内生效。

10.3.1.3 BGP/MPLS VPN 报文转发

在基本 L3VPN 应用中（不包括跨域的情况），VPN 报文转发采用两层标签方式：

- ☞ 第一层（外层）标签在骨干网内部进行交换，指示从 PE 到对端 PE 的一条 LSP。
- ☞ 第二层（内层）标签在从对端 PE 到达 CE 时使用，指示报文应被送到哪个 Site，或者更具体一些，到达哪一个 CE。这样，对端 PE 根据内层标签可以找到转发报文的接口。

特殊情况下，属于同一个 VPN 的两个 Site 连接到同一个 PE，这种情况下只需要知道如何到达对端 CE。

以图 3-3 为例，说明 VPN 报文的转发：

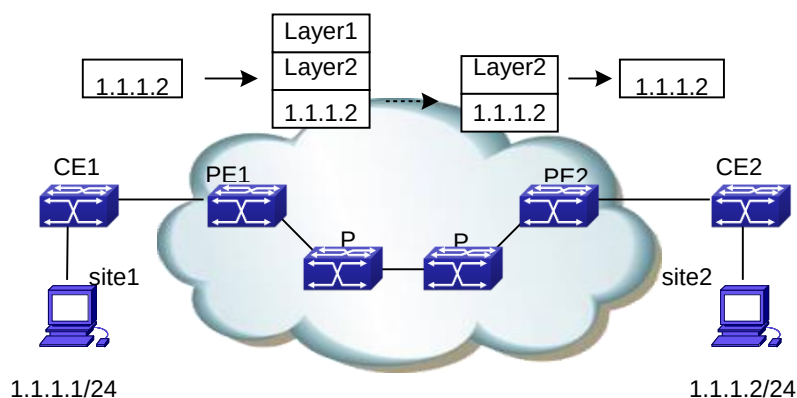


图 10-11 VPN 报文转发示意图

- (1) Site1 发出一个目的地址为 1.1.1.2 的 IP 报文，由 CE1 将报文发送至 PE1。
- (2) PE1 根据报文到达的接口及目的地址查找 VPN-instance 表项，匹配后将报文转发出去，同时打上内层和外层两个标签。
- (3) MPLS 网络利用报文的外层标签，将报文传送到 PE2（报文在到达 PE2 前一跳时已经被剥离外层标签，仅含内层标签）。
- (4) PE2 根据内层标签和目的地址查找 VPN-instance 表项，确定报文的出接口，将报文转发至 CE2。
- (5) CE2 根据正常的 IP 转发过程将报文传送到目的地。

10.3.1.4 BGP/MPLS VPN 组网方案

在 BGP/MPLS VPN 网络中，通过 VPN Target 属性来控制 VPN 路由信息在各 site 之间的发布和接收。VPN Export Target 和 Import Target 的设置相互独立，并且都可以设置多个值，能够实现灵活的 VPN 访问控制，从而实现多种 VPN 组网方案。

基本的 VPN

最简单的情况下，一个 VPN 中的所有用户形成闭合用户群，相互之间能够进行流量转发，VPN 中的用户不能与任何本 VPN 以外的用户通信。

对于这种组网，需要为每个 VPN 分配一个 VPN Target，作为该 VPN 的 Export Target 和 Import Target，并且，此 VPN Target 不能被其他 VPN 使用。

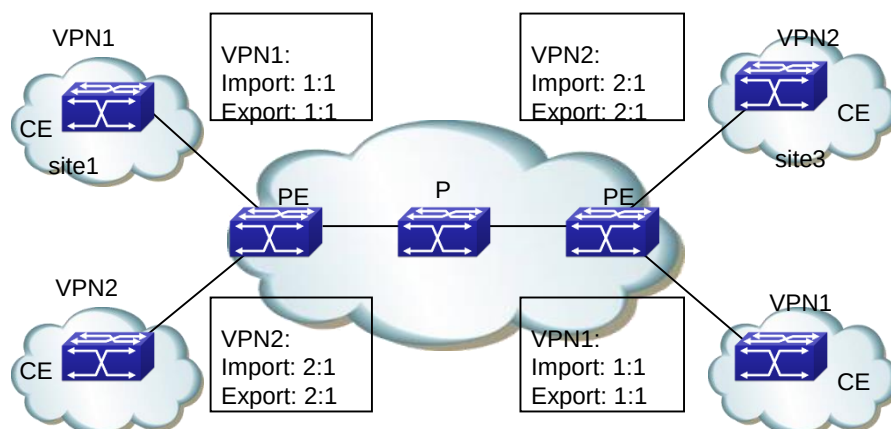


图 10-12 基本的 VPN 组网方案

在图 3-4 中，PE 上为 VPN1 分配的 VPN Target 值为 100:1，为 VPN2 分配的 VPN Target 值为 200:1。VPN1 的两个 site 之间可以互访，VPN2 的两个 site 之间也可以互访，但 VPN1 和 VPN2 的 site 之间不能互访。

Hub&Spoke VPN

如果希望在 VPN 中设置中心访问控制设备，其它用户的互访都通过中心访问控制设备进行，可以使用 Hub&Spoke 组网方案，从而实现中心设备对两端设备之间的互访进行监控和过滤等功能。

对于这种组网，需要设置两个 VPN Target，一个表示“Hub”，另一个表示“Spoke”。各 site 在 PE 上的 VPN 实例的 VPN Target 设置规则为：

- ☞ 连接 Spoke 站点的（Spoke-PE）：Export Target 为“Spoke”，Import Target 为“Hub”；
- ☞ 连接 Hub 站点的（Hub-PE）：Hub-PE 上需要使用两个接口或子接口，一个用于接收 Spoke-PE 发来的路由，其 VPN 实例的 Import Target 为“Spoke”；另一个用于向 Spoke-PE 发布路由，其 VPN 实例的 Export Target 为“Hub”。

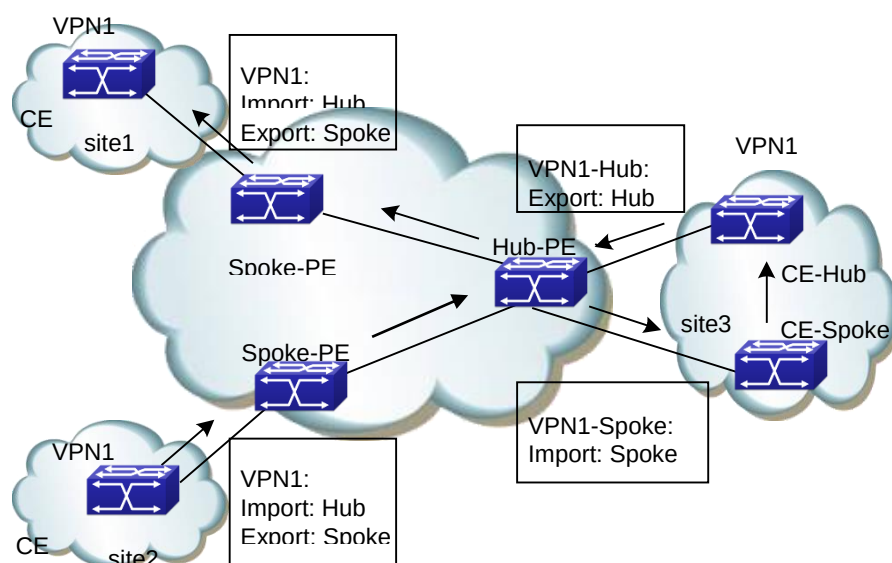


图 10-13 Hub&Spoke 组网方案

在图 3-5 中，Spoke 站点之间的通信通过 Hub 站点进行（图中箭头所示为 site2 的路由向 site1 的发布过程）：

- ☞ Hub-PE 能够接收所有 Spoke-PE 发布的 VPN-IPv4 路由；
- ☞ Hub-PE 发布的 VPN-IPv4 路由能够为所有 Spoke-PE 接收；
- ☞ Hub-PE 将从 Spoke-PE 学到的路由发布给其他 Spoke-PE，因此，Spoke 站点之间可以通过 Hub 站点互访。
- ☞ 任意 Spoke-PE 的 Import Target 属性不与其它 Spoke-PE 的 Export Target 属性相同。因此，任意两个 Spoke-PE 之间不直接发布 VPN-IPv4 路由，Spoke 站点之间不能直接互访。

Extranet VPN

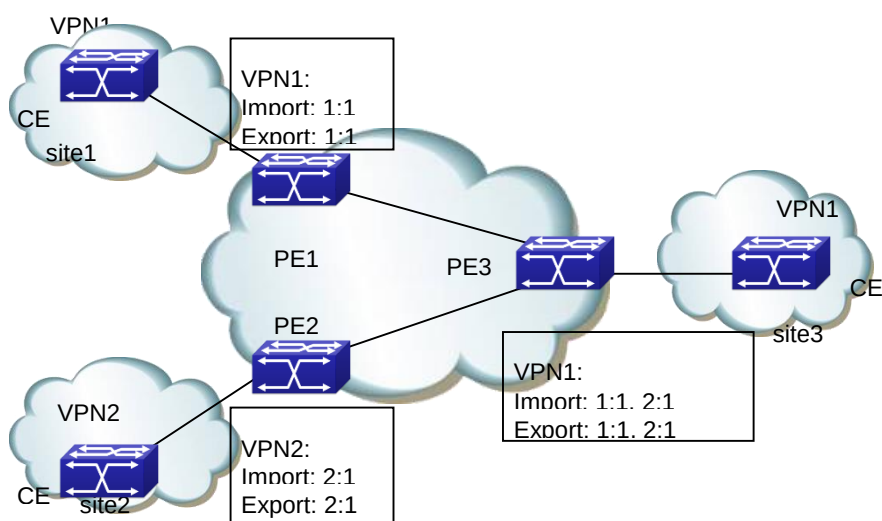


图 10-14 Extranet 组网方案

如果一个 VPN 用户希望提供部分本 VPN 的站点资源给非本 VPN 的用户访问，可以使用 Extranet 组网方案。

对于这种组网，如果某个 VPN 需要访问共享站点，则该 VPN 的 Export Target 必须包含在共享站点的 VPN 实例的 Import Target 中，而其 Import Target 必须包含在共享站点 VPN 实例的 Export Target 中。

在图 3-6 中，VPN1 的 site3 能够被 VPN1 和 VPN2 访问：

- ☞ PE3 能够接受 PE1 和 PE2 发布的 VPN-IPv4 路由；
- ☞ PE3 发布的 VPN-IPv4 路由能够为 PE1 和 PE2 接受；
- ☞ 基于以上两点，VPN1 的 site1 和 site3 之间能够互访，VPN2 的 site2 和 VPN1 的 site3 之间能够互访；

PE3 不把从 PE1 接收的 VPN-IPv4 路由发布给 PE2，也不把从 PE2 接收的 VPN-IPv4 路由发布给 PE1（IBGP 邻居学来的条目是不会再发送给别的 IBGP 邻居），因此，VPN1 的 site1 和 VPN2 的 site2 之间不能互访。

10.3.1.5 BGP/MPLS VPN 路由发布

在基本 BGP/MPLS VPN 组网中，VPN 路由信息的发布涉及 CE 和 PE，P 路由器只维护骨干网的路由，不需要了解任何 VPN 路由信息。PE 路由器也只维护与它直接相连的 VPN 的路由信息，不维护所有 VPN 路由。因此，BGP/MPLS VPN 网络具有良好的可扩展性。

VPN 路由信息的发布过程包括三部分：本地 CE 到入口 PE、入口 PE 到出口 PE、出口 PE 到远端 CE。完成这三部分后，本地 CE 与远端 CE 之间将建立可达路由，VPN 私网路由信息能够在骨干网上发布。

下面分别对这三部分进行介绍。

本地 CE 到入口 PE 的路由信息交换

CE 与直接相连的 PE 建立邻接关系后，把本站点的 VPN 路由发布给 PE。

CE 与 PE 之间可以使用静态路由、RIP、OSPF、IS-IS 或 EBGp。无论使用哪种路由协议，CE 发布给 PE 的都是标准的 IPv4 路由。

入口 PE 到出口 PE 的路由信息交换

PE 从 CE 学到 VPN 路由信息后, 为这些标准 IPv4 路由增加 RD 和 VPN Target 属性, 形成 VPN-IPv4 路由, 存放为 CE 创建的 VPN 实例中。

入口 PE 通过 MP-BGP 把 VPN-IPv4 路由发布给出口 PE。出口 PE 根据 VPN-IPv4 路由的 Export Target 属性与自己维护的 VPN 实例的 Import Target, 决定是否将该路由加入到 VPN 实例的路由表。

PE 之间通过 IGP 来保证内部的连通性。

出口 PE 到远端 CE 的路由信息交换

远端 CE 有多种方式可以从出口 PE 学习 VPN 路由, 包括静态路由、RIP、OSPF、IS-IS 和 EBGP, 与本地 CE 到入口 PE 的路由信息交换相同。

10.3.1.6 跨域 VPN 简介

实际组网应用中, 某用户一个 VPN 的多个 Site 可能会连接到使用不同 AS 号的多个服务提供商, 或者连接到一个服务提供商的多个 AS。这种 VPN 跨越多个自治系统的应用方式被称为跨域 VPN (Multi-AS VPN)。RFC 2547 中提出了三种跨域 VPN 解决方案, 分别是:

- ☞ VRF-to-VRF: ASBR 间使用 VRF 接口建立 EBGP 邻居并管理 VPN 路由, 也称为 Inter-Provider Option A;
- ☞ EBGP Redistribution of labeled VPN-IPv4 routes: ASBR 间通过 MP-EBGP 发布标签 VPN-IPv4 路由, 也称为 Inter-Provider Option B;
- ☞ Multihop EBGP redistribution of labeled VPN-IPv4 routes: PE 间通过 Multi-hop MP-EBGP 发布标签 VPN-IPv4 路由, 也称为 Inter-Provider Option C。

目前我们能够实现第一种 VRF-to-VRF 的跨域 VPN 解决方案。

跨域 VPN 解决方案

如图 3-7 所示, 在这种方式下, 两个 AS 的 PE 路由器直接相连, PE 路由器同时也是各自所在自治系统的边界路由器 ASBR。作为 ASBR 的 PE 之间通过 VRF 接口相连, 并使其 IMPORT 所有本侧想要对方所知的 RT, EXPORT 所有本侧想从对方获得的 RT, 并在双方的 VRF 上建立 EBGP 的连接, 从而使各自服务的 CE 象在同一 AS 内一样达到相互通信和隔离的目的。此时两个 PE 都把对方作为自己的 CE 设备对待, 报文在 AS 内部作为 VPN 报文, 采用两层标签转发方式; 在 ASBR 之间则采用普通 IP 转发方式。

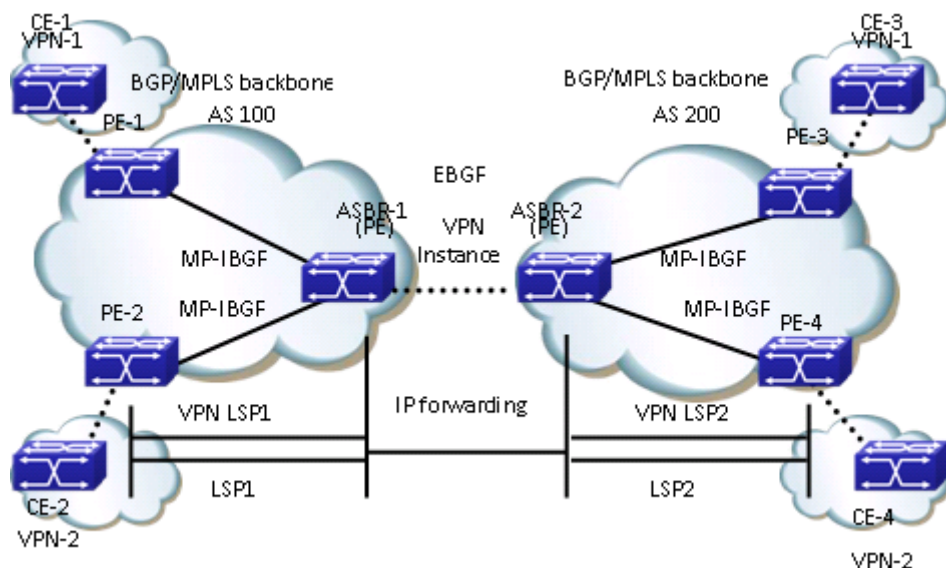


图 10-15 跨域 VPN 组网图

- 使用这种方式实现跨域 VPN 的优点是实现简单：两个作为 ASBR 的 PE 之间不需要为跨域进行特殊配置。
- 缺点是可扩展性差：作为 ASBR 的 PE 需要管理所有 VPN 路由，为每个 VPN 创建 VPN 实例。这将导致 PE 上的 VPN-IPv4 路由数量过于庞大。

10.3.2 BGP MPLS VPN 配置

BGP MPLS VPN 配置任务序列：

1. 全局启动 MPLS（必选）
2. 配置 VPN 实例（必选）
 - (1) 创建 VPN 实例，并进入 VPN 实例视图
 - (2) 配置 VPN 实例的 RD
 - (3) 配置 VPN 实例的 RT
 - (4) 配置 VPN 实例与接口关联
3. 配置基本 MPLS VPN（必选）
 - (1) 配置 PE-CE 间使用 EBG
 - 1) 配置对端 PE 为公网 VPNv4 邻居
 - 2) 进入 BGP-VPN 实例视图
 - 3) 将 CE 配置为 VPN 私网邻居
 - 4) 发布本端的私网路由
 - (2) 配置 PE-CE 间使用 OSPF
 - 1) 配置对端 PE 为公网 VPNv4 邻居
 - 2) 创建 PE-CE 间的 OSPF 实例，并进入 Router OSPF 视图
 - 3) 使能 PE-CE 间网段的 OSPF
 - 4) 配置重发布 BGP 路由
 - 5) 进入 BGP-VPN 实例视图
 - 6) 配置重发布 OSPF 路由

- 7) 发布本端的私网路由
- (3) 配置 PE-CE 间使用 RIP
 - 1) 配置对端 PE 为公网 VPNv4 邻居
 - 2) 进入 RIP VPN 实例视图
 - 3) 使能 PE-CE 间网段的 RIP
 - 4) 配置重发布 BGP 路由
 - 5) 进入 BGP-VPN 实例视图
 - 6) 配置重发布 RIP 路由
 - 7) 发布本端的私网路由
- (4) 配置 PE-CE 间使用静态路由
 - 1) 配置对端 PE 为公网 VPNv4 邻居
 - 2) 配置静态 VPN 路由
 - 3) 进入 BGP-VPN 实例视图
 - 4) 配置重发布静态路由
 - 5) 发布本端的私网路由

1. 全局启动 MPLS（必选）

命令	解释
全局配置模式	
mpls enable no mpls enable	启动 mpls 协议；本命令的 no 操作关闭 MPLS 协议。

2. 配置 VPN 实例（必选）

- (1) 创建 VPN 实例，并进入 VPN 实例视图
- (2) 配置 VPN 实例的 RD
- (3) 配置 VPN 实例的 RT
- (4) 配置 VPN 实例与接口关联

命令	解释
全局配置模式	
ip vrf <vrf-name> no ip vrf <vrf-name>	创建 VPN 实例；缺省情况下未创建 VPN 实例。
VRF 配置模式	
rd <ASN:nn_or_IP-address:nn> no rd <ASN:nn_or_IP-address:nn>	配置 VPN 实例的 RD。缺省情况下未创建 RD。
route-target {import export both} <rt-value> no route-target {import export both} <rt-value>	配置 VPN 实例的 RT。
接口配置模式	

ip vrf forwarding <vrf-name> no ip vrf forwarding <vrf-name>	配置 VPN 实例与接口关联。
ip address <ip-address> <mask> no ip address <ip-address> <mask>	配置 PE-CE 间直连接口的私网 IP 地址。

3 配置基本 MPLS VPN（必选）

(1) 配置 PE-CE 间使用 EBGp（必选）

- 1) 配置对端 PE 为公网 VPNv4 邻居
- 2) 进入 BGP-VPN 实例视图
- 3) 将 CE 配置为 VPN 私网邻居
- 4) 发布本端的私网路由

命令	解释
BGP 协议配置模式	
neighbor <ip-address> remote-as <as-num>	配置对端 PE 为公网 VPNv4 邻居。建议选取 loopback 接口建立公网 PE 间的 BGP 邻居。
neighbor <ip-address> update-source <as-num>	指定建立邻居的本地 loopback 接口。
进入 BGP-VPNv4 视图	
address-family vpnv4 [unicast]	创建 BGP 协议 VPNv4。缺省情况下未创建任何 VPNv4。
neighbor <ip-address> active no neighbor <ip-address> active	激活 VPNv4 下的邻居。缺省情况下 VPNv4 视图下所有邻居都是未激活的。
BGP 协议配置模式	
address-family ipv4 {unicast multicast vrf <vrf-name>} no address-family ipv4 {unicast multicast vrf <vrf-name>}	创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4。
BGP-VPN 实例视图	
neighbor <ip-address> remote-as <as-num> no neighbor <ip-address> remote-as <as-num>	将 CE 配置为 VPN 私网邻居。缺省情况下未配置任何私网邻居。
neighbor <ip-address> active no neighbor <ip-address> active	激活 IPv4 下的邻居。缺省情况下所有邻居都是激活的。
redistribute {connected ospf rip static} no redistribute {connected ospf rip static}	配置重发布直连路由和其他协议的路由。缺省情况下未重发布任何路由。

(2) 配置 PE-CE 间使用 OSPF

- 1) 配置对端 PE 为公网 VPNv4 邻居
- 2) 创建 PE-CE 间的 OSPF 实例，并进入 OSPF VPN 实例视图
- 3) 使能 PE-CE 间网段的 OSPF
- 4) 配置重发布 BGP 路由
- 5) 进入 BGP-VPN 实例视图
- 6) 配置重发布 OSPF 路由
- 7) 发布本端的私网路由

命令	解释
BGP 协议配置模式	
neighbor <ip-address> remote-as <as-num>	配置对端 PE 为公网 VPNv4 邻居。建议选取 loopback 接口建立公网 PE 间的 BGP 邻居。
neighbor <ip-address> update-source <as-num>	指定建立邻居的本地 loopback 接口。
进入 BGP-VPNv4 视图	
address-family vpnv4 [unicast]	创建 BGP 协议 VPNv4。缺省情况下未创建任何 VPNv4。
neighbor <ip-address> active no neighbor <ip-address> active	可选 激活 VPNv4 下的邻居。缺省情况下所有邻居都是激活的。
全局配置模式	
router ospf [<process_id> [<vrf-nam>]] no router ospf [<process_id> [<vrf-nam>]]	可选 创建 PE-CE 间的 OSPF 实例，并进入 OSPF VPN 实例视图
OSPF VPN 实例视图	
network {<network> <mask> <network>/<prefix>} area <area_id> no network {<network> <mask> <network>/<prefix>} area <area_id>	可选 使能 PE-CE 间网段的 OSPF。缺省情况下未使能任何网段

redistribute {bgp connected static rip kernel} [metric-type {1 2}] [tag <tag>] [metric <cost_value>] [router-map <WORD>] no redistribute {bgp connected static rip kernel} [metric-type {1 2}] [tag <tag>] [metric <cost_value>] [router-map <WORD>]	可选 配置重发布 BGP 路由。缺省情况下未重发布任何路由
BGP 协议配置模式	
address-family ipv4 {unicast multicast vrf <vrf-nam>} no address-family ipv4 {unicast multicast vrf <vrf-nam>}	可选 创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4
BGP-VPN 实例视图	
redistribute {connected ospf rip static} no redistribute {connected ospf rip static}	可选 配置重发布直连路由和其他协议路由。缺省情况下未重发布任何路由。

(3) 配置 PE-CE 间使用 RIP

- 1) 配置对端 PE 为公网 VPNv4 邻居
- 2) 进入 RIP VPN 实例视图
- 3) 使能 PE-CE 间网段的 RIP
- 4) 配置重发布 BGP 路由
- 5) 进入 BGP-VPN 实例视图
- 6) 配置重发布 RIP 路由
- 7) 发布本端的私网路由

命令	解释
BGP 协议配置模式	
neighbor <ip-address> remote-as <as-num>	配置对端 PE 为公网 VPNv4 邻居。建议选取 loopback 接口建立公网 PE 间的 BGP 邻居。
neighbor <ip-address> update-source <as-num>	指定建立邻居的本地 loopback 接口
进入 BGP-VPNv4 视图	
address-family vpnv4 [unicast]	创建 BGP 协议 VPNv4。缺省情况下未创建任何 VPNv4。

neighbor <ip-address> active no neighbor <ip-address> active	可选 激活 VPNv4 下的邻居。缺省情况下所有邻居都是激活的。
RIP 协议配置模式	
address-family ipv4 vrf <vrf-name> no address-family ipv4 vrf <vrf-name>	可选 创建 PE-CE 间的 RIP IPv4 协议族, 并进入 RIP VPN 实例视图
RIP VPN 实例视图	
network {A.B.C.D/M ifname vlan<id> loopback <1-1024> } no network {A.B.C.D/M ifname vlan<id> loopback <1-1024> }	可选 使能 PE-CE 间网段的 RIP
redistribute {kernel connected static ospf isis bgp} [metric <value>] [route-map <word>] no redistribute {kernel connected static ospf isis bgp} [metric <value>] [route-map <word>]	可选 配置重发布 BGP 路由。缺省情况下未重发布任何路由
BGP 协议配置模式	
address-family ipv4 {unicast multicast vrf <vrf-name>} no address-family ipv4 {unicast multicast vrf <vrf-name>}	可选 创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4
BGP-VPN 实例视图	
redistribute {connected ospf rip static} no redistribute {connected ospf rip static}	可选 配置重发布直连路由和其他协议路由。缺省情况下未重发布任何路由。

(4) 配置 PE-CE 间使用静态路由

- 1) 配置对端 PE 为公网 VPNv4 邻居
- 2) 配置静态 VPN 路由
- 3) 进入 BGP-VPN 实例视图
- 4) 配置重发布静态路由
- 5) 发布本端的私网路由

命令	解释
BGP 协议配置模式	

neighbor <ip-address> remote-as <as-num>	配置对端 PE 为公网 VPNv4 邻居。建议选取 loopback 接口建立公网 PE 间的 BGP 邻居。
neighbor <ip-address> update-source <as-num>	指定建立邻居的本地 loopback 接口。
进入 BGP-VPNv4 视图	
address-family vpnv4 [unicast]	创建 BGP 协议 VPNv4。缺省情况下未创建任何 VPNv4。
neighbor <ip-address> active no neighbor <ip-address> active	可选 激活 VPNv4 下的邻居。缺省情况下所有邻居都是激活的。
全局配置模式	
ip route vrf <vrf-name> {<ip-prefix> <mask> <ip-prefix/> <prefix-length>} {<gateway-address> null0} no ip route vrf <vrf-name> {<ip-prefix> <mask> <ip-prefix/> <prefix-length>} {<gateway-address> null0}	可选 手动配置 PE-CE 间的静态 VPN 路由。缺省情况下没有配置任何静态路由
BGP 协议配置模式	
address-family ipv4 {unicast multicast vrf <vrf-name>} no address-family ipv4 {unicast multicast vrf <vrf-name>}	可选 创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4
BGP-VPN 实例视图	
redistribute {connected ospf rip static} no redistribute {connected ospf rip static}	可选 配置重发布静态路由、直连路由和其他协议路由。缺省情况下未重发布任何路由。

10.3.3 BGP MPLS VPN 典型案例

10.3.3.1 PE-CE 间使用 EBGP 建立 BGP MPLS VPN

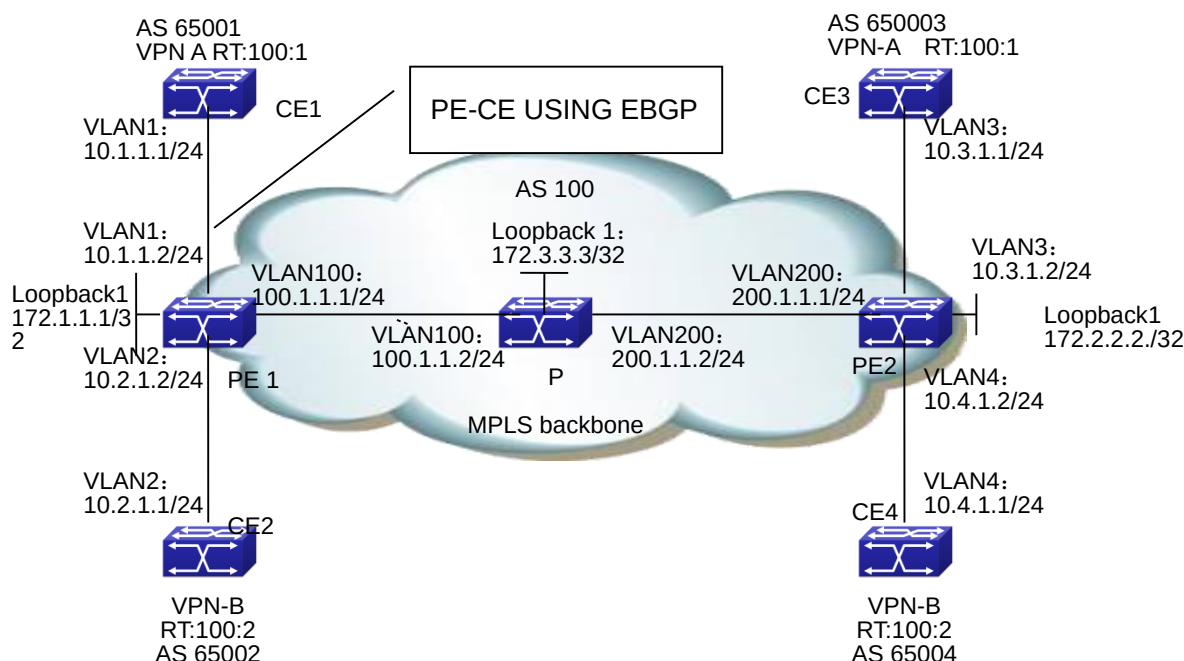


图 10-16 PE-CE 间使用 EBGP 建立 BGP MPLS VPN

CE1 的配置如下: (CE2~CE4 另外三个设备配置与 CE1 类似, 所以省略)

```
CE1#config
CE1(config)# interface vlan 1
CE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# router bgp 65001
CE1(config-router)#neighbor 10.1.1.2 remote-as 100
CE1(config-router)#redistribute connect
CE1(config-router)#exit
```

PE1 交换机的 MPLS BGP 配置如下:

(1) 配置 VPN 实例

```
PE1#config
PE1(config)#ip vrf vpna
PE1(config-vrf)#rd 100:1
PE1(config-vrf)#route-target both 100:1
PE1(config)#ip vrf vpnb
PE1(config-vrf)#rd 100:2
PE1(config-vrf)#route-target both 100:2
```

(2) 配置接口绑定 VPN 实例

```
PE1(config)# interface vlan 1
PE1(config-if-Vlan1)# ip vrf forwarding vpna
PE1(config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
```

```
PE1(config)# interface vlan 2
PE1(config-if-Vlan2)# ip vrf forwarding vpnb
PE1(config-if-Vlan2)#ip address 10.2.1.2 255.255.255.0
PE1(config-if-Vlan2)#exit
(3) 全局启动 MPLS 和 LDP
PE1(config)#mpls enable
PE1(config)#router ldp
PE1(config-router)#exit
(4) 配置接口以及使能 LDP
PE1(config)# interface loopback 1
PE1(config-if-Loopback1)# ip address 172.1.1.1 255.255.255.255
PE1(config-if-Loopback1)# exit
PE1(config)# interface vlan 100
PE1(config-if-Vlan100)#ip address 100.1.1.1 255.255.255.0
PE1(config-if-Vlan100)#label-switching
PE1(config-if-Vlan100) #ldp enable
PE1(config-if-Vlan100)#exit
(5) 启动 OSPF 协议传送内网路由
PE1(config)#router ospf
PE1(config-router)# ospf router-id 172.1.1.1
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)# redistribute connected
(6) 配置 BGP
PE1(config)# router bgp 100
PE1(config-router)#neighbor 172.2.2.2 remote-as 100
PE1(config-router)#neighbor 172.2.2.2 update-source 172.1.1.1
PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 172.2.2.2 activate
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpna
PE1(config-router-af)#neighbor 10.1.1.1 remote-as 65001
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpnb
PE1(config-router-af)#neighbor 10.2.1.1 remote-as 65002
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#exit
PE1(config-router)#exit
```

P 交换机的配置如下：

```
(1) 全局使能 MPLS 并且配置相关接口的 LDP
P#config
P(config)#mpls enable
P(config)#router ldp
```

```
P(config-router)#exit
P(config)# interface loopback 1
P(config-if-Loopback1)# ip address 172.3.3.3 255.255.255.255
P(config-if-Loopback1)# exit
P(config)#interface vlan 100
P(config-if-Vlan100)#ip address 100.1.1.2 255.255.255.0
P(config-if-Vlan100)#label-switching
P(config-if-Vlan100)#ldp enable
P(config-if-Vlan100)#exit
P(config)#interface vlan200
P(config-if-Vlan200)#ip address 200.1.1.2 255.255.255.0
P(config-if-Vlan200)#label-switching
P(config-if-Vlan200)#ldp enable
P(config-if-Vlan200)#exit
(2) 配置 OSPF 协议
P(config)#router ospf
P(config-router)# ospf router-id 172.3.3.3
P(config-router)# network 0.0.0.0/0 area 0
P(config-router)# redistribute connected
```

PE2 交换机的配置如下：

```
(1) 配置 VPN 实例
PE2#config
PE2(config)#ip vrf vpna
PE2(config-vrf)#rd 100:1
PE2(config-vrf)#route-target both 100:1
PE2(config)#ip vrf vpnb
PE2(config-vrf)#rd 100:2
PE2(config-vrf)#route-target both 100:2
(2) 配置接口绑定 VPN 实例
PE2(config)# interface vlan 3
PE2(config-if-Vlan3)# ip vrf forwarding vpna
PE2(config-if-Vlan3)#ip address 10.3.1.2 255.255.255.0
PE2(config-if-Vlan3)#exit
PE2(config)# interface vlan 4
PE2(config-if-Vlan4)# ip vrf forwarding vpnb
PE2(config-if-Vlan4)#ip address 10.4.1.2 255.255.255.0
PE2(config-if-Vlan4)#exit
(3) 全局启动 MPLS 和 LDP
PE2(config)#mpls enable
PE1(config)#router ldp
PE1(config-router)#exit
(4) 配置接口以及使能 LDP
PE2(config)# interface loopback 1
```



```
PE2(config-if-Loopback1)# ip address 172.2.2.2 255.255.255.255
PE2(config-if-Loopback1)# exit
PE2(config)# interface vlan 200
PE2(config-if-Vlan200)#ip address 200.1.1.1 255.255.255.0
PE2(config-if-Vlan200)#label-switching
PE2(config-if-Vlan200) #ldp enable
PE2(config-if-Vlan200)#exit
(5) 启动 OSPF 协议传送内网路由
PE2(config)#router ospf
PE2(config-router)# ospf router-id 172.2.2.2
PE2(config-router)# network 0.0.0.0/0 area 0
PE2(config-router)# redistribute connected
(6) 配置 BGP
PE2(config)# router bgp 100
PE2(config-router)#neighbor 172.1.1.1 remote-as 100
PE2(config-router)#neighbor 172.1.1.1 update-source 172.2.2.2
PE2(config-router)#address-family vpnv4
PE2(config-router-af)#neighbor 172.1.1.1 activate
PE2(config-router-af)#exit
PE2(config-router)# address-family ipv4 vrf vpna
PE2(config-router-af)#neighbor 10.3.1.1 remote-as 65003
PE2(config-router-af)#redistribute connected
PE2(config-router-af)#exit
PE2(config-router)# address-family ipv4 vrf vpnb
PE2(config-router-af)#neighbor 10.4.1.1 remote-as 65004
PE2(config-router-af)#redistribute connected
PE2(config-router-af)#exit
PE2(config-router)#exit
```

10.3.3.2 PE-CE 间使用 OSPF 建立 BGP MPLS VPN

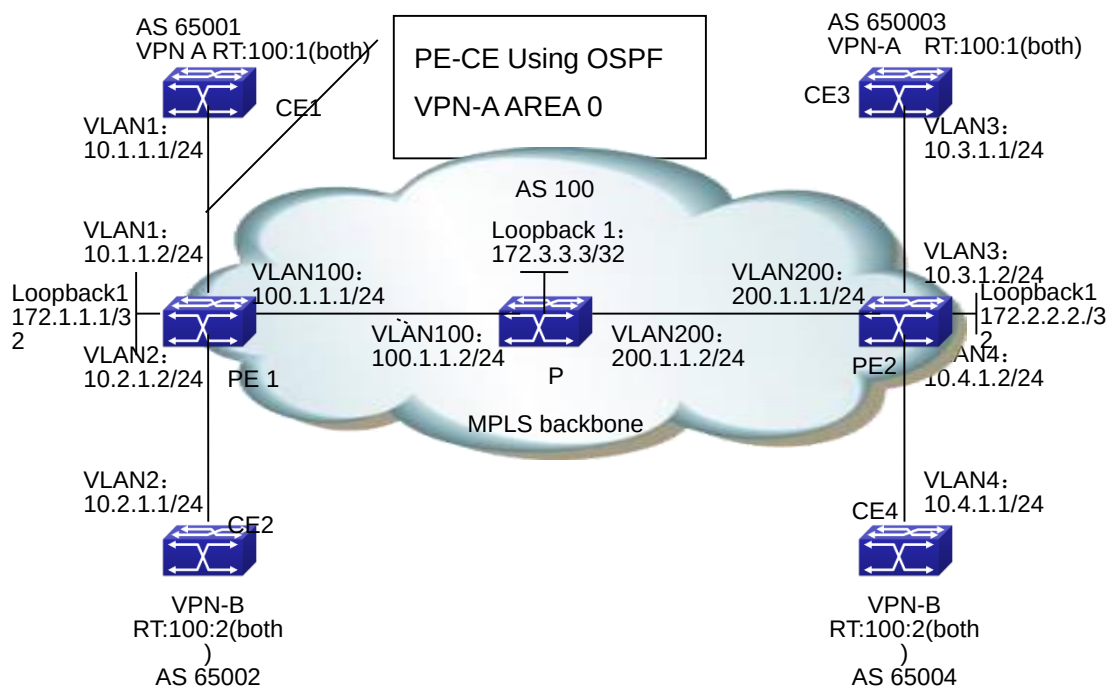


图 10-17 PE-CE 间使用 OSPF 建立 BGP MPLS VPN

CE1 的配置如下: (CE2~CE4 另外三个设备配置与 CE1 类似, 所以省略)

```
CE1#config
CE1(config)# interface vlan 1
CE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# router ospf
CE1(config-router)#network 0.0.0.0/0 area 0
CE1(config-router)#redistribute connect
CE1(config-router)#exit
```

PE1 交换机的 MPLS BGP 配置如下: (PE2 配置类似, 所以省略)

```
(1) 配置 VPN 实例
PE1#config
PE1(config)#ip vrf vpna
PE1(config-vrf)#rd 100:1
PE1(config-vrf)#route-target both 100:1
PE1(config)#ip vrf vpnb
PE1(config-vrf)#rd 100:2
PE1(config-vrf)#route-target both 100:2

(2) 配置接口绑定 VPN 实例
PE1(config)# interface vlan 1
PE1(config-if-Vlan1)# ip vrf forwarding vpna
PE1(config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
```

```
PE1(config)# interface vlan 2
PE1(config-if-Vlan2)# ip vrf forwarding vpnb
PE1(config-if-Vlan1)#ip address 10.2.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
(3) 全局启动 MPLS 和 LDP
PE1(config)#mpls enable
PE1(config)#router ldp
PE1(config-router)#exit
(4) 配置接口以及使能 LDP
PE1(config)# interface loopback 1
PE1(config-if-Loopback1)# ip address 172.1.1.1 255.255.255.255
PE1(config-if-Loopback1)# exit
PE1(config)# interface vlan 100
PE1(config-if-Vlan100)#ip address 100.1.1.1 255.255.255.0
PE1(config-if-Vlan100)# label-switching
PE1(config-if-Vlan100) #ldp enable
PE1(config-if-Vlan100)#exit
(5) 启动 OSPF 协议传送内网路由
PE1(config)#router ospf
PE1(config-router)# ospf router-id 172.1.1.1
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)# redistribute connected
PE1(config-router)#exit
(6) 配置 OSPF VRF 传送私网路由
PE1(config)#router ospf 1 vpna
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)#redistribute connected
PE1(config-router)#redistribute bgp
PE1(config-router)#exit
PE1(config)#router ospf 1 vpnb
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)#redistribute connected
PE1(config-router)#redistribute bgp
PE1(config-router)#exit
(7) 配置 BGP
PE1(config)# router bgp 100
PE1(config-router)#neighbor 172.2.2.2 remote-as 100
PE1(config-router)#neighbor 172.2.2.2 update-source 172.1.1.1
PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 172.2.2.2 activate
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpna
PE1(config-router-af)#redistribute connected
```

```
PE1(config-router-af)#redistribute ospf
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpnb
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute ospf
PE1(config-router-af)#exit
PE1(config-router)#exit
```

P 交换机的配置如下：

- (1) 全局使能 MPLS 并且配置相关接口的 LDP

```
P#config
P(config)#mpls enable
P(config)#router ldp
P(config-router)#exit
P(config)# interface loopback 1
P(config-if-Loopback1)# ip address 172.3.3.3 255.255.255.255
P(config-if-Loopback1)# exit
P(config)#interface vlan 100
P(config-if-Vlan100)#ip address 100.1.1.2 255.255.255.0
P(config-if-Vlan100)#label-switching
P(config-if-Vlan100)#ldp enable
P(config-if-Vlan100)#exit
P(config)#interface vlan200
P(config-if-Vlan200)#ip address 200.1.1.2 255.255.255.0
P(config-if-Vlan200)#label-switching
P(config-if-Vlan200)#ldp enable
P(config-if-Vlan200)#exit
(2) 配置 OSPF 协议
P(config)#router ospf
P(config-router)# ospf router-id 172.3.3.3
P(config-router)# network 0.0.0.0/0 area 0
P(config-router)# redistribute connected
```

10.3.3.3 PE-CE 间使用 RIP 建立 BGP MPLS VPN

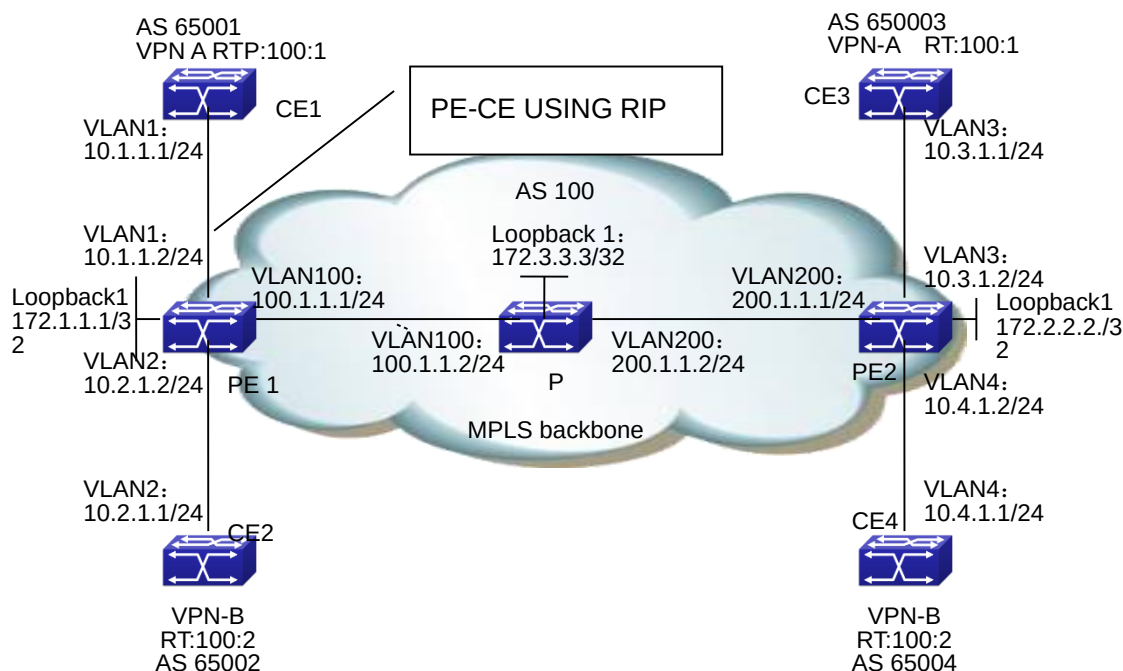


图 10-18 PE-CE 间使用 RIP 建立 BGP MPLS VPN

CE1 的配置如下: (CE2~CE4 另外三个设备配置与 CE1 类似, 所以省略)

```
CE1#config
CE1(config)# interface vlan 1
CE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# router rip
CE1(config-router)#network 0.0.0.0/0
CE1(config-router)#redistribute connect
CE1(config-router)#exit
```

PE1 交换机的 MPLS BGP 配置如下: (PE2 配置类似, 所以省略)

```
(1) 配置 VPN 实例
PE1#config
PE1(config)#ip vrf vpna
PE1(config-vrf)#rd 100:1
PE1(config-vrf)#route-target both 100:1
PE1(config)#ip vrf vpnb
PE1(config-vrf)#rd 100:2
PE1(config-vrf)#route-target both 100:2

(2) 配置接口绑定 VPN 实例
PE1(config)# interface vlan 1
PE1(config-if-Vlan1)# ip vrf forwarding vpna
PE1(config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
PE1(config)# interface vlan 2
```

```
PE1(config-if-Vlan2)# ip vrf forwarding vpnb
PE1(config-if-Vlan1)#ip address 10.2.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
(3) 全局启动 MPLS 和 LDP
PE1(config)#mpls enable
PE1(config)#router ldp
PE1(config-router)#exit
(4) 配置接口以及使能 LDP
PE1(config)# interface loopback 1
PE1(config-if-Loopback1)# ip address 172.1.1.1 255.255.255.255
PE1(config-if-Loopback1)# exit
PE1(config)# interface vlan 100
PE1(config-if-Vlan100)#ip address 100.1.1.1 255.255.255.0
PE1(config-if-Vlan100)#label-switching
PE1(config-if-Vlan100) #ldp enable
PE1(config-if-Vlan100)#exit
(5) 启动 OSPF 协议传送内网路由
PE1(config)#router ospf
PE1(config-router)# ospf router-id 172.1.1.1
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)# redistribute connected
PE1(config-router)#exit
(6) 配置 RIP VRF 传送私网路由
PE1(config)#router rip
PE1(config-router)#address-family ipv4 vrf vpna
PE1(config-router-af)#network 0.0.0.0/0
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute bgp
PE1(config-router-af)#exit
PE1(config-router)#address-family ipv4 vrf vpnb
PE1(config-router-af)#network 0.0.0.0/0
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute bgp
PE1(config-router-af)#exit
PE1(config-router)#exit
(7) 配置 BGP
PE1(config)# router bgp 100
PE1(config-router)#neighbor 172.2.2.2 remote-as 100
PE1(config-router)#neighbor 172.2.2.2 update-source 172.1.1.1
PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 172.2.2.2 activate
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpna
```

```
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute ospf
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpnb
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute ospf
PE1(config-router-af)#exit
PE1(config-router)#exit
```

P 交换机的配置如下：

(1) 全局使能 MPLS 并且配置相关接口的 LDP

```
P#config
P(config)#mpls enable
P(config)#router ldp
P(config-router)#exit
P(config)# interface loopback 1
P(config-if-Loopback1)# ip address 172.3.3.3 255.255.255.255
P(config-if-Loopback1)# exit
P(config)#interface vlan 100
P(config-if-Vlan100)#ip address 100.1.1.2 255.255.255.0
P(config-if-Vlan100)#label-switching
P(config-if-Vlan100)#ldp enable
P(config-if-Vlan100)#exit
P(config)#interface vlan200
P(config-if-Vlan200)#ip address 200.1.1.2 255.255.255.0
P(config-if-Vlan200)#label-switching
P(config-if-Vlan200)#ldp enable
P(config-if-Vlan200)#exit
(2) 配置 OSPF 协议
P(config)#router ospf
P(config-router)# ospf router-id 172.3.3.3
P(config-router)# network 0.0.0.0/0 area 0
P(config-router)# redistribute connected
```

10.3.3.4 PE-CE 间使用静态路由建立 BGP MPLS VPN

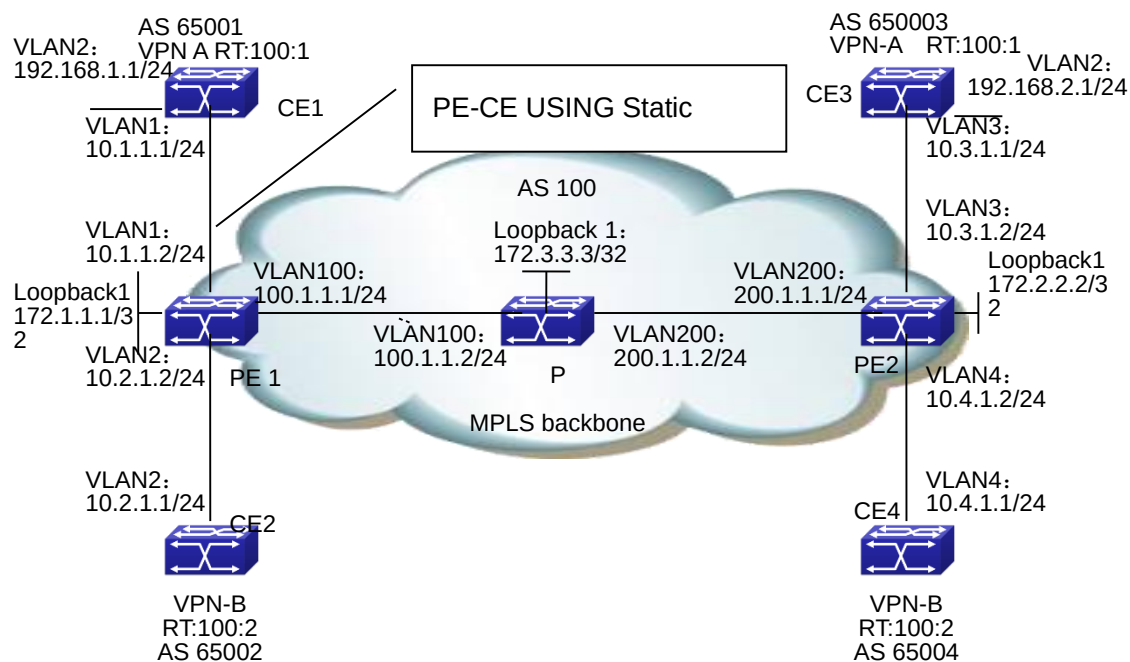


图 10-19 PE-CE 间使用静态路由建立 BGP MPLS VPN

CE1 的配置如下: (CE2~CE4 另外三个设备配置与 CE1 类似, 所以省略)

```
CE1#config
CE1(config)# interface vlan 1
CE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# interface loopback 1
CE1(config-if-Vlan1)#ip address 192.168.1.1 255.255.255.0
CE1(config-if-Vlan1)# exit
CE1(config)# ip route vrf vpna 192.168.2.1/24 10.1.1.2
```

PE1 交换机的 MPLS BGP 配置如下: (PE2 配置类似, 所以省略)

(1) 配置 VPN 实例

```
PE1#config
PE1(config)#ip vrf vpna
PE1(config-vrf)#rd 100:1
PE1(config-vrf)#route-target both 100:1
PE1(config)#ip vrf vpnb
PE1(config-vrf)#rd 100:2
PE1(config-vrf)#route-target both 100:2
```

(2) 配置接口绑定 VPN 实例

```
PE1(config)# interface vlan 1
PE1(config-if-Vlan1)# ip vrf forwarding vpna
PE1(config-if-Vlan1)#ip address 10.1.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
PE1(config)# interface vlan 2
```



```
PE1(config-if-Vlan2)# ip vrf forwarding vpnb
PE1(config-if-Vlan1)#ip address 10.2.1.2 255.255.255.0
PE1(config-if-Vlan1)#exit
(3) 全局启动 MPLS 和 LDP
PE1(config)#mpls enable
PE1(config)#router ldp
PE1(config-router)#exit
(4) 配置接口以及使能 LDP
PE1(config)# interface loopback 1
PE1(config-if-Loopback1)# ip address 172.1.1.1 255.255.255.255
PE1(config-if-Loopback1)# exit
PE1(config)# interface vlan 100
PE1(config-if-Vlan100)#ip address 100.1.1.1 255.255.255.0
PE1(config-if-Vlan100) #ldp enable
PE1(config-if-Vlan100)#exit
(5) 启动 OSPF 协议传送内网路由
PE1(config)#router ospf
PE1(config-router)# ospf router-id 172.1.1.1
PE1(config-router)# network 0.0.0.0/0 area 0
PE1(config-router)# redistribute connected
PE1(config-router)#exit
(6) 配置静态私网路由
PE1(config)# ip route vrf vpna 192.168.1.1/24 10.1.1.2
PE1(config)# ip route vrf vpnb 192.168.2.1/24 10.1.1.2
PE1(config-router)#address-family ipv4 vrf vpna
PE1(config-router-af)#network 0.0.0.0/0
PE1(config-router-af)#redistribute connected
PE1(config-router)#exit
(7) 配置 BGP
PE1(config)# router bgp 100
PE1(config-router)#neighbor 172.2.2.2 remote-as 100
PE1(config-router)#neighbor 172.2.2.2 update-source 172.1.1.1
PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 172.2.2.2 activate
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpna
PE1(config-router-af)#redistribute connected
PE1(config-router-af)#redistribute static
PE1(config-router-af)#exit
PE1(config-router)# address-family ipv4 vrf vpnb
PE1(config-router-af)#redistribute connected
PE1(config-router-af)# redistribute static
PE1(config-router-af)#exit
```

```
PE1(config-router)#exit
```

P 交换机的配置如下：

(1) 全局使能 MPLS 并且配置相关接口的 LDP

```
P#config
```

```
P(config)#mpls enable
```

```
P(config)#router ldp
```

```
P(config-router)#exit
```

```
P(config)# interface loopback 1
```

```
P(config-if-Loopback1)# ip address 172.3.3.3 255.255.255.255
```

```
P(config-if-Loopback1)# exit
```

```
P(config)#interface vlan 100
```

```
P(config-if-Vlan100)#ip address 100.1.1.2 255.255.255.0
```

```
P(config-if-Vlan100)#ldp enable
```

```
P(config-if-Vlan100)#exit
```

```
P(config)#interface vlan200
```

```
P(config-if-Vlan200)#ip address 200.1.1.2 255.255.255.0
```

```
P(config-if-Vlan100)#ldp enable
```

```
P(config-if-Vlan200)#exit
```

(2) 配置 OSPF 协议

```
P(config)#router ospf
```

```
P(config-router)# ospf router-id 172.3.3.3
```

```
P(config-router)# network 0.0.0.0/0 area 0
```

```
P(config-router)# redistribute connected
```

10.3.4 MPLS BGP VPN 排错帮助

在配置、使用 MPLS BGP VPN 协议时，可能会由于物理连接、配置错误等原因导致 VPN 未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证 PE1、P、PE2 之间的 OSPF 邻居建立正常，包括 loopback 口在内的路由发布正常，并且 PE 之间的 BGP 邻居成功建立。
- ☞ 其次应该保证 PE1、P、PE2 上全局启动了 LDP 并且在活动接口上正确启动了 LDP，检查下 PE1、P、PE2 上的 LDP session 建立是否正常；
- ☞ 然后，确认选取建立 VPN 所采用的 PE-CE 间的传送路由方式，以及对应的配置是否正确，是否在 CE 上发布了相关的私网路由到对端 PE，同时需要注意 CE 端是不需要配置 VRF 实例的。如果使用 EBGp 传送私网路由，注意 CE1 和 CE2 之间的 BGP 自治系统号不要相同，否则 BGP 的环路检测会过滤对应的私网路由；
- ☞ 接着，确认 PE 上 BGP VPN 实例的配置正确，如果采用 OSPF 或者 RIP 建立传送 PE-CE 间私网路由时，注意要引入 BGP 路由，并且在 BGP VPN 实例下引入对应的 OSPF 和 RIP 路由，正确配置后在 PE1 上执行 `show ip bgp vpnv4 all` 可以看到 CE1、CE2 的路由信息。之后在 PE 上执行 `show mpls vrf-table` 可以查看对应的私网路由已经分配了标签，并且其状态是 UP 的，如果对应的私网路由的 `vrf-table` 中 Oper Status 状态为 down，那么执行 `show mpls ftn-table` 查看对应的 FEC 是否生成了 ftn；
- ☞ 最后，在上述过程都正确的情况下，在 CE1 和 CE2 上使用 `show ip route` 命令可

以查看到自己 VPN 内的正确路由信息。除非已经对 BGP MPLS VPN 非常熟悉，我们不推荐用户采用静态路由的方式建立 VPN；

- ✎ 另外，如果正确的配置下保存重启设备后，CE 上没有立刻看到对端 CE 的路由，请耐心等待一段时间，OSPF、LDP、BGP 的连接建立以及发布路由都需要时间。

10.4 MPLS VPN 访问公网

10.4.1 访问公网简介

VPN 访问公网是指 VPN 站点能够访问公共 Internet，协议基本规范由 RFC4364 定义。RFC4364 定义了几种 VPN 访问 Internet 的方式：

- ✎ 非 VRF Internet 接入
- ✎ VRF Internet 接入方式一
- ✎ VRF Internet 接入方式三

10.4.1.1 非 VRF Internet 接入方式

非 VRF Internet 接入方式中，如图 4-1，PE 路由器与 Internet 网关通过非 VRF 接口通信，VPN 站点访问 Internet 的往返流量经过 PE 路由器的全局路由表转发。具有访问 Internet 功能的 CE 路由器与 PE 路由器存在两条连接，一条连接与 PE 的公网接口相连（公网连接），另一条连接与 PE 的私网接口相连（私网连接）。PE 路由器的全局路由表可以包含完整的 Internet 路由或者一部分，或者只有一条指向 Internet 网关的默认路由，CE 路由器通过公网连接学习 Internet 路由，并将 VPN 站点内全球注册的 IP 地址子网路由通过公网连接发布给 PE，PE 再将这些路由发布到 Internet 网关，最后发布到 Internet。VPN 站点访问 Internet 的往返流量也通过公网连接。CE 和 PE 之间的私网连接用于 CE 路由器学习和发布 VPN 内的私网路由，VPN 站点间也通过私网连接通信，经由 PE 路由器 VRF 路由表转发。在这种方式中，PE 路由器的全局路由表和 VRF 路由表之间完全隔离，VPN 路由分发和 Internet 路由分发过程完全独立。

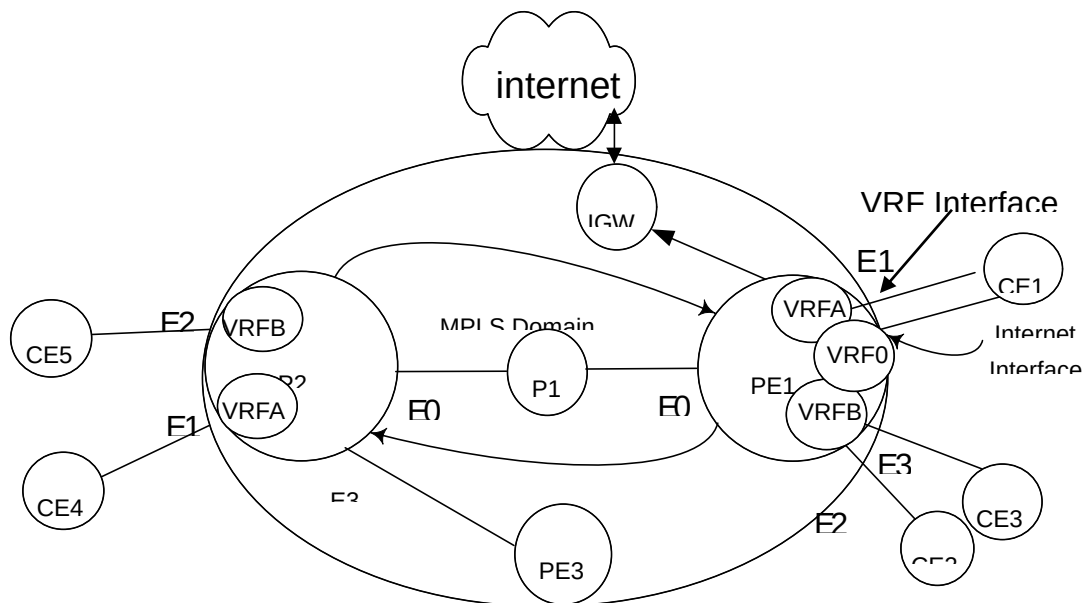


图 10-20 非 VRF Internet 接入

10.4.1.2 VRF Internet 接入方式一

VRF Internet 接入方式中,如图 4-2,PE 路由器与 Internet 网关通过非 VRF 接口通信,VPN 站点访问其他站点的 VPN 往返流量和访问 Internet 的往返流量均经过 CE 和 PE 的私网连接。PE 路由器包含完整的 Internet 路由或者只有一条到达 Internet 网关的缺省路由。VPN 访问 Internet 的 IP 报文到达 PE 的 VRF 接口时,如果在 VRF 路由表中查找失败,可以跳转到全局路由表中查找,如果命中,则报文可转发到 Internet 网关,再通过 Internet 网关转发到 Internet。为了实现 Internet 主机回访 VPN 站点,需要在 PE 全局路由表创建一条特殊的静态路由,该路由的目的网段为 VPN 站点内全球注册的 IP 地址子网,出接口为指向 VPN 站点的私网接口,下一跳为 CE 路由器,这条静态路由通过 PE 发布到 Internet 网关,经过 Internet 网关发布到 Internet。Internet 主机回访 VPN 站点的报文到达 PE 的公网接口,查找 PE 的全局路由表,如果匹配指向 VPN 站点的静态路由,则将报文通过私网接口转发到下一跳。在这种方式中,PE 路由器的全局路由表和 VRF 路由表并不是完全隔离的,PE 路由器的全局路由表包含了部分的 VPN 路由。

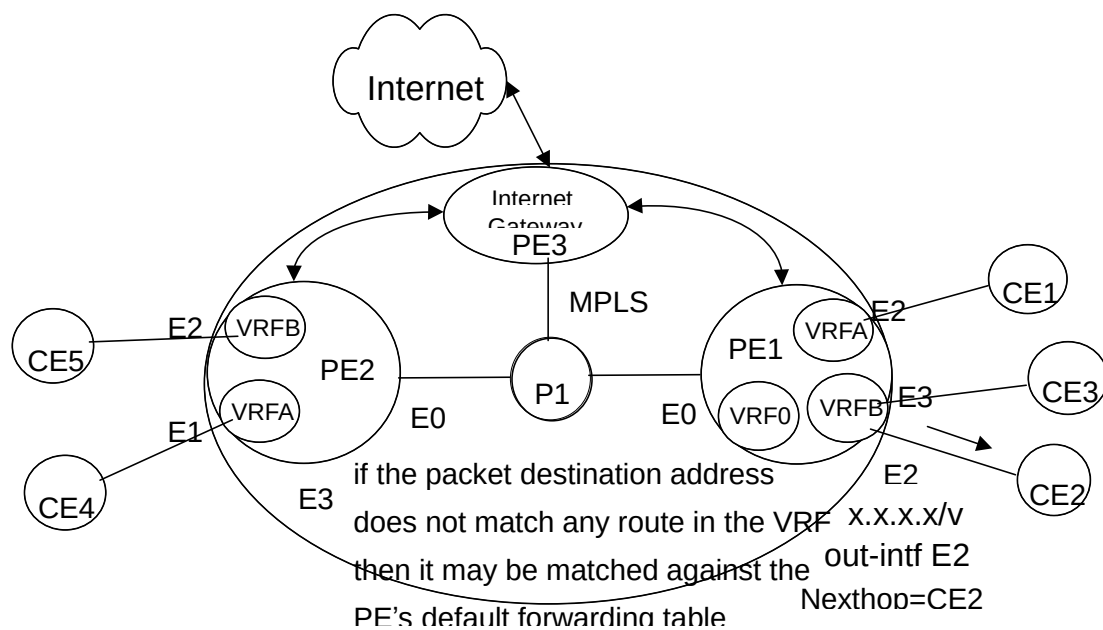


图 10-21 VRF Internet 接入方式一

10.4.1.3 VRF Internet 接入方式三

VRF Internet 接入 3 方式中，如图 4-3，VPN 站点通过 PE 和 CE 间的私网连接访问 Internet。PE 路由器的 VRF 路由表包含 Internet 路由。这些 Internet 路由是通过与 Internet 网关相连的 PE 路由器（称为 Internet PE）学到的，Internet PE 创建一个 Internet VRF，通过绑定 Internet VRF 的接口与 Internet 网关相连，这样 Internet 网关可将 Internet 路由发布给 Internet PE 的 VRF 路由表中，这些路由以 VPNv4 路由发布到其他 PE 路由器的 VRF 中。需访问 Internet 的 VPN 站点相连的 PE 路由器将对应的 VRF 路由（最好仅为目的网段为 VPN 站点内全球注册的 IP 地址子网的路由）通过 VPNv4 路由发布到 Internet PE，安装在 Internet VRF 中，Internet 网关再将这些路由发布到 Internet。这些路由的引入引出策略依赖于 MBGP 和 vrf 配置的 route-target。注意的是，在这种方式中，可以访问 Internet 的 VPN 站点是不允许出现地址或路由重叠的。

值得注意的是本种方式需要在 PE 上引入大量 Internet 路由，不建议用户使用此种方式实现访问公网。

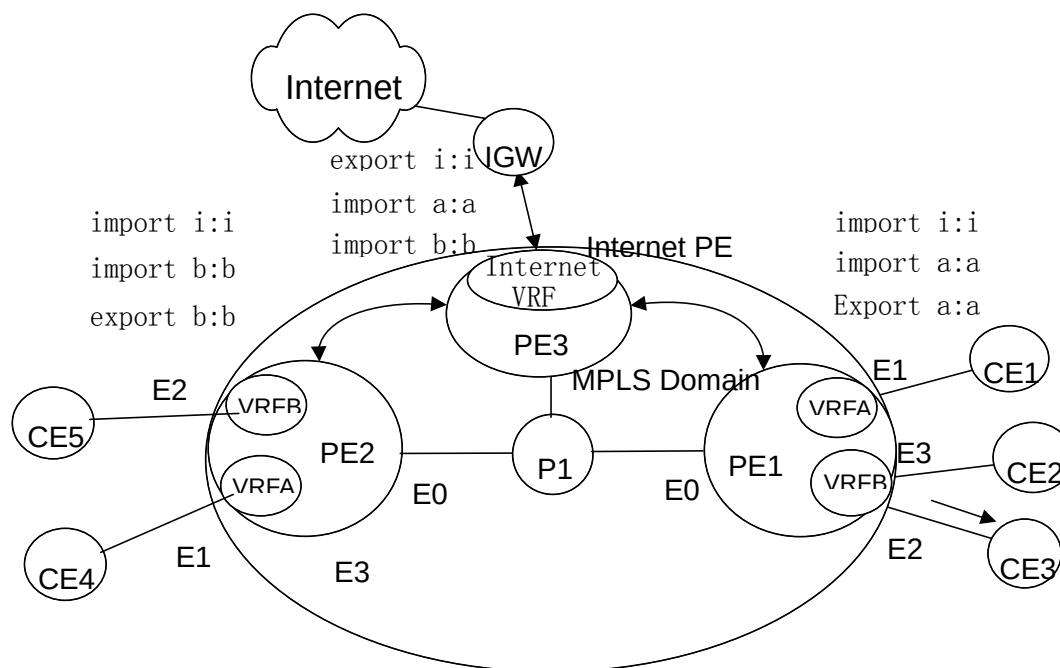


图 10-22 VRF Internet 接入 3

10.4.2 访问公网配置

访问公网配置任务序列：

1. 配置非 VRF Internet 接入
 - (1) 配置普通的 L3VPN
 - (2) 在 CE 和 PE 上增加一条公网连接，连接接口为 non-VRF 接口。
 - (3) 在 CE 上进行路由过滤，将公网路由通过公网连接发送到 PE，私网路由通过私网连接发送到 PE，分别在全局路由表和相应的 VRF 表中维护。
 - (4) 在 PE 和 CE 的公网接口上配置合适的过滤策略，将源地址和目的地址为私网地址的报文过滤掉。
 - (5) 配置缺省路由
 - 1) IGW 将缺省路由引入 BGP 协议
 - 2) PE 将缺省路由通过公网连接发布到 CE
 - 3) CE 通过私网连接将缺省路由发布到 PE，进而发布到其他 CE 上
 - (6) 配置静态路由
 - 1) 在 CE1 上配置静态路由，指向 Internet。
 - 2) 在 PE1 上配置静态路由，指向 CE 的公网接口。
2. 配置 VRF Internet 接入方式一
 - (1) 配置普通的 L3VPN
 - (2) PE 的私网接口配置 `ip vrf forwarding VPNA fallback global`。
 - (3) 配置 3 条静态路由，具体如下：
 - 1) 在 CE 上配置一条默认路由，下一跳为代理服务器。
 - 2) 在 PE 上增加一条到 Internet 的默认路由，下一跳为 IGW。PE 通过 OSPF 通告一条缺省路由，下一跳为自己。

- 3) 在 PE 全局路由表上增加一条从 Internet 回到代理服务器的静态路由, 目的为 VPN 公网地址, 下一跳为代理服务器, 并将该路由通过 OSPF 通告给其他 PE。

配置非 VRF Internet 接入

该配置不涉及单独的命令行, 按照配置序列完成配置即可。具体的命令参见相应功能的配置说明。

配置 VRF Internet 接入方式一

1. 配置 VRF Internet 接入方式一

- (1) 配置普通的 L3VPN
- (2) PE 的私网接口配置 `ip vrf forwarding VPNA fallback global`。
- (3) 配置 3 条静态路由, 具体如下:
 - 1) 在 CE 上配置一条默认路由, 下一跳为代理服务器。
 - 2) 在 PE 上增加一条到 Internet 的默认路由, 下一跳为 IGW。PE 通过 OSPF 通告一条缺省路由, 下一跳为自己。
 - 3) 在 PE 全局路由表上增加一条从 Internet 回到代理服务器的静态路由, 目的为 VPN 公网地址, 下一跳为代理服务器, 并将该路由通过 OSPF 通告给其他 PE。

命令	解释
配置普通的 L3VPN	参见 BGP MPLS VPN 配置部分
接口配置模式	
<code>ip vrf forwarding <vrf_name> fallback global no ip vrf forwarding <vrf_name> fallback global</code>	必选 配置 VRF 路由表的全局二次查找功能。缺省情况下未配置, 在配置二次查找以前需要先取消接口视图下 VRF 的配置。
全局配置模式	
<code>ip route vrf <vrf-name> {<ip-prefix> <mask> <ip-prefix/prefix-length>} {<gateway-address> null0} no ip route vrf <vrf-name> {<ip-prefix> <mask> <ip-prefix/<prefix-length>} {<gateway-address> null0}</code>	必选 配置静态路由。这里需要配置三条, 一条是 CE1 的默认路由, 一条是 PE3 上到 Internet 的默认路由, 最后一条是 PE1 上从 Internet 回到代理服务器的静态路由。

10.4.3 访问公网典型案例

10.4.3.1 非 VRF Internet 接入方式

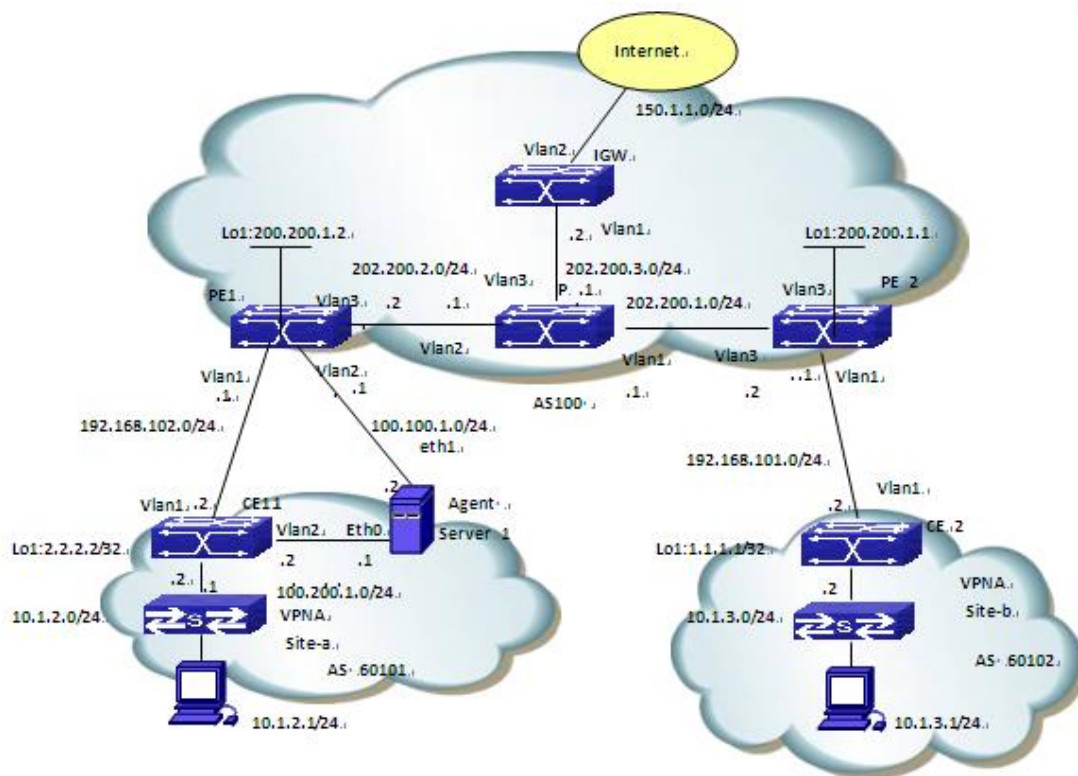


图 10-23 非 VRF Internet 接入方式

CE1 的配置如下:

```
CE1#config
CE1(config)#access-list 1 deny 100.100.1.0 0.0.0.255
CE1(config)#access-list 1 deny 100.200.1.0 0.0.0.255
CE1(config)#access-list 1 permit any-source
CE1(config)#access-list 2 permit 10.1.1.0 0.0.0.255
CE1(config)#access-list 2 permit 10.1.2.0 0.0.0.255
CE1(config)#access-list 2 deny any-source
CE1(config)# interface vlan 1
CE1(config-if-Vlan1)#ip address 192.168.102.2 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# interface vlan 2
CE1(config-if-Vlan1)#ip address 100.200.1.2 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# interface vlan 3
CE1(config-if-Vlan1)#ip address 10.1.2.2 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)# interface loopback 1
CE1(config-if-Vlan1)#ip address 2.2.2.2 255.255.255.255
CE1(config-if-Vlan1)# exit
CE1(config)#router bgp 60102
CE1(config-router)#network 120.1.1.0/24
CE1(config-router)#network 120.1.2.0/24
```



```
CE1(config-router)#network 10.1.2.0/24
CE1(config-router)#redistribute connected
CE1(config-router)#neighbor 100.100.1.1 remote-as 100
CE1(config-router)#neighbor 100.100.1.1 distribute-list 2 out
CE1(config-router)#neighbor 192.168.102.1 remote-as 100
CE1(config-router)#neighbor 192.168.102.1 default-originate
CE1(config-router)#neighbor 192.168.102.1 distribute-list 1 out
CE1(config-router)#exit
CE1(config)# ip route 100.100.1.1 255.255.255.0 100.200.1.1
CE1(config)# ip route 0.0.0.0/0 100.200.1.1
CE1(config)# exit
```

PE1 配置如下:

```
PE1#config
PE1(config)#access-list 100 deny ip 10.1.2.0 0.0.0.255 any-destination
PE1(config)#access-list 100 deny ip 10.1.2.0 0.0.0.255 any-destination
PE1(config)#access-list 100 deny ip 10.1.3.0 0.0.0.255 any-destination
PE1(config)#access-list 100 deny ip anysource 200.200.1.0 0.0.0.255
PE1(config)#access-list 100 deny ip anysource 202.200.0.0 0.0.255.255
PE1(config)#firewall enable
PE1(config-vrf)#ip vrf VRF-A
PE1(config-vrf)#rd 100:10
PE1(config-vrf)#route-target both 100:10
PE1(config-vrf)#exit
PE1(config)#interface vlan1
PE1(config-if-Vlan1)#ip vrf forwarding VRF-A
PE1(config-if-Vlan1)#ip address 192.168.102.1 255.255.255.0
PE1(config-if-Vlan1)#exit
PE1(config)#interface vlan2
PE1(config-if-Vlan2)#ip address 100.100.1.1 255.255.255.0
PE1(config-if-Vlan2)#ip access-group 1 in
PE1(config-if-Vlan2)#exit
PE1(config)# interface vlan3
PE1(config-if-Vlan3)#label-switching
PE1(config-if-Vlan3)#ldp enable
PE1(config-if-Vlan3)#ip address 202.200.2.2 255.255.255.0
PE1(config-if-Vlan3)#exit
PE1(config)#interface Loopback1
PE1(config)#ip address 200.200.1.2 255.255.255.255
PE1(config)#router ospf
PE1(config-router)#network 200.200.1.2/32 area 0
PE1(config-router)#network 202.200.2.0/24 area 0
PE1(config-router)#exit
PE1(config)#router bgp 100
```

```
PE1(config-router)#neighbor 100.200.1.2 remote-as 60102
PE1(config-router)#neighbor 200.200.1.1 remote-as 100
PE1(config-router)#neighbor 202.200.3.2 remote-as 100
PE1(config-router)#neighbor 202.200.3.2 next-hop-self
PE1(config-router)#address-family vpnv4 unicast
PE1(config-router-af)#neighbor 200.200.1.1 activate
PE1(config-router-af)#exit-address-family
PE1(config-router)#address-family ipv4 vrf VRF-A
PE1(config-router-af)#neighbor 192.168.102.2 remote-as 60102
PE1(config-router-af)#no neighbor 192.168.102.2 send-community extended
PE1(config-router-af)#exit-address-family
PE1(config-router)#exit
PE1(config)# router ldp
PE1(config-router)#ip route 100.200.1.2 255.255.255.0 100.100.1.2
```

P 配置如下

```
P#config
P(config)#interface Vlan1
P(config-if-Vlan1)#label-switching
P(config-if-Vlan1)#ldp enable
P(config-if-Vlan1)#ip address 202.200.1.1 255.255.255.0
P(config-if-Vlan1)#exit
P(config)#interface Vlan2
P(config-if-Vlan2)#label-switching
P(config-if-Vlan2)#ldp enable
P(config-if-Vlan2)#ip address 202.200.2.1 255.255.255.0
P(config-if-Vlan2)#exit
P(config)#interface Vlan3
P(config-if-Vlan3)#ip address 202.200.3.1 255.255.255.0
P(config-if-Vlan3)#exit
P(config)#router ospf
P(config-router)#network 202.200.1.0/24 area 0
P(config-router)#network 202.200.2.0/24 area 0
P(config-router)#network 202.200.3.0/24 area 0
P(config-router)#exit
P(config)#router ldp
```

PE2 如下配置:

```
PE2#config
PE2(config)#ip vrf VRF-A
PE2(config-vrf)#rd 100:10
PE2(config-vrf)#route-target both 100:10
PE2(config-vrf)#exit
PE2(config)#interface Vlan1
PE2(config-if-Vlan1)#ip vrf forwarding VRF-A
```

```
PE2(config-if-Vlan1)#ip address 192.168.101.1 255.255.255.0
PE2(config-if-Vlan1)#exit
PE2(config)#interface Vlan2
PE2(config-if-Vlan2)#label-switching
PE2(config-if-Vlan2)#ldp enable
PE2(config-if-Vlan2)#ip address 202.200.1.2 255.255.255.0
PE2(config-if-Vlan2)#exit
PE2(config)#interface Loopback1
PE2(config-if-loopback1)#ip address 200.200.1.1 255.255.255.255
PE2(config-if-loopback1)#exit
PE2(config)#router ospf
PE2(config-router)#network 200.200.1.1/32 area 0
PE2(config-router)#network 202.200.1.0/24 area 0
PE2(config-router)#exit
PE2(config)#router bgp 100
PE2(config-router)#address-family vpnv4 unicast
PE2(config-router-af)#neighbor 200.200.1.1 activate
PE2(config-router-af)#exit-address-family
PE2(config-router)#address-family ipv4 vrf VRF-A
PE2(config-router-af)#neighbor 192.168.101.2 remote-as 60101
PE2(config-router-af)#no neighbor 192.168.101.2 send-community extended
PE2(config-router-af)#exit-address-family
PE2(config-router)#exit
PE2(config)#router ldp
```

CE2 配置如下:

```
CE2#config
CE2(config)#interface vlan 1
CE2(config-if-Vlan1)#ip address 192.168.101.2 255.255.255.0
CE2(config-if-Vlan1)#exit
CE2(config)#interface Loopback1
CE2(config-if-Loopback1)#ip address 1.1.1.1 255.255.255.255
CE2(config-if-Loopback1)#exit
CE2(config)#router bgp 60101
CE2(config-router)#network 10.1.3.0/24
CE2(config-router)#neighbor 192.168.101.1 remote-as 100
```

IGW 如下配置:

```
IGW#config
IGW(config)#interface Vlan1
IGW(config-if-Vlan1)#ip address 202.200.3.2 255.255.255.0
IGW(config-if-Vlan1)#exit
IGW(config)#interface Vlan2
IGW(config-if-Vlan2)#ip address 150.1.1.1 255.255.255.0
IGW(config-if-Vlan2)#exit
```

```

IGW(config)#router ospf
IGW(config-router)#network 202.200.3.0 0.0.0.255 area 0
IGW(config-router)#exit
IGW(config)#router bgp 100
IGW(config-router)#neighbor 202.200.2.2 remote-as 100
IGW(config-router)#neighbor 202.200.2.2 default-originate

```

10.4.3.2 VRF Internet 接入方式一

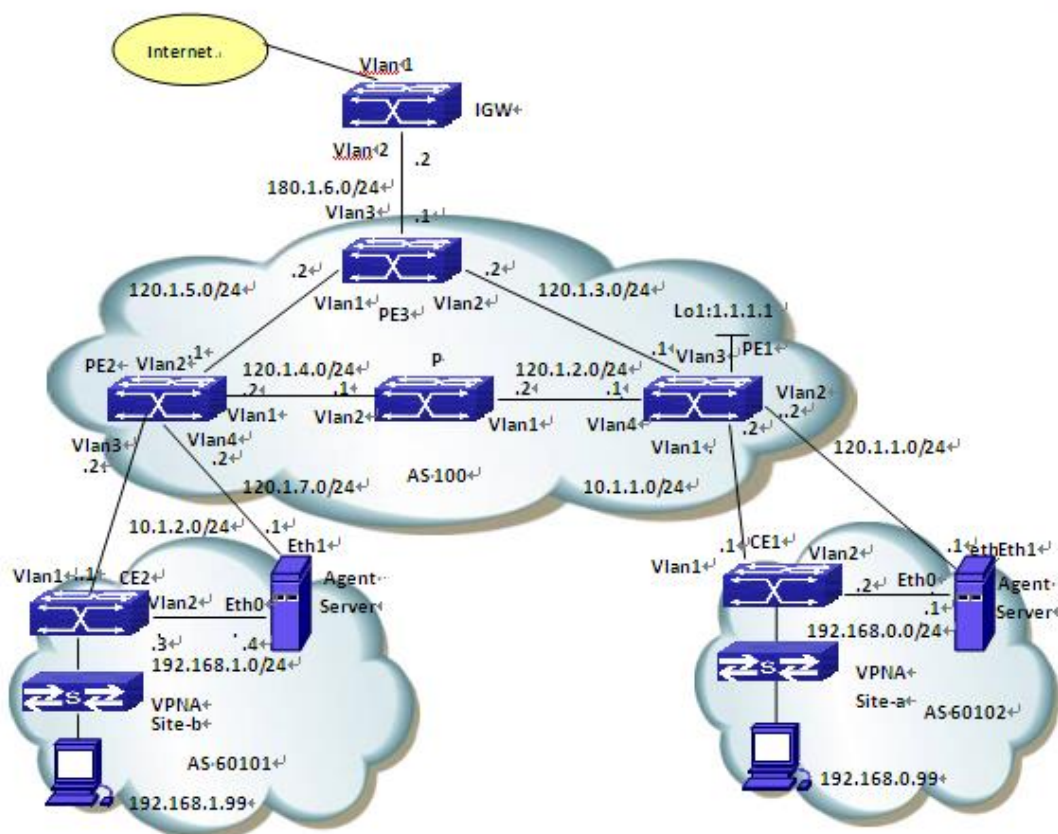


图 10-24 VRF Internet 接入方式一

site-a 和 site-b 属于 VPNA，site-a 和 site-b 之间的用户可以互访，site-a 和 site-b 的用户均有访问 Internet 的需求，在 site-a 和 site-b 内单独配置代理服务器实现 site-a 和 site-b 用户使用私网地址访问 Internet 进行 NAT 转换。

CE1 的配置如下

```

CE1#config
CE1(config)#interface Vlan1
CE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
CE1(config-if-Vlan1)#exit
CE1(config)#interface Vlan2
CE1(config-if-Vlan2)#ip address 192.168.0.2 255.255.255.0
CE1(config-if-Vlan2)#exit

```

```
CE1(config)#interface loopback1
CE1(config-if-Loopback1)#ip address 11.11.11.11 255.255.255.255
CE1(config-if-Loopback1)#exit
CE1(config)#ip route 0.0.0.0/0 192.168.0.1
CE1(config)#router bgp 60101
CE1(config-router)#neighbor 10.1.1.2 remote-as 100
CE1(config-router)#network 192.168.0.0/24
```

PE1 的配置文件

```
PE1#config
PE1(config)#ip vrf VPNA
PE1(config-vrf)#rd 100:10
PE1(config-vrf)#route-target both 100:10
PE1(config-vrf)#exit
PE1(config)#interface Vlan1
PE1(config-if-Vlan1)#ip vrf forwarding VPNA
PE1(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
PE1(config-if-Vlan1)#exit
PE1(config)#interface Vlan2
PE1(config-if-Vlan2)#ip vrf forwarding VPNA fallback global
PE1(config-if-Vlan2)#ip address 120.1.1.2 255.255.255.0
PE1(config-if-Vlan2)#exit
PE1(config)#interface Vlan3
PE1(config-if-Vlan3)#ip address 120.1.3.1 255.255.255.0
PE1(config-if-Vlan3)#exit
PE1(config)#interface Vlan4
PE1(config-if-Vlan4)#label-switching
PE1(config-if-Vlan4)#ldp enable
PE1(config-if-Vlan4)#ip address 202.200.1.2 255.255.255.0
PE1(config-if-Vlan4)#exit
PE1(config)#interface loopback1
PE1(config-if-Loopback1)#ip address 1.1.1.1 255.255.255.255
PE1(config-if-Loopback1)#exit
PE1(config)#router ospf
PE1(config-router)#redistribute static
PE1(config-router)#network 1.1.1.1/32 area 0
PE1(config-router)#network 120.1.2.0/24 area 0
PE1(config-router)#network 120.1.3.0/24 area 0
PE1(config-router)#exit
PE1(config)#router bgp 100
PE1(config-router)#neighbor 2.2.2.2 remote-as 100
PE1(config-router)#neighbor 2.2.2.2 update-source 1.1.1.1
PE1(config-router)#address-family vpnv4 unicast
PE1(config-router-af)#neighbor 2.2.2.2 activate
```

```
PE1(config-router-af)#exit-address-family
PE1(config-router)#address-family ipv4 vrf VPNA
PE1(config-router-af)#network 120.1.1.0/24
PE1(config-router-af)#neighbor 10.1.1.1 remote-as 60101
PE1(config-router-af)#no neighbor 10.1.1.1 send-community extended
PE1(config-router-af)#exit-address-family
PE1(config-router)#exit
PE1(config)#router ldp
PE1(config-router)#exit
PE1(config)#ip route 120.1.1.0/24 vlan 2 120.1.1.1
```

P 的配置文件

```
P#config
P(config)#interface Vlan1
P(config-if-Vlan1)#label-switching
P(config-if-Vlan1)#ldp enable
P(config-if-Vlan1)#ip address 120.1.2.2 255.255.255.0
P(config-if-Vlan1)#exit
P(config)#interface Vlan2
P(config-if-Vlan2)#label-switching
P(config-if-Vlan2)#ldp enable
P(config-if-Vlan2)#ip address 120.1.4.1 255.255.255.0
P(config-if-Vlan2)#exit
P(config)#router ospf
P(config-router)#network 0.0.0.0/0 area 0
P(config-router)#exit
P(config)#router ldp
```

PE2 的配置文件

```
PE2#config
PE2(config)#ip vrf VPNA
PE2(config-vrf)#rd 100:10
PE2(config-vrf)#route-target both 100:10
PE2(config-vrf)#exit
PE2(config)#interface Vlan1
PE2(config-if-Vlan1)#label-switching
PE2(config-if-Vlan1)#ldp enable
PE2(config-if-Vlan1)#ip address 120.1.4.2 255.255.255.0
PE2(config-if-Vlan1)#exit
PE2(config)#interface Vlan2
PE2(config-if-Vlan2)#ip address 120.1.5.1 255.255.255.0
PE2(config-if-Vlan2)#exit
PE2(config)#interface Vlan3
PE2(config-if-Vlan3)#ip vrf forwarding VPNA
PE2(config-if-Vlan1)#ip address 10.1.2.2 255.255.255.0
```

```
PE2(config-if-Vlan1)#exit
PE2(config)#interface Vlan4
PE2(config-if-Vlan4)#ip vrf forwarding VPNA fallback global
PE2(config-if-Vlan4)#ip address 120.1.7.2 255.255.255.0
PE2(config-if-Vlan4)#exit
PE2(config)#interface Loopback1
PE2(config-if-Loopback1)#ip address 2.2.2.2 255.255.255.255
PE2(config-if-Loopback)#exit
PE2(config)#router ospf
PE2(config-router)#redistribute static
PE2(config-router)#network 2.2.2.2/32 area 0
PE2(config-router)#network 120.1.4.0/24 area 0
PE2(config-router)#network 120.1.5.0/24 area 0
PE2(config-router)#exit
PE2(config)#router bgp 100
PE2(config-router)#neighbor 1.1.1.1 remote-as 100
PE2(config-router)#neighbor 1.1.1.1 update-source 2.2.2.2
PE2(config-router)#address-family vpnv4 unicast
PE2(config-router-af)#neighbor 1.1.1.1 activate
PE2(config-router-af)#exit-address-family
PE2(config-router)#address-family ipv4 vrf VPNA
PE2(config-router-af)#network 120.1.7.0/24
PE2(config-router-af)#neighbor 10.1.2.1 remote-as 60102
PE2(config-router-af)#no neighbor 10.2.1.1 send-community extended
PE2(config-router-af)#exit-address-family
PE2(config-router)#exit
PE2(config)#router ldp
PE2(config-router)#exit
PE2(config)#ip route 120.1.7.0/24 vlan 4 120.1.7.1
```

PE3 的配置文件

```
PE3#config
PE3(config)#interface Loopback1
PE3(config-if-Loopback1)#ip address 3.3.3.3 255.255.255.255
PE3(config-if-Loopback1)#exit
PE3(config-if-Vlan1)#interface Vlan1
PE3(config-if-Vlan1)#ip address 120.1.5.2 255.255.255.0
PE3(config-if-Vlan1)#exit
PE3(config)#interface Vlan2
PE3(config-if-Vlan2)#ip address 120.1.3.2 255.255.255.0
PE3(config-if-Vlan2)#exit
PE3(config)#interface Vlan3
PE3(config-if-Vlan3)#ip address 180.1.6.1 255.255.255.0
PE3(config-if-Vlan3)#exit
```

```
PE3(config)#router ospf 1
PE3(config-router)#default-information originate
PE3(config-router)# network 0.0.0.0/0 area 0
PE3(config-router)#exit
PE3(config)#router bgp 100
PE3(config-router)#network 120.1.1.0 mask 255.255.255.0
PE3(config-router)#network 120.1.7.0 mask 255.255.255.0
PE3(config-router)#neighbor 180.1.6.2 remote-as 200
PE3(config-router)#exit
PE3(config)#ip route 0.0.0.0/0 180.1.6.2
```

CE2 配置如下

```
CE2#config
CE2(config)#interface Vlan1
CE2(config-if-Vlan1)#ip address 10.1.2.1 255.255.255.0
CE2(config-if-Vlan1)#exit
CE2(config)#interface Vlan2
CE2(config-if-Vlan2)#ip address 192.168.1.3 255.255.255.0
CE2(config-if-Vlan2)#exit
CE2(config-if-Loopback1)#interface Loopback1
CE2(config-if- Loopback1)#ip address 22.22.22.22 255.255.255.255
CE2(config-if- Loopback1)#exit
CE2(config)#ip route 0.0.0.0/0 192.168.1.4
CE2(config)#router bgp 60101
CE2(config-router)#neighbor 10.1.2.2 remote-as 100
CE2(config-router)#network 192.168.1.0/24
CE2(config-router)#exit
```

IGW 的配置文件

```
IGW#config
IGW(config)#interface Vlan1
IGW(config-if-Vlan1)#ip address 180.1.5.2 255.255.255.0
IGW(config-if-Vlan1)#exit
IGW(config)#interface Vlan2
IGW(config-if-Vlan2)#ip address 180.1.6.2 255.255.255.0
IGW(config-if-Vlan2)#exit
IGW(config)#router bgp 200
IGW(config-router)#neighbor 180.1.6.1 remote-as 100
IGW(config-router)#exit
```

10.4.4 访问公网排错帮助

在配置、使用访问公网功能时，可能会由于物理连接、配置错误等原因导致未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证普通的 MPLS BGP VPN 功能正常运行，私网间可以互通，如果 VPN

不通请参考第三章 MPLS BGP VPN 的排错帮助。

- ✎ 其次应该确认所采用的访问公网是 VRF 接入方式还是非 VRF 接入方式，二者配置有较大的不同；
- ✎ 然后，如果是非 VRF 接入方式请注意一定要在 PE-CE 的非 VRF 接口上配置过滤规则，禁止私网路由和流量通过公网接口进入 PE 中，否则会有安全隐患；此外注意默认路由的发布以及 IGW 的 NAT 配置是否正确；
- ✎ 接着，如果是 VRF 接入方式一请注意配置私网接口时需要使用 `ip vrf forwarding vrf_name fallback global` 命令，确保在私网路由查找不到的时候才会二次查找全局路由表；此外注意公网回来的路由发布是否正确以及 IGW 的 NAT 配置是否正确；
- ✎ 最后，在上述过程都正确的情况下，在 CE 上都可以访问 Internet，如上所述的组网中，不管哪种方式其他 CE 都是通过 VPN 将流量发布到 PE 上以后访问 Internet 的，从 Internet 返回的流量也必须经过 PE 后转发。

10.5 VPLS

10.5.1 VPLS 简介

- ✎ 当今 IP 网络已经遍布全球，利用现有 IP 网络为企业提提供低成本专网逐渐成为各大运营商的关注点。因此，一种在 IP 网上提供 VPN 服务、可方便设定速率、配置简单的技术应运而生，这种技术即 MPLS VPN 技术。基于 MPLS 的 VPN 技术有两种，分别是 MPLS L3VPN 和 MPLS L2VPN。MPLS L3VPN 需要介入用户的内部路由管理，运营商的管理比较复杂。传统 VLL 方式的 MPLS L2VPN，在公网中提供一种点到点的 L2VPN 业务，可以让两个站点之间的连接效果像直接用链路连接一样，但它不能直接在服务提供者处进行多点间的交换。VPLS 在传统 MPLS L2VPN 方案的基础上发展而成，它可以实现多点到多点的 VPN 组网。VPLS 为运营商提供了一种更加完备的解决方案

VPLS，即 Virtual Private LAN Service（虚拟专用 LAN 业务），是一种在 MPLS 网络上提供类似 LAN 的一种业务，实际上是一种基于以太网的 L2VPN 技术。VPLS 集以太网和 MPLS 的优点于一体，他使分散在不同地理位置上的用户网络可以互相通信，就像它们直接相互连接在一起一样，VPLS 能使用户延伸他们的 LAN 到 MAN，甚至 WAN 上。

VPWS 是 VPLS 的一个特例，而 VPLS 是 VPWS 的扩展。VPLS 提供一个多点到多点的 VPN 组网，从这个角度来看，VPLS 网络群对用户 CE 而言类似一个交换机。而 VPWS 则相当于一根网线，它仅能提供基于点到点的

L2VPN 解决方案。

VPLS 可选择使用 LDP 信令和 BGP 信令来构建 PW，基于 LDP 协议的信令通过在每一对 PE 之间建立点到点的 LDP 会话来建立虚电路，由于 LDP 协议提出的较早，应用简单，目前有许多厂商的产品支持这一机制，所以从兼容性的角度考虑我们实现的是基于 LDP 的信令机制。

10.5.1.1 VPLS 基本概念

☞ VPLS (Virtual Private LAN Service)

虚拟专用 LAN 服务。是一种在 MPLS 网络上提供类似 LAN 的业务，它可以使用户从多个地理位置分散的点同时接入网络，相互访问，就像这些点直接接入到 LAN 上一样。VPLS 使用户延伸他们的 LAN 到 MAN，甚至 WAN 上。

☞ VC (Virtual Circuit)

虚电路。在两个节点之间的单向逻辑连接。一个 PW 有一对反向 VC 组成。

☞ S-TAG(Service Tag)、S-VID(Service VLAN ID)

服务提供商网络为了区分用户而压入的“服务界定符”。

☞ P-TAG(Provider Tag)、P-VID(Provider VLAN ID)

对端 PE 期望的 VLAN Tag。

☞ VFI (Virtual Forward Instance)

每个 VFI 提供单独的 VPLS 服务，VFI 实现以太网桥接功能，通过 VFI，可以将 VPLS 的实际接入链路映射到各条 PW 上，反过来并能够终结 PW，将 PW 映射到实际接入链路上。

☞ UPE (User facing-Provider Edge)

面向用户侧的 PE 设备，主要作为用户接入 VPN 的汇聚设备。

☞ NPE (Network Provider Edge)

网络核心 PE 设备，处于 VPLS 网络的核心域边缘，提供在核心网之间的 VPLS 透明传输服务。

☞ MTU (MaximumTransmission Unit)

最大传输单元。

☞ QinQ (802.1Q in 802.1Q)

一种基于 802.1Q 封装的隧道协议，能够提供点到多点的 L2VPN 服务。它将用户私网 VLAN Tag 封装到公网 VLAN Tag 中，最终让报文携带两层 Tag 穿越服务提供商的骨干网络，从而为用户提供一种较为简单的二层 VPN 隧道。

☞ PW Signaling

PW 信令，用于创建和维护 PW，是 VPLS 的实现基础。PW 信令协议还可以用于自动发现 VFI 的对端 PE 设备。目前 PW 信令协议主要有 LDP 和 BGP。本实现仅支持 LDP。

☞ VPWS (Virtual Private Wire Service)

虚拟专用线服务，是一种点对点的 L2VPN 服务，对于一端

(AC 或 PW)收到的报文直接转发到另一端(PW 或 AC)。VPWS 相对于 VPLS 来说不需要进行 MAC 地址的学习、查找,也不存在广播、多播等情况,所以效率更高。

10.5.1.2 VPLS 基本网络模型

VPLS 模型主要有五部分组成: CE、PE、P、AC 和 PW。其基本概念如下。

☞ CE (Custom Edge)

直接和服务提供商相连的用户边缘设备。CE 可以是路由器或交换机,也可以是一台主机。

☞ PE (Provider Edge)

服务提供商网络上的边缘设备,与 CE 相连,主要负责 VPN 业务的接入。它完成报文从私网到公网隧道,并从公网隧道到私网的映射和转发。

☞ P (Provider)

服务提供商网络中的骨干路由器,不与 CE 直接相连。P 设备只需要具备基本 MPLS 转发能力,不维护 VPN 信息。

☞ AC (Attachment Circuit)

接入电路。在 L2VPN 中,CE 通过 AC 接入到 PE。AC 可以是物理链路也可以是逻辑链路,AC 用于在 CE 和 PE 之间传输帧。

☞ PW (Pseudo Wire)

虚链路,简单的说就是 VC 加隧道,隧道可以是 LSP、GRE、或者是 CR-LSP。PW 对于 VPLS 来说,就像是一条本地 AC 到对端 AC 之间的一条直连通道,完成用户的二层数据透传。

VPLS 相关的 rfc4762 中提供了两种 VPLS 网络模型,分别是: PE 全连接的 VPLS 模型和分层 VPLS 模型(H-VPLS)。如图 5-1、5-2 所示。

10.5.1.3 VPLS 的优点

VPLS 具有以下优点:

- ☞ VPLS 在面向用户网一侧使用以太网接口,简化了 LAN/WAN 边界,可以支持快速和灵活的服务部署
- ☞ VPLS 将用户网络的路由策略控制和维护的权利交给了用户,简化了运营商网络的管理
- ☞ VPLS 服务内的所有用户 CE 是相同子网的一部分,简化了 IP 地址的规划
- ☞ VPLS 服务不需要参与 IP 寻址和路由

10.5.1.4 PE 全连接的基本 VPLS 网络模型

接入 VPLS 的各站点的 PE 设备之间逻辑全连接，PE 设备能在多点之间进行 MAC 地址学习以及数据包的转发。MPLS 网络提供隧道来透传 VPN 站点间的报文，网络中的 P 设备作用类似于 L3VPN 中的 P 设备，它不参与 MAC 地址的学习和交换，只对 MPLS 报文进行转发。PE 上各 VPN 之间的转发表相互独立，使得各 VPN 间的 MAC 地址可以重叠。

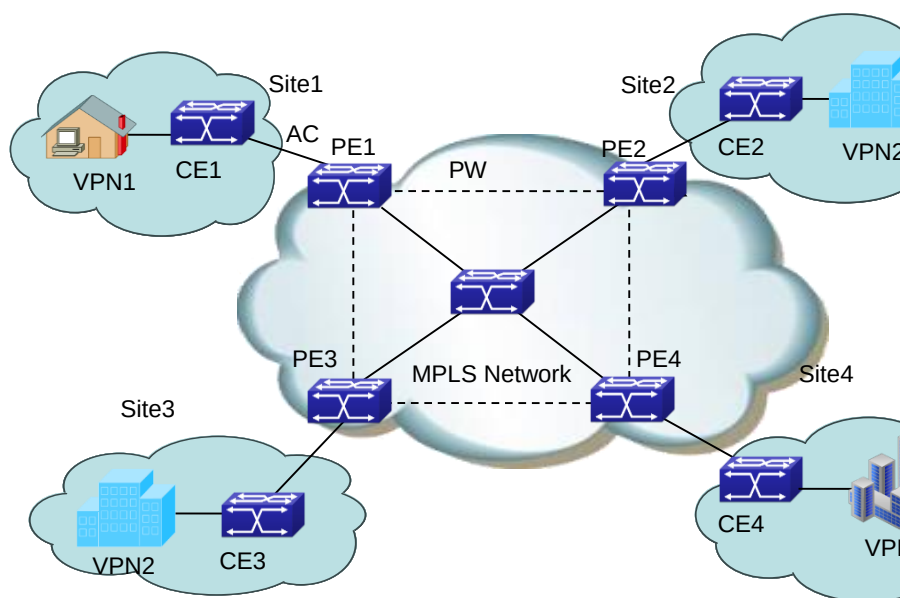


图 10-25 PE 全连接 VPLS 模型

PE 全连接基本 VPLS 模型环路避免方法

VPLS 全连接模型是通过水平分割转发来避免环路，每个 PE 设备如果从 PW 上收到报文，那么这个报文将不再向这个 VFI 关联的其它 PW 上转发，也就是说要求任意两个 PE 之间通过直接相连的 PW 通信，而不能通过第三个 PE 设备中转报文，这也就是 VPLS 的 VFI 之间要求建立全连接 PW 的原因。

10.5.1.5 分层 VPLS 模型

所有的 NPE 设备之间逻辑全连接，UPE 只与最近的 NPE 建立虚连接，通过 NPE 与对端 VPN 站点进行报文交换，这样层次化了网络拓扑。UPE 主要用于 VPN 业务的接入，对性能要求比较低，NPE 主要用于流量的汇聚，对性能要求比较高。另外，还可以在 UPE 和 NPE 之间增加链路备份，保证网络的健壮性。UPE 和 NPE 之间的虚连接可以使用 QinQ 或者 LDP 来建立。

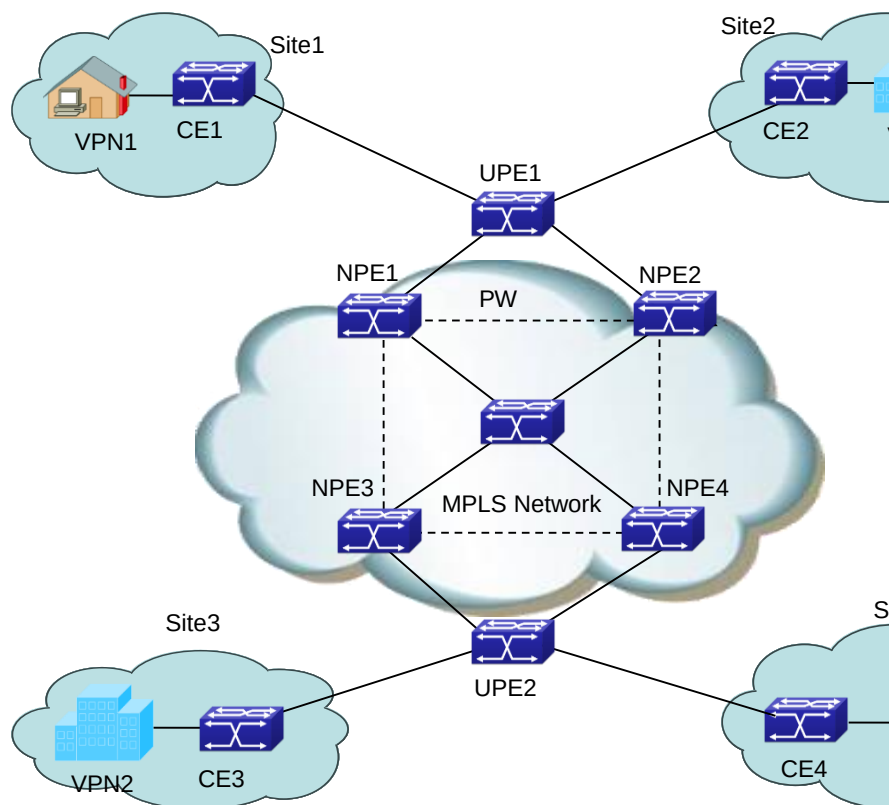


图 10-26 分层 VPLS 模型

根据 UPE 和 NPE 之间连接方式的不同, H-VPLS 分为:
LSP 接入方式和 QinQ 接入方式

☞ LSP 接入方式

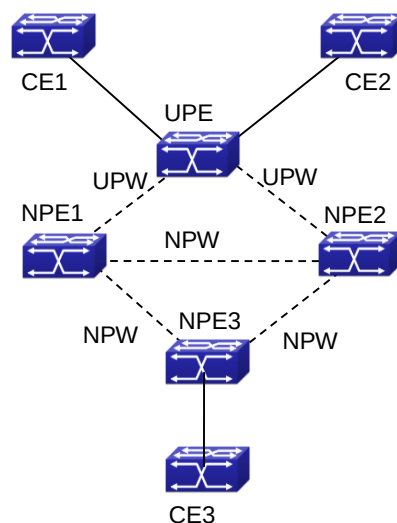


图 10-27 LSP 接入方式

如图 5-3 所示, UPE 作为汇聚设备, 它与 NPE1 和 NPE2 之间通过 LSP 接入, 需要在 UPE 和 NPE1、NPE2 之间分

别建立虚连接 U-PW（建立 U-PW 需要在 UPE 和 NPE 设备上分别创建 VFI，指定对等体，并要求两台设备的 PWID 必须相同），UPE 跟其他所有的对端都不建立虚连接。

LSP 接入方式的数据转发流程如下：

(1) UPE 负责将 CE 上送的报文发给 NPE1，同时打上 U-PW 对应的 VC 标签（NPE1 分配的 VC 标签，作为多路 PW 复用分离标记）；

(2) NPE1 收到报文后，先根据 VC 标签判断报文所属的 VFI，再根据该报文的目的 MAC 压入 N-PW 对应的 VC 标签，然后转发该报文；

(3) NPE1 从 N-PW 侧收到报文后，打上 U-PW 对应的 VC 标签，将报文发送给 UPE，UPE 再将报文转发给 CE。

CE1 与 CE2 之间的数据交换为本地 CE 之间交换时，如果 UPE 本身具有桥接功能，UPE 将直接完成两者间的报文转发，而无需将报文上送给 NPE1。不过对于目的 MAC 未知的第一个数据报文或广播报文，UPE 在将数据通过桥广播到 CE2 的同时，仍然会通过 U-PW 转发给 NPE1，由 NPE1 来完成报文的复制并转发到各个对端 CE。

QinQ 接入方式

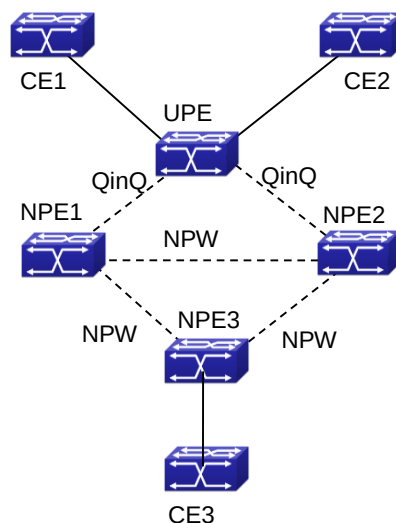


图 10-28 QinQ 接入方式

如图 5-4 所示，UPE 为标准的桥接设备，在 UPE 和 PE1 和 PE2 之间建立点到点的以太网 QinQ 连接（即在 UPE 面向 CE 的接口上使能 QinQ，在与 UPE 直连的 PE1 上使用 VLAN 接入方式）。UPE 从 CE 上收到的报文都将被打上一层外层 VLAN 标记，报文转发到 PE1 时，根据配置的 VLAN 接入方式，外层 VLAN 标记被解释为服务提供商 VLAN 标记，也就是服务提供商为用户分配的服务界定符，根据这个界定符，报文被映射到相应的 VFI，由这个 VFI 决策出报文如何转发（单播或广播）。

QinQ 接入方式的数据转发流程如下：

(1) 在 CE 接入端口使能 QinQ，为收到的报文添加压入 VLAN TAG 作为多路复用分离标记，在 UPE 与 PE1 之间通过 QinQ 隧道将报文透明传输到 PE1；

(2) PE1 先根据报文携带的 VLAN TAG 判断所属的 VFI，再根据该报文的目的 MAC 为其压入 PW 对应的 PW 标签，然后将其转发；

(3) PE 1 从 PW 侧收到报文后，根据 PW 标签，判断报文所属的 VFI，再根据用户报文的目的 MAC 打上 VLAN TAG 通过 QinQ 隧道将报文转发给 UPE，由 UPE 将报文转发给 CE。

如果 CE1 与 CE 2 之间的数据交换为本地 CE 之间交换，由于 UPE 本身具有桥接功能，UPE 将直接完成两者间的报文转发，而无需将报文上送给 PE1。不过对于目的 MAC 未知的第一个数据报文或广播报文，UPE 在通过桥广播到 CE2 的同时，仍然会通过 QinQ 隧道转发给 PE1，由 PE1 来完成报文的复制并转发到各个对端 CE。

🔗 H-VPLS 模型环路避免方法

和全连接相比，分层 VPLS 模型环路避免方法略有不同，由于 H-VPLS 只需在 NPE 之间建立全连接，UPE 和 NPE 之间不需要全连接，因此每个 NPE 设备上，从与 NPE 连接的 PW 上收到的报文，不再向这个 VFI 关联的，与其它 NPE 连接的 PW 转发，但可以向与 UPE 连接的 PW 转发。每个 NPE 设备上，从与 UPE 连接的 PW 上收到的报文，可以向这个 VFI 关联的所有与其它 NPE 连接的 PW 转发。

10.5.1.6 VPLS 的报文转发

在 VPLS 模型中的 AC 和 PW 上的报文封装方式分别有两种，其中 AC 上的是 VLAN 接入和 Ethernet 接入两种。含义分别如下：

🔗 VLAN 接入

CE 发送给 PE 或 PE 发送给 CE 的以太网帧头带有一个 VLAN TAG，该 TAG 是一个服务提供商网络为了区分用户而压入的“服务界定符”。服务界定符一般是服务提供商设备添加的，我们把这个作为服务界定符的 TAG 称为 S-TAG。

🔗 Ethernet 接入

CE 发送给 PE 或 PE 发送给 CE 的以太网帧头中没有服务界定符，如果此时帧头中有 VLAN TAG，则说明它只是用户报文的内部 VLAN TAG，对于 PE 设备没有意义。这种用户内部 VLAN 的 TAG 称为 U-TAG。

PW 上的报文封装方式分别是 Raw 模式和 Tagged 模式。含义分别如下：

🔗 Raw 模式

PW 上传输的帧不能带 S-TAG：对于 CE 侧的报文，如果收到带有服务界定符的报文，则将其去除后再压入 PW 标签

和隧道标签后转发；如果收到不带服务界定符的报文，则直接压入 PW 标签和隧道标签后转发。对于 PE 侧的下行报文，根据实际配置选择添加或不添加服务界定符后转发给 CE，但是它不允许重写或删除已经存在的任何 TAG。

🔗 Tagged 模式

PW 上传输的帧必须带 S-TAG：对于 CE 侧的报文，如果收到带有服务界定符的报文，则通过与对端 PE 发送过来的期望的 VLAN TAG 进行比较，如果相同，则保留 S-TAG，否则将 S-TAG 改写为对端 PE 期望的 VLAN TAG，如果对端 PE 没有传递期望的 VLAN TAG，则填写空 TAG（TAG 值为 0），然后再压入 PW 标签和隧道标签后转发；如果收到不带服务界定符的报文，则添加一个对端 PE 发送过来的期望的 VLAN TAG 或空 TAG 后，再压入 PW 标签和隧道标签后转发。对于 PE 侧的下行报文，根据实际配置选择重写、去除或保留服务界定符后转发给 CE。

VPLS 报文的转发方式根据 AC 接入方式和 PW 上报文封装方式的不同，可以分为以下四种：

🔗 Ethernet 接入 Raw 模式的报文转发

以图 5-1 的网络模型为例，AC 采用 Ethernet 接入方式，PW 上的报文封装模式为 Raw 模式，从 CE1 到 CE4 的报文转发过程如下：

1. CE1 将携带用户所属 VLAN 信息（U-Tag）的报文发送给 PE1；
2. PE1 根据用户目的 MAC 地址，选择合适的 PW，并在报文中添加 PW 对应的 VC 标签；
3. 为了在公网上利用 MPLS 隧道转发报文，PE1 在报文中添加公网隧道标签，通过公网隧道将报文传递给 PE4；
4. PE4 收到报文后，根据 VC 标签找到报文所属的 VFI，并将携带 U-TAG 的报文发送给 CE4。

🔗 Ethernet 接入 Tagged 模式的报文转发

以图 5-1 的网络模型为例，AC 采用 Ethernet 接入方式，PW 上的报文封装模式为 Tagged 模式时，报文的转发过程与 Ethernet 接入方式、Raw 模式的报文转发过程类似。所不同的是，PW 上传输的帧必须带 S-TAG。PE1 从 CE1 收到不带 S-TAG 的报文后，首先添加一个对端 PE 发送过来的期望的 VLAN TAG 或空 TAG，再压入两层 MPLS 标签转发。PE4 接收到报文后，去除 MPLS 两层标签和 S-TAG，再将报文转发给 CE4。

🔗 VLAN 接入 Raw 模式的报文转发

以图 5-1 的网络模型为例，AC 采用 VLAN 接入方式，PW 上的报文封装模式为 Raw 模式，从 CE1 到 CE4 的报文转发过程如下：

1. CE1 发送的报文应携带服务界定符 S-TAG，否则，

如果 CE1 发送的报文不带 VLAN TAG 或者所携带的 VLAN TAG 和 S-TAG 不匹配，则走普通的二层报文转发流程。

2. PE1 接收到报文后去除 S-TAG，并添加两层 MPLS 标签，通过公网 MPLS 隧道，将报文发送给 PE4。

3. PE4 去除接收到报文中的两层 MPLS 标签，增加 S-TAG 后，将报文转发给 CE4。

VLAN 接入 Tagged 模式的报文转发

以图 5-1 的网络模型为例，AC 采用 VLAN 接入方式，PW 上的报文封装模式为 Tagged 模式时，报文的转发过程与 VLAN 接入方式 Raw 模式的报文转发过程类似。二者的区别是，PW 上传输的帧携带 S-TAG，PE 接收到报文后，根据和对端 PE 发送过来的所期望的 VLAN TAG 相比较，如果相同，则保留 S-TAG，否则将 S-TAG 改写为对端 PE 期望的 VLAN TAG，如果对端 PE 没有传递期望的 VLAN TAG，则填写空 TAG（TAG 值为 0）。

10.5.2 VPLS 配置

VPLS 配置任务序列：

1. 全局启动路由协议（必选）
2. MPLS 基本配置（必选）
 - (1) 全局使能 MPLS
 - (2) 在接口使能标签交换能力
3. 配置 LDP 会话（必选）
 - (1) 配置模式下进入 ldp 视图，同时也代表开启了 LDP 功能
 - (2) 配置远端 LDP 邻居，使用 targeted-peer 命令
 - (3) 在相应的三层接口视图下使能 LDP 命令
4. 创建 PW 模板（可选）
 - (1) 创建 pw-class 并进入 pw 视图
 - (2) 设定 pw-class 的传输模式
5. 配置 VFI（必选）
 - (1) 创建 VFI，指定 VFI ID，并进入 VFI 视图
 - (2) 配置远端 PW 的终结点（一般情况与 LDP 的 targeted peer 相同）
 - (3) 可以为每个 peer 指定 pw-class
 - (4) 可以设置 VFI 的 PW 封装模式
6. 配置用户接入方式并绑定 VFI（VPLS 必选）
 - (1) 进入端口视图
 - (2) 配置端口绑定 VFI
7. 配置端口绑定 PW（VPWS 必选）
 - (1) 创建 L2VC，配置 VPWS 的对端 PE
 - (2) 进入端口视图
 - (3) 将端口与指定 PW 进行绑定，并指定接入模式

1. 全局启动路由协议（必选）

命令	解释
全局配置模式	
router ospf	进入 OSPF 路由配置模式。
OSPF 路由配置模式	
network 0.0.0.0/0 area 0	配置使能 OSPF 的接口网段地址，缺省情况下接口上 OSPF 未使能。

2. 全局 MPLS（必选）

命令	解释
全局配置模式	
mpls enable no mpls enable	启动 MPLS 协议；本命令的 no 操作关闭 MPLS 协议。
三层接口视图	
label-switching enable no label-switching	使能标签交换功能；缺省情况下未使能标签交换。

3. 配置 LDP 会话（必选）

- (1) 配置模式下进入 ldp 视图，同时也代表开启了 LDP 功能
- (2) 配置远端 LDP 邻居，使用 **targeted-peer** 命令
- (3) 在相应的三层接口视图下使能 LDP 命令

命令	解释
全局配置模式	
router ldp no router ldp	打开 LDP 协议；本命令的 no 操作为关闭 LDP 协议，缺省情况下未使能 LDP。
targeted-peer 1.1.1.1	配置远端 LDP 邻居；缺省没有远端邻居
接口配置模式	
ldp {enable disable}	接口上启动 LDP 协议；本命令的 no 操作为关闭接口上的 LDP 协议，接口缺省情况下未使能 LDP。

4. 创建 PW 模板（可选）

- (1) 创建 pw-class 并进入 pw 视图
- (2) 设定 pw-class 的传输模式

命令	解释
全局配置模式	
pw-class <pw-class-name> no pw-class <pw-class-name>	创建 pw-class；缺省情况下不存在任何 PW 模板。
pw 模板配置模式	

transport-mode {ethernet vlan}	配置 PW 模板的 PW 上的报文封装模式，ethernet 对应 raw 模式，vlan 对应 tagged 模式。

5. 配置 VFI（必选）

- (1) 创建 VFI，指定 VFI ID，并进入 VFI 视图
- (2) 配置远端 PW 的终结点（一般情况与 LDP 的 targeted peer 相同）
- (3) 可以为每个 peer 指定 pw-class
- (4) 可以设置 VFI 的 PW 封装模式

命令	解释
全局配置模式	
vfi <vfi-name> <vfi-id> no vfi <vfi-name>	创建 VFI 并指定 VFI ID；缺省情况下不存在任何 VFI。
VFI 配置模式	
peer ip-address [pw-id pw-id] [no-split-horizon] [pw-class class-name]	配置 VPLS 实例包含的对端 PE 可以配置是否使能水平分割(默认使能)和 pw 模板。
transport-mode {ethernet vlan}	配置 PW 模板的 PW 上的报文封装模式。

6. 配置用户接入方式并绑定 VFI（VPLS 必选）

- (1) 进入端口视图
- (2) 配置端口绑定 VFI

命令	解释
端口配置模式	
xconnect vfi vfi-id [mode {ethernet vlan [svid <svid>]}]	配置用来将一个端口绑定到一个 VFI，并配置 AC 的接入模式。缺省情况下端口不绑定任何 VFI。

7. 配置端口绑定 PW（VPWS 必选）

- (1) 创建 L2VC，配置 VPWS 的对端 PE
- (2) 进入端口视图
- (3) 将端口与指定 PW 进行绑定，并指定接入模式

命令	解释
全局配置模式	
l2-vc ip-address pw-id pw-id [group group-id] [pw-class class-name]	创建 L2VC，配置 VPWS 的对端 PE。缺省情况下没有配置对端 PE。
进入端口视图	
xconnect l2-vc pw-id <pw-id> [mode {ethernet vlan [svid <svid>]}]	将端口与指定 PW 进行绑定，并指定接入模式，缺省情况下端口不绑定到任何 PW。

10.5.3 VPLS 典型案例

10.5.3.1 全连接的 VPLS 配置

10.5.3.1.1 组网需求

- ✎ CE1 和 CE2 分属于两个站点，同属于一个 VPN1
- ✎ CE1 和 CE2 分别通过端口 Eth1/1 接入到 PE 设备
- ✎ CE1 到 PE1 的接入链路配置为 Ethernet 接入，CE2 和 PE2 之间的接入链路配置为 VLAN 接入 Svid 为 200
- ✎ PE1 和 PE2 之间 PW 上的报文封装模式配置为 RAW(Ethernet)模式，PE1 和 PE3，PE2 和 PE3 之间的报文封装模式配置为 Tagged(Vlan)模式
- ✎ 要求通过配置 VPLS 的方式建立一个二层 VPN1，使得 CE1 和 CE2 之间可以进行二层互访

10.5.3.1.2 组网图

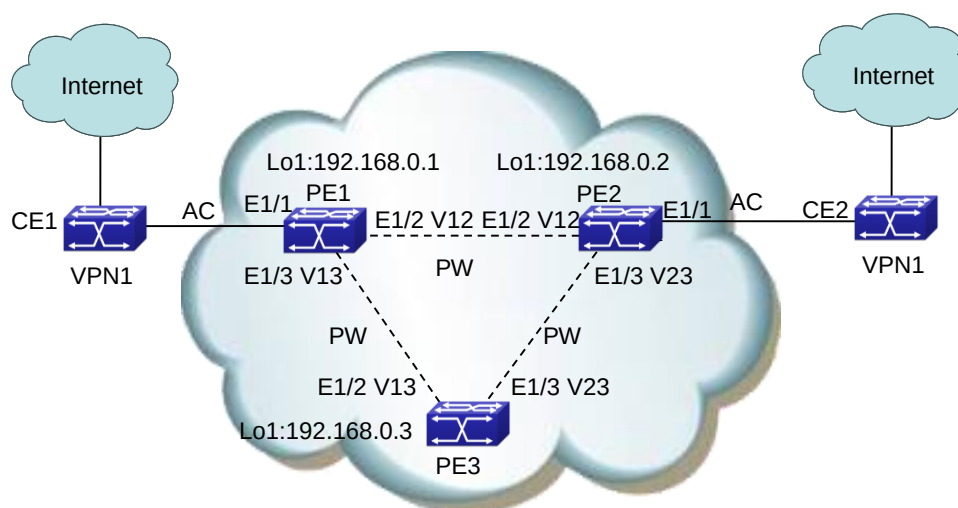


图 10-29 PE 全连接 VPLS 模型

10.5.3.1.3 配置步骤

请按照图 5-5 配置各接口 IP 地址和掩码，包括 VLAN 接口和 Loopback 接口，具体配置过程略。本配置只给出 LDP 方式下 VPLS 实例相关的配置。

(1) PE1 的配置

```
#创建 PW 模板
PE1(config)#pw-class c1
PE1(config-class) #transport-mode ethernet
PE1(config-class)#exit
PE1(config)#pw-class c2
```

```
PE1(config-class)#transport-mode vlan
```

```
PE1(config-class)#exit
```

#配置路由

```
PE1(config)#router ospf
```

```
PE1(config-router)#router-id 192.168.0.1
```

```
PE1(config-router)#network 0.0.0.0/0 area 0
```

```
PE1(config-router)#exit
```

#配置 MPLS 能力和标签交换能力(非 H-VPLS 情况下与 CE 相连的接口不用使能标签交换能力)

```
PE1(config)#mpls enable
```

```
PE1(config)#int vlan 12
```

```
PE1(config-if-vlan12)#label-switching
```

```
PE1(config-if-vlan12)#exit
```

```
PE1(config)#int vlan 13
```

```
PE1(config-if-vlan13)#label-switching
```

```
PE1(config-if-vlan13)#exit
```

#配置 LDP

```
PE1(config)#router ldp
```

```
PE1(config-router)#router-id 192.168.0.1
```

```
PE1(config-router)#targeted-peer 192.168.0.2
```

```
PE1(config-router)#targeted-peer 192.168.0.3
```

```
PE1(config-router)#exit
```

```
PE1(config)#int vlan 12
```

```
PE1(config-if-vlan12)#ldp enable
```

```
PE1(config-if-vlan12)#exit
```

```
PE1(config)#int vlan 13
```

```
PE1(config-if-vlan13)#ldp enable
```

```
PE1(config-if-vlan13)#exit
```

#配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```
PE1(config)#vfi v1 100
```

```
PE1(config-vfi)#peer 192.168.0.2 pw-class c1
```

```
PE1(config-vfi)#peer 192.168.0.3 pw-class c2
```

```
PE1(config-vfi)#exit
```

#在面向 CE1 的端口上绑定 vpls 实例模式为 ethernet

```
PE1(config-if-ethernet1/1)#xconnect vfi 100 mode ethernet
```

(2) PE2 的配置

#创建 PW 模板

```
PE2(config)#pw-class c1
```

```
PE2(config-class)#transport-mode ethernet
```

```
PE2(config-class)#exit
```

```
PE2(config)#pw-class c2
```

```
PE2(config-class)#transport-mode vlan
```

```
PE2(config-class)#exit
```

#配置路由

```
PE2(config)#router ospf
```

```
PE2(config-router)#router-id 192.168.0.2
```

```
PE2(config-router)#network 0.0.0.0/0 area 0
```

```
PE2(config-router)#exit
```

#配置 MPLS 能力和标签交换能力(非 H-VPLS 情况下与 CE 相连的接口不用使能标签交换能力)

```
PE2(config)#mpls enable
```

```
PE2(config)#int vlan 12
```

```
PE2(config-if-vlan12)#label-switching
```

```
PE2(config-if-vlan12)#exit
```

```
PE2(config)#int vlan 23
```

```
PE2(config-if-vlan23)#label-switching
```

```
PE2(config-if-vlan23)#exit
```

#配置 LDP

```
PE2(config)#router ldp
```

```
PE2(config-router)#router-id 192.168.0.2
```

```
PE2(config-router)#targeted-peer 192.168.0.1
```

```
PE2(config-router)#targeted-peer 192.168.0.3
```

```
PE2(config-router)#exit
```

```
PE2(config)#int vlan 12
```

```
PE2(config-if-vlan12)#ldp enable
```

```
PE2(config-if-vlan12)#exit
```

```
PE2(config)#int vlan 23
PE2(config-if-vlan23)#ldp enable
PE2(config-if-vlan23)#exit
```

#配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```
PE2(config)#vfi v1 100
PE2(config-vfi)#peer 192.168.0.1 pw-class c1
PE2(config-vfi)#peer 192.168.0.3 pw-class c2
PE2(config-vfi)#exit
```

#在面向 CE2 的端口上绑定 VPLS 实例模式为 VLAN，Svid 为 200

```
PE2(config-if-ethernet1/1)#xconnect vfi 100 mode vlan svid 200
```

(3) PE3 的配置

同 PE1 和 PE2 类似。

10.5.3.2 LSP 方式接入的 H-VPLS

10.5.3.2.1 组网需求

- ✎ CE1 和 CE2 分属于两个站点，同属于一个 VPN1
- ✎ CE1 和 CE2 分别通过端口 Eth1/1 接入到 PE 设备
- ✎ N-PE1、N-PE2、N-PE3 之间构建成网络侧全连接的 VPLS 网络
- ✎ U-PE 作为用户侧的接入设备，以分层的 VPLS PW 方式接入 N-PE1
- ✎ CE1 到 U-PE 的接入链路配置为 Ethernet 接入，CE2 和 N-PE3 之间的接入链路配置为 VLAN 接入 Svid 为 200
- ✎ N-PE 之间 PW 上的报文封装模式配置为 RAW(Ethernet)模式
- ✎ U-PE 和 N-PE1 之间为 PW 连接
- ✎ 要求通过配置 VPLS 的方式建立一个二层 VPN1，使得 CE1 和 CE2 之间可以进行二层互访

10.5.3.2.2 组网图

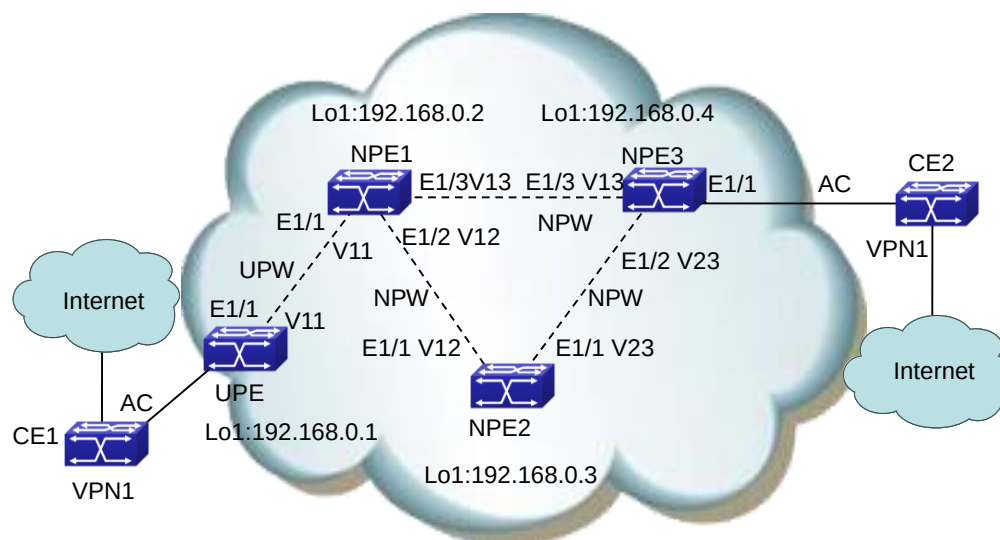


图 10-30 LSP 接入的 H-VPLS 应用组网

10.5.3.2.3 配置步骤

请按照图 5-6 配置各接口 IP 地址和掩码，包括 VLAN 接口和 Loopback 接口，具体配置过程略。本配置只给出 LDP 方式下实现 H-VPLS 相关的配置。

(1) UPE 的配置

#创建 PW 模板

```
UPE(config)#pw-class c1
UPE(config-class)#transport-mode ethernet
UPE(config-class)#exit
```

#配置路由

```
UPE(config)#router ospf
UPE(config-router)#router-id 192.168.0.1
UPE(config-router)#network 0.0.0.0/0 area 0
UPE(config-router)#exit
```

#配置 MPLS 能力和标签交换能力(与 CE 相连的接口不用使能标签交换能力)

```
UPE(config)#mpls enable
UPE(config)#int vlan 11
UPE(config-if-vlan11)#label-switching
UPE(config-if-vlan11)#exit
```

#配置 LDP

```
UPE(config)#router ldp
```



```
UPE(config-router)#router-id 192.168.0.1
UPE(config-router)#targeted-peer 192.168.0.2
UPE(config-router)#exit
UPE(config)#int vlan 11
UPE(config-if-vlan11)#ldp enable
UPE(config-if-vlan11)#exit
```

#配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```
UPE(config)#vfi v1 100
UPE(config-vfi)#peer 192.168.0.2 pw-class c1
UPE(config-vfi)#exit
```

#在面向 CE1 的端口上绑定 vpls 实例模式为 ethernet

```
UPE(config-if-ethernet1/1)#xconnect vfi 100 mode ethernet
```

(2) NPE1 的配置

#创建 PW 模板

```
NPE1(config)#pw-class c1
NPE1(config-class)#transport-mode ethernet
NPE1(config-class)#exit
```

#配置路由

```
NPE1(config)#router ospf
NPE1(config-router)#router-id 192.168.0.2
NPE1(config-router)#network 0.0.0.0/0 area 0
NPE1(config-router)#exit
```

#配置 MPLS 能力和标签交换能力

```
NPE1(config)#mpls enable
NPE1(config)#int vlan 11
NPE1(config-if-vlan11)#label-switching
NPE1(config-if-vlan11)#exit
NPE1(config)#int vlan 12
NPE1(config-if-vlan12)#label-switching
NPE1(config-if-vlan12)#exit
NPE1(config)#int vlan 13
NPE1(config-if-vlan13)#label-switching
NPE1(config-if-vlan13)#exit
```

#配置 LDP

```
NPE1(config)#router ldp
NPE1(config-router)#router-id 192.168.0.2
NPE1(config-router)#targeted-peer 192.168.0.1
NPE1(config-router)#targeted-peer 192.168.0.3
NPE1(config-router)#targeted-peer 192.168.0.4
NPE1(config-router)#exit
NPE1(config)#int vlan 11
NPE1(config-if-vlan11)#ldp enable
NPE1(config-if-vlan11)#exit
NPE1(config)#int vlan 12
NPE1(config-if-vlan12)#ldp enable
NPE1(config-if-vlan12)#exit
NPE1(config)#int vlan 13
NPE1(config-if-vlan13)#ldp enable
NPE1(config-if-vlan13)#exit
```

配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE(UPE 的 peer 需要关闭水平分割功能)

```
NPE1(config)#vfi v1 100
NPE1(config-vfi)#peer 192.168.0.3 pw-class c1
NPE1(config-vfi)#peer 192.168.0.4 pw-class c1
NPE1(config-vfi)#peer 192.168.0.1 no-split-horizon pw-class c1
NPE1(config-vfi)#exit
```

(3) NPE2 的配置

与 NPE1 类似，区别在于没有与 UPE 的 peer，不用关闭水平分割

(4) NPE3 的配置

#创建 PW 模板

```
NPE3(config)#pw-class c1
NPE3(config-class)#transport-mode ethernet
NPE3(config-class)#exit
```

#配置路由

```
NPE3(config)#router ospf
NPE3(config-router)#router-id 192.168.0.4
```

```
NPE3(config-router)#network 0.0.0.0/0 area 0
```

```
NPE3(config-router)#exit
```

#配置 MPLS 能力和标签交换能力

```
NPE3(config)#mpls enable
```

```
NPE3(config)#int vlan 13
```

```
NPE3(config-if-vlan11)#label-switching
```

```
NPE3(config-if-vlan11)#exit
```

```
NPE3(config)#int vlan 23
```

```
NPE3(config-if-vlan12)#label-switching
```

```
NPE3(config-if-vlan12)#exit
```

#配置 LDP

```
NPE3(config)#router ldp
```

```
NPE3(config-router)#router-id 192.168.0.4
```

```
NPE3(config-router)#targeted-peer 192.168.0.2
```

```
NPE3(config-router)#targeted-peer 192.168.0.3
```

```
NPE3(config-router)#exit
```

```
NPE3(config)#int vlan 13
```

```
NPE3(config-if-vlan11)#ldp enable
```

```
NPE3(config-if-vlan11)#exit
```

```
NPE3(config)#int vlan 23
```

```
NPE3(config-if-vlan12)#ldp enable
```

```
NPE3(config-if-vlan12)#exit
```

#配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```
NPE3(config)#vfi v1 100
```

```
NPE3(config-vfi)#peer 192.168.0.2 pw-class c1
```

```
NPE3(config-vfi)#peer 192.168.0.3 pw-class c1
```

```
NPE3(config-vfi)#exit
```

#在面向 CE2 的端口上绑定 vpls 实例

```
N-PE3(config-if-ethernet1/1)#xconnect vfi 100 mode vlan svid 200
```

10.5.3.3 QinQ 方式接入的 H-VPLS

10.5.3.3.1 组网需求

- ☞ CE1 和 CE2 分属于两个站点，同属于一个 VPN1
- ☞ CE1 和 CE2 分别通过端口 Eth1/2 和 Eth1/1 接入到 PE 设备
- ☞ N-PE1、N-PE2、N-PE3 之间构建网络侧全连接的 VPLS 网络
- ☞ U-PE 作为用户侧的接入设备，以分层的 QinQ 方式接入 N-PE1，Svid 为 100
- ☞ CE2 上采用正常的 VPLS 接入方式模式为 VLAN，Svid 为 200
- ☞ N-PE 之间 PW 上的报文封装模式配置为 Tagged 模式
- ☞ U-PE 和 N-PE1 之间为 QinQ 连接
- ☞ 要求通过配置 VPLS 的方式建立一个二层 VPN1，使得 CE1 和 CE2 之间可以进行二层互访

10.5.3.3.2 组网图

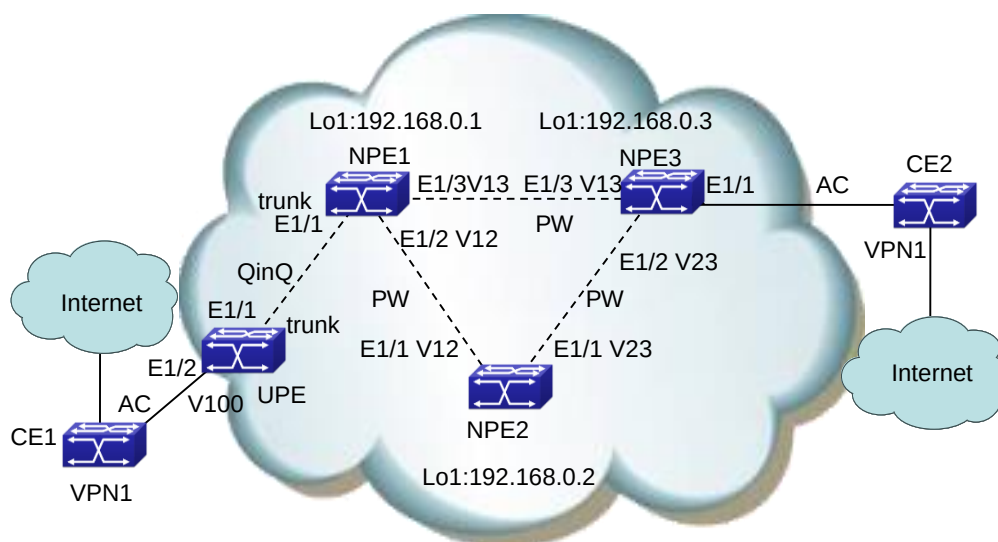


图 10-31 QinQ 接入的 H-VPLS 应用组网

10.5.3.3.3 配置步骤

请按照图 5-7 配置各接口 IP 地址和掩码，包括 VLAN 接口和 Loopback 接口，具体配置过程略。本配置只给出 QinQ 方式下实现 H-VPLS 相关的配置。

(1) UPE 的配置

#在 CE1 的接入端口上使能 QinQ，该端口所属的 VLAN 为 100

```
UPE(config-if-ethernet1/2)#switchport access vlan 100
```

```
UPE(config-if-ethernet1/2)#dot1q-tunnel enable
```

#把与 NPE1 连接的端口配置为 Trunk，默认情况下 trunk 端口允许所有 VLAN 通过

```
UPE(config-if-ethernet1/1)#switchport mode trunk
```

(2) NPE1 的配置

#创建 PW 模板

```
NPE1(config)#pw-class c1
NPE1(config-class)#transport-mode Vlan
NPE1(config-class)#exit
```

#配置路由

```
NPE1(config)#router ospf
NPE1(config-router)#router-id 192.168.0.1
NPE1(config-router)#network 0.0.0.0/0 area 0
NPE1(config-router)#exit
```

#配置 MPLS 能力和标签交换能力

```
NPE1(config)#mpls enable
NPE1(config)#int vlan 12
NPE1(config-if-vlan12)#label-switching
NPE1(config-if-vlan12)#exit
NPE1(config)#int vlan 13
NPE1(config-if-vlan13)#label-switching
NPE1(config-if-vlan13)#exit
```

#配置 LDP

```
NPE1(config)#router ldp
NPE1(config-router)#router-id 192.168.0.1
NPE1(config-router)#targeted-peer 192.168.0.2
NPE1(config-router)#targeted-peer 192.168.0.3
NPE1(config-router)#exit
NPE1(config)#int vlan 12
NPE1(config-if-vlan12)#ldp enable
NPE1(config-if-vlan12)#exit
NPE1(config)#int vlan 13
NPE1(config-if-vlan13)#ldp enable
NPE1(config-if-vlan13)#exit
```

配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```
NPE1(config)#vfi v1 100
NPE1(config-vfi)#peer 192.168.0.2 pw-class c1
NPE1(config-vfi)#peer 192.168.0.3 pw-class c1
NPE1(config-vfi)#exit
```

配置与 UPE QinQ 方式接入的端口上绑定 vpls 实例模式为 VLAN，Svid 为 100

```
NPE1(config-if-ethernet1/2)#switchport mode trunk
```

```
NPE1 (config-if-ethernet1/2)#xconnect vfi 100 mode vlan svid 100
```

(3) NPE2 的配置

与 NPE1 类似

(4) NPE3 的配置

#创建 PW 模板

```
NPE3(config)#pw-class c1
```

```
NPE3(config-class)#transport-mode Vlan
```

```
NPE3(config-class)#exit
```

#配置路由

```
NPE3(config)#router ospf
```

```
NPE3(config-router)#router-id 192.168.0.3
```

```
NPE3(config-router)#network 0.0.0.0/0 area 0
```

```
NPE3(config-router)#exit
```

#配置 MPLS 能力和标签交换能力

```
NPE3(config)#mpls enable
```

```
NPE3(config)#int vlan 13
```

```
NPE3(config-if-vlan11)#label-switching
```

```
NPE3(config-if-vlan11)#exit
```

```
NPE3(config)#int vlan 23
```

```
NPE3(config-if-vlan12)#label-switching
```

```
NPE3(config-if-vlan12)#exit
```

#配置 LDP

```
NPE3(config)#router ldp
```

```
NPE3(config-router)#router-id 192.168.0.3
```

```
NPE3(config-router)#targeted-peer 192.168.0.1
```

```
NPE3(config-router)#targeted-peer 192.168.0.2
```

```
NPE3(config-router)#exit
```

```
NPE3(config)#int vlan 13
```

```
NPE3(config-if-vlan11)#ldp enable
```

```
NPE3(config-if-vlan11)#exit
```

```

NPE3(config)#int vlan 23
NPE3(config-if-vlan12)#ldp enable
NPE3(config-if-vlan12)#exit

```

#配置 LDP 方式下的 VPLS 实例 V1 以及实例中的远端 PE

```

NPE3(config)#vfi v1 100
NPE3(config-vfi)#peer 192.168.0.1 pw-class c1
NPE3(config-vfi)#peer 192.168.0.2 pw-class c1
NPE3(config-vfi)#exit

```

#在面向 CE2 的端口上绑定 VPLS 实例

```

N-PE3(config-if-ethernet1/1)#xconnect vfi 100 mode vlan svid 200

```

10.5.3.4 VPWS 配置

10.5.3.4.1 组网需求

- ✎ CE1 和 CE2 分属于两个站点，同属于一个 VPN1
- ✎ CE1 和 CE2 分别通过端口 Eth1/1 接入到 PE 设备
- ✎ CE1 到 PE1 的接入链路配置为 Ethernet 接入，CE2 和 PE2 之间的接入链路配置为 VLAN 接入 Svid 为 200
- ✎ PE1 和 PE2 之间 PW 上的报文封装模式配置为 RAW(Ethernet)模式
- ✎ 要求通过配置 VPLS 的方式建立一个 VPWS 方式的二层 VPN1，使得 CE1 和 CE2 之间可以进行二层互访

10.5.3.4.2 组网图

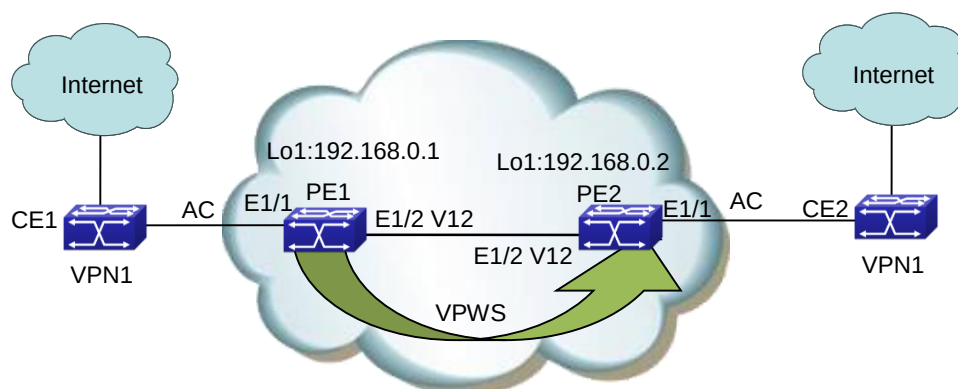


图 10-32 VPWS 组网图

10.5.3.4.3 配置步骤

请按照图 5-8 配置各接口 IP 地址和掩码，包括 VLAN 接口和 Loopback 接口，具体配置过程略。本配置只给出 VPWS 相关的配置。

(1) PE1 的配置

#创建 PW 模板

```
PE1(config)#pw-class c1
```

```
PE1(config-class)#transport-mode ethernet
```

```
PE1(config-class)#exit
```

#配置路由

```
PE1(config)#router ospf
```

```
PE1(config-router)#router-id 192.168.0.1
```

```
PE1(config-router)#network 0.0.0.0/0 area 0
```

```
PE1(config-router)#exit
```

#配置 MPLS 能力和标签交换能力(非 H-VPLS 情况下与 CE 相连的接口不用使能标签交换能力)

```
PE1(config)#mpls enable
```

```
PE1(config)#int vlan 12
```

```
PE1(config-if-vlan12)#label-switching
```

```
PE1(config-if-vlan12)#exit
```

#配置 LDP

```
PE1(config)#router ldp
```

```
PE1(config-router)#router-id 192.168.0.1
```

```
PE1(config-router)#targeted-peer 192.168.0.2
```

```
PE1(config-router)#exit
```

#配置 VPWS 的对端 PE

```
PE1(config)#l2-vc 192.168.0.2 pw-id 1 pw-class c1
```

#在面向 CE1 的端口上绑定 VPWS 模式为 ethernet

```
PE1(config-if-ethernet1/1)#xconnect l2-vc pw-id 1 mode Ethernet
```

(2) PE2 的配置

#创建 PW 模板


```
PE2(config)#pw-class c1
PE2(config-class)#transport-mode vlan
PE2(config-class)#exit
```

#配置路由

```
PE2(config)#router ospf
PE2(config-router)#router-id 192.168.0.2
PE2(config-router)#network 0.0.0.0/0 area 0
PE2(config-router)#exit
```

#配置 MPLS 能力和标签交换能力(非 H-VPLS 情况下与 CE 相连的接口不用使能标签交换能力)

```
PE2(config)#mpls enable
PE2(config)#int vlan 12
PE2(config-if-vlan12)#label-switching
PE2(config-if-vlan12)#exit
```

#配置 LDP

```
PE2(config)#router ldp
PE2(config-router)#router-id 192.168.0.2
PE2(config-router)#targeted-peer 192.168.0.1
PE2(config-router)#exit
PE2(config)#int vlan 12
PE2(config-if-vlan12)#ldp enable
PE2(config-if-vlan12)#exit
```

#配置 VPWS 的对端 PE

```
PE2(config)#l2-vc 192.168.0.1 pw-id 1 pw-class c1
```

#在面向 CE2 的端口上绑定 VPWS 模式为 VLAN，Svid 为 200

```
PE2(config-if-ethernet1/1)#xconnect l2-vc pw-id 1 mode vlan svid 200
```

10.5.4 VPLS 排错帮助

- ☞ 在配置、使用 VPLS 时，可能会由于物理连接、配置错误等原因导致 L2 VPN 未能正常运行。因此，用户应注意以下要点：
- ☞ 首先应该保证 PE1、P、PE2 之间的 OSPF 邻居建立正常，包括 loopback 口在内的路由发布正常，即检查所有 PE 设备上是否都存在对方 PE、P 的路由；
- ☞ 其次应该保证 PE1、P、PE2 上全局启动了 MPLS 和 LDP 并且在活动接口上正确启动了

Label-switching 和 LDP，检查下 PE1、P、PE2 上的 LDP session 建立是否正常，确认 LDP 的远端邻居即 `targeted peer` 地址是否配置正确，是否正确建立了邻居，同时检查 LDP 远端 `peer` 是否和配置的 VFI 上或者 L2VC 上的 `peer` 地址是否一致；

- ☞ 然后，通过 `show vpls peer xxx` 命令查看 PE 上 PW 是否建立成功，对应的 Peer 其 `pw` 状态应该为 UP，如果没有 UP 请检查两端配置的 PW 封装方式(ethernet 或者 vlan)是否一致，VFI ID 是否一致，MTU、传输模式等参数协商是否一致；
- ☞ 接着，判断绑定到 VFI 的端口是否绑定到正确的 VFI 上，其接入模式是否正确，Svid 是否存在错误，如果为 VPWS 方式，检查端口上绑定的 `pw-id` 是否正确，两端 PE 上绑定的 `pw-id` 是否相同；
- ☞ 最后，在上述过程都正确的情况下，在 CE1 和 CE2 上使用 `show vfi mac-addresses-table` 可以查看到自己 VPN 内的正确 MAC 地址信息，且两端都可以显示对方 CE 上的 MAC 地址信息；
- ☞ 另外，如果正确的配置下保存重启设备后，CE 上没有立刻看到对端 CE 的 MAC 地址流量也没有立即恢复，请耐心等待一段时间，OSPF、LDP、PW 的连接建立都需要一定时间；
- ☞ 注意： PE 端口在绑定到 VFI 以后建议该端口不要打开 802.1X、STP、QinQ 等可能会影响用户侧的网络功能，另外该端口所属的三层 VLAN 端口如果有路由(例如 OSPF、RIP)、组播(IGMP、MLD)等协议可能会造成运营商的部分 IP 等信息泄漏到 CE 一侧，所以建议该端口不要加入公网上正在使用的三层接口中。

第11章 虚拟化操作

11.1 VSF

11.1.1 概述

11.1.1.1 VSF 简介

VSF 就是将多台设备通过 VSF 口连接起来形成一台虚拟的逻辑设备。用户对这台虚拟设备进行管理，来实现对虚拟设备中所有物理设备的管理。

传统的园区和数据中心网络是使用多层网络拓扑结构设计的，如图 1-1 所示。这些网络类型有以下缺点：

- (1) 网络和服务复杂，从而导致运营效率低、运营开支高。
- (2) 无状态的网络级故障切换会延长应用恢复时间和业务中断时间。
- (3) 使用率低下资源降低了投资回报（ROI），提高了资本开支。

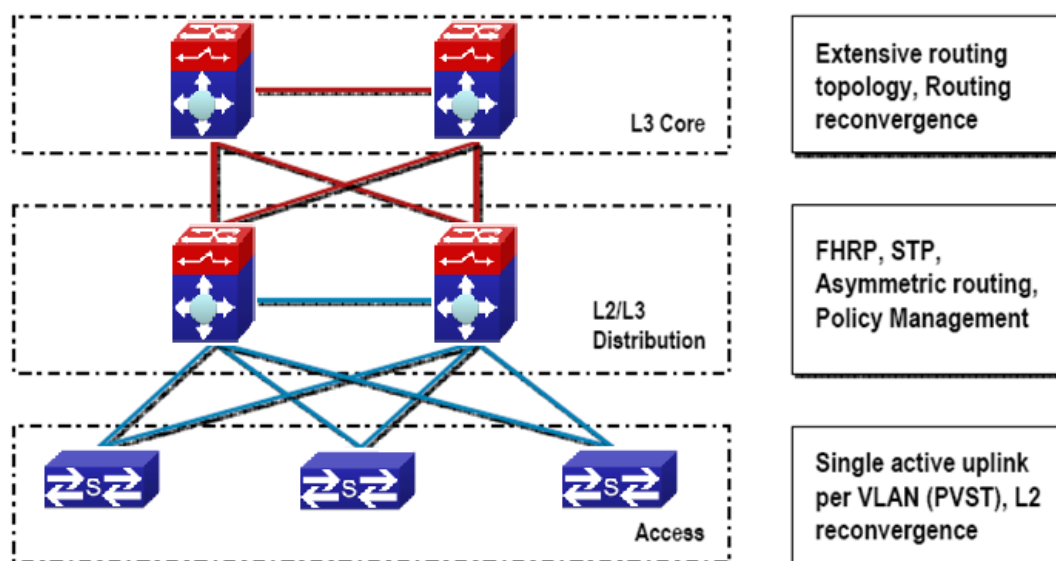


图 11-1 传统的企业网

为了解决这些问题，出现了 VSF 技术，将多台支持 VSF 的设备组合为单一虚拟交换机。在 VSF 中，这两个交换机中的管理引擎的数据面板和交换阵列能同时激活。VSF 成员通过 VSF 链路（VSL）连接。VSL 在虚拟交换机成员之间使用标准万兆以太网连接（多达 4 条，以提供冗余性），如图 1-2。

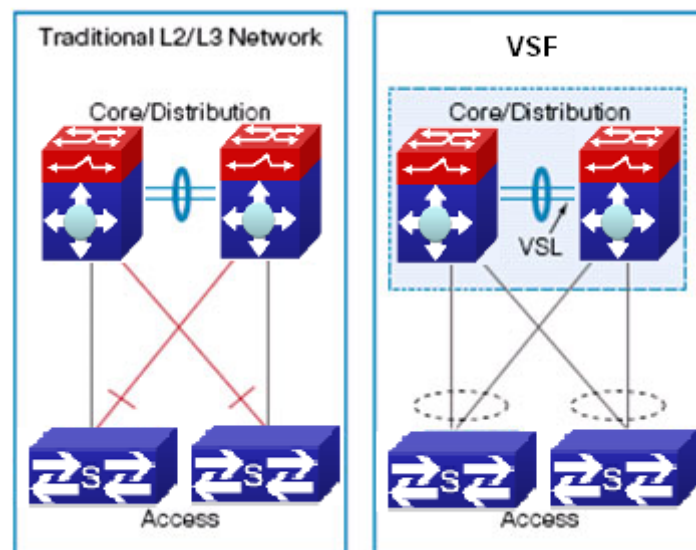


图 11-2 VSF 示意图

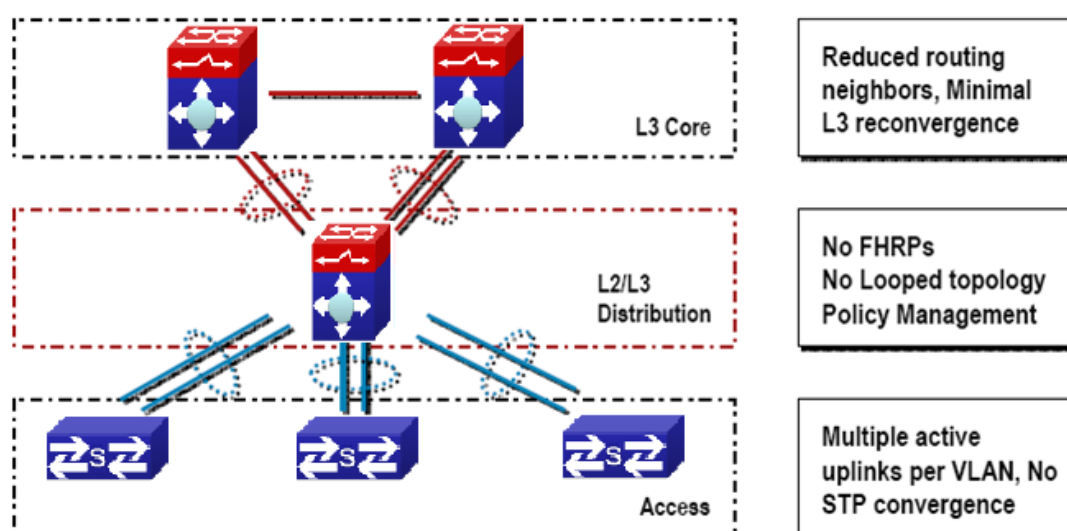


图 11-3 采用虚拟化技术的企业网

与传统的 L2/L3 网络设计相比，VSF 提供了多项显著优势。大体说来，其优势可归纳为以下三个主要方面：

1. VSF 能够提高运营效率

单管理点，包括配置文件和单一网关 IP 地址（无需 HSRP/VRRP/GLBP）

多台物理设备堆叠形成的一台逻辑设备创建了简单的无环路拓扑结构，不再依靠生成树协议（STP）

底层物理交换机经由标准万兆以太网接口相连，在位置方面提供了灵活的部署选项

2. VSF 能够优化不间断通信

机箱间状态化故障切换不会干扰需要使用网络状态信息的应用。凭借 VSF，在一个虚拟交换机成员发生故障时，不再需要进行 L2/L3 重收敛，能在较短时间内实现确定性虚拟交

换机的恢复。port group 的模式要求必须使用 active 模式。

3. VSF 能够大大扩展系统带宽容量

在 VSF 交换机上激活所有可用的 L2 带宽，在扩大带宽的同时，还可以在 VSF 多个成员上进行精确的负载均衡。

11.1.1.2 基本概念

(1) 角色

VSF 中每台设备都称为成员设备。成员设备按照功能不同，分为两种角色：

Master: 负责管理整个 VSF。

Standby Master: VSF 的备份成员，作为 Master 的备份设备运行，当 Master 故障时，系统由 Standby Master 自动接替原 Master 工作。

Slave: VSF 中除 Master 和 Standby Master 的成员设备。

Master、Standby Master 和 Slave 均由角色选举产生。一个 VSF 中同时只能存在一台 Master，一台 Standby Master，其它成员设备都是 Slave。

(2) VSF 端口

一种专用于 VSF 的逻辑接口，分为 vsf-port1 和 vsf-port2，需要和 VSF 物理端口绑定之后才能生效。

(3) VSF 物理端口

设备上可以用于 VSF 连接的物理端口。VSF 物理端口可能是 VSF 专用接口、以太网接口或者光口（设备上哪些端口可用作 VSF 物理端口与设备的型号有关，请以设备的实际情况为准）。通常情况下，以太网接口和光口负责向网络中转发业务报文，当它们与 VSF 端口绑定后就作为 VSF 物理端口，用于成员设备之间转发报文。可转发的报文包括 VSF 相关协商报文以及需要跨成员设备转发的业务报文。

(4) VSF 合并

两个 VSF 各自已经稳定运行，通过物理连接和必要的配置，形成一个 VSF，这个过程称为 VSF 合并（merge）。



图 11-4 VSF 合并

(5) VSF 分裂

一个 VSF 形成后，由于 VSF 链路故障，导致 VSF 中两相邻成员设备物理上不连通，一个 VSF 变成两个 VSF，这个过程称为 VSF 分裂（split）。

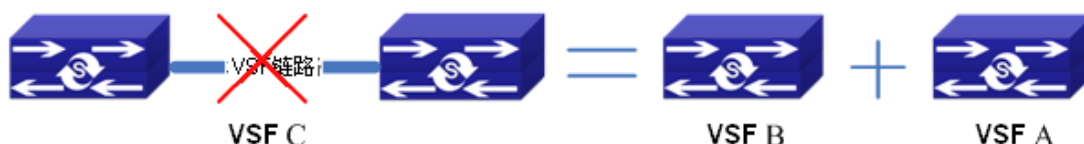


图 11-5 VSF 分裂

(6) 成员优先级

成员优先级是成员设备的一个属性，主要用于角色选举过程中确定成员设备的角色。优先级越高当选为 Master 的可能性越大。设备的缺省优先级均为 1，如果想让某台设备当选为 Master，则在组建 VSF 前，可以通过命令行手工提高该设备的成员优先级。

11.1.1.3 术语

VSF(Virtual Switching Framework):虚拟交换框架。

VSF Port: VSF 口，为一个 mode on 的 port-channel

VSF AM: VSF 中的活动主控卡

VSF SM: VSF 中的备份主控卡

VSF Slave: VSF 中的线卡，运行 VSF 协议，在机架交换机中指 Chassis AM，未被选为 VSF AM 或 VSF SM

VSF Member: VSF 中的成员设备

VSF Master Member: VSF 中的主成员设备

VSF Standby Master Member: VSF 中的备份主成员设备

VSF Slave Member: VSF 中的 slave 成员设备

HA(High Availability):高可靠性。

AM (Active Master): 活动主控，工作模式为 Master 且处于活动状态的交换机，一个盒式 vsf 组每一时刻只能有一个 AM。

SM (Standby Master): 备份主控，盒式 vsf 组中工作模式为 Master 且处于备份状态的交换机。

SSO(Stateful Switchover): 全状态主备倒换，同步协议应用层的数据，发生主备倒换，备份主控的协议应用模块能快速接管相应的功能，本模块的数据不丢失，但是路由协议的动态数据不同步。

NSF (None Stop Forwarding): 利用了数据层面控制层面处理的独立性，在控制层面发生故障的时候，数据层面依然能够正常转发数据。

NSF/SSO(Nonstop Forwarding with Stateful Switchover): 指在进行 SSO“主备倒换”的时候，数据的转发是不受影响的。

Syn process: HA 同步进程，负责活动主控或备份主控协议模块数据的收集或分发，负责活动主控和备份主控的数据同步通信。

GR(Graceful Restart): 优雅重启。为了实现不间断转发，需要路由协议做扩展以支持 GR 能力。

MAD: Multi-Active Detection，多 Active 检测。

11.1.1.4 VSF 典型应用

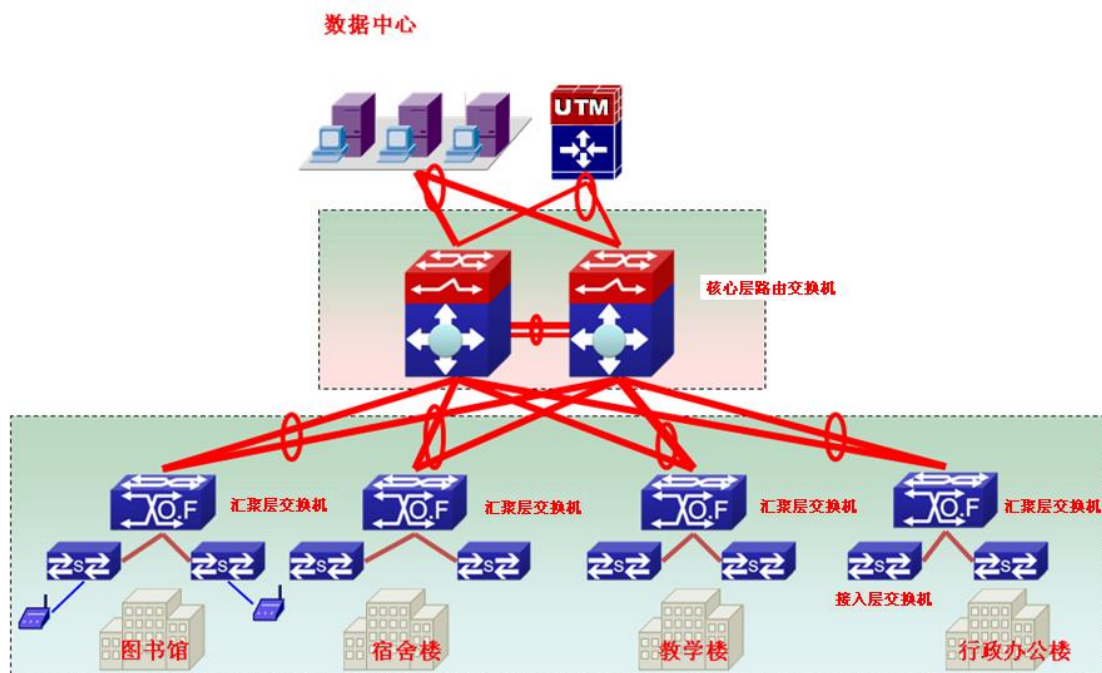


图 11-6 VSF 在校园网数据中心应用

图 1-6 为 VSF 在校园网中的应用，使用 VSF 后，汇聚层的多个设备成为了一个单一的逻辑设备，接入设备直接连接到虚拟设备。这个简化后的组网不再需要使用 MSTP、VRRP 协议，简化了网络配置。同时依靠跨设备的链路聚合（port group 的模式要求必须使用 active 模式），在成员出现故障时不再依赖 MSTP、VRRP 等协议的收敛，提高了可靠性。

11.1.1.5 LACP MAD

lacp mad 是基于 lacp 的动态聚合方式，vsf 的每个成员设备都至少有一个端口和中间设备连接。

注意：中间设备必须为我司支持 lacp 扩展功能的设备。

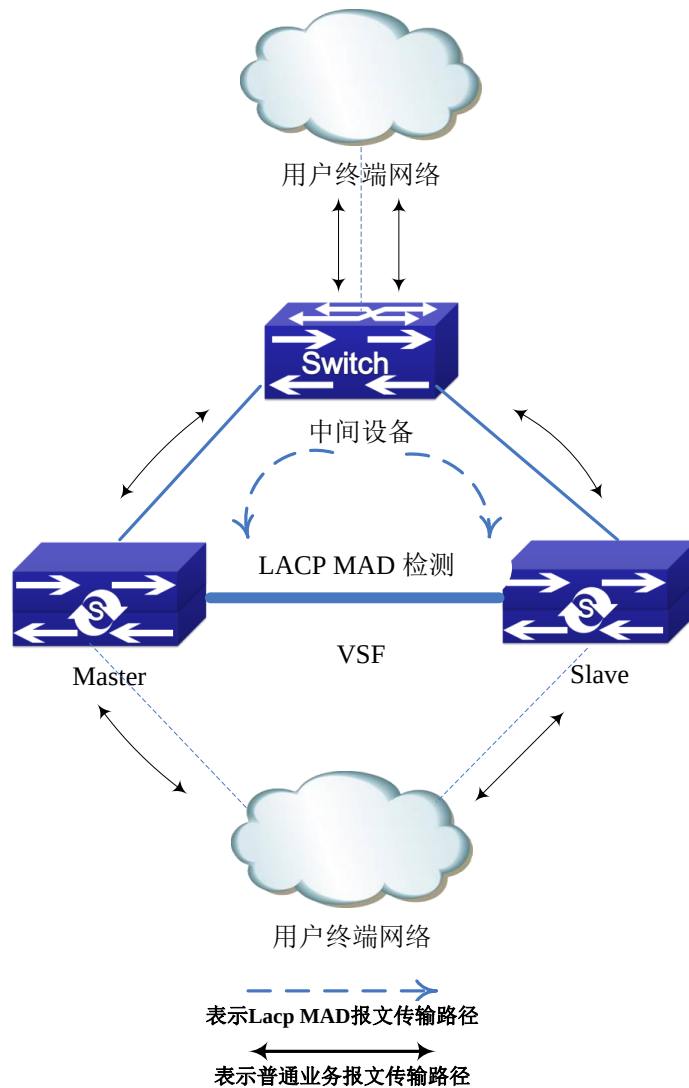


图 11-7 LACP MAD 检测

LACP MAD 检测是通过扩展 LACP 协议报文内容实现的，在 LACP 协议报文的扩展字段中定义一个新的 TLV (Type Length Value)，该 TLV 用于交互 VSF 的 ActiveID。对于 VSF 系统来说，ActiveID 的值是唯一的，用 VSF 中 master 设备的成员编号来表示。使能 LACP MAD 检测后，成员设备通过 LACP 协议报文和其它成员设备交互 ActiveID 信息。

当 VSF 正常运行时，所有成员设备发送的 LACP 协议报文中的 **ActiveID** 值相同，没有发生多 Active 冲突；

当 VSF 分裂后会形成两个 vsf 时,不同 VSF 中的成员设备发送的 LACP 协议报文中的 ActiveID 值不同,从而检测到多 Active 冲突。

11.1.1.6 BFD MAD

Bfd mad 检测的拓扑搭建比 lacp 简单，但是一旦某个 vlan 被选定用于 bfd mad 检测，该 vlan 以及 vlan 中的端口都将作为 bfd mad 的专有 vlan 和端口，不能再配置使能其它的功能。

搭建方法：在 member1 上选定一个端口，在 member2 选定一个端口，在两者之间连一根线。



图 11-8 BFD MAD 检测

BFD MAD 检测是通过 BFD 协议来实现的。要使 BFD MAD 检测功能正常运行，除在三层接口下使能 BFD MAD 检测功能外，还需要在该接口上配置 MAD IP 地址。MAD IP 地址与普通 IP 地址不同的地方在于 MAD IP 地址与成员设备是绑定的，VSF 中的每个成员设备上都需要配置，且必须属于同一网段。

当 VSF 正常运行时，只有 master 上配置的 MAD IP 地址生效，配置的和 Slave 设备关联的 MAD IP 地址不生效，BFD 会话处于 down 状态；

当 VSF 分裂后形成两个 VSF 时，新形成的 vsf 上与此 member 关联的 mad ip 地址生效，通过 bfd mad 检测的连接线，两个 vsf 之间会建立起 bfd 会话，从而检测到多 Active 冲突。

11.1.2 VSF 相关配置

11.1.2.1 VSF 配置

VSF 配置任务序列：

- 5. 配置 VSF 成员编号（必选）
- 6. 配置 VSF 成员优先级（可选）
- 7. 配置 VSF 域（可选）
- 8. 配置逻辑 VSF 口
 - (1) 配置逻辑 VSF 口
 - (2) 将物理口与逻辑口绑定
- 9. 设备由独立运行模式转换到 VSF 运行模式
- 10. 配置 VSF 自动合并（可选）
- 11. 对 VSF 成员进行描述（可选）
- 12. 配置 VSF 链路 down 延迟上报功能（可选）
- 13. 配置 VSF 分裂后 VSF 组 MAC 地址保留时间（可选）
- 14. 设备由 VSF 运行模式转换到独立运行模式
- 15. 快速检测 VSF 链路状态变化

5. 配置 VSF 成员编号（必选）

命令	解释
全局配置模式	
vsf member <member-id> no vsf member <member-id>	配置或删除 VSF 成员编号

6. 配置 VSF 成员优先级和所在 VSF 域（可选）

命令	解释
全局配置模式	
vsf priority <priority> no vsf priority	配置或删除 VSF 成员优先级。
vsf domain <domain-id> no vsf domain	配置或删除 VSF 域，no 操作相当于恢复到默认域号 1。

7. 配置逻辑 VSF 口并与物理口进行绑定

命令	解释
全局配置模式	
vsf port-group <port-number> no vsf port-group <port-number>	配置或删除逻辑 VSF 口
VSF 口配置模式	
vsf port-group interface Ethernet <interface-list> no vsf port-group interface Ethernet <interface-list>	将物理口与逻辑 VSF 口进行绑定或删除绑定

8. 设备进行运行模式转换

命令	解释
全局配置模式	
switch convert mode (stand-alone vsf)	设置设备由独立运行模式转换到 VSF 运行模式，或者由 VSF 运行模式转换到独立运行模式

9. 其他配置

命令	解释
全局配置模式	
vsf auto-merge enable no vsf auto-merge enable	使能 VSF 组自动合并功能，该命令的 no 命令去除自动合并功能。
vsf member <member-id> description <text> no vsf member <member-id> description	对 VSF 成员进行描述，此描述信息只写入 VSF 主控配置文件中。该命令的 no 命令为删除对应 VSF 成员的描述信息。
vsf link delay<interval> no vsf link delay	配置 VSF 链路 down 延迟上报功能，用于避免因端口链路层状态在短时间内频繁改变，导致 VSF 分裂、合并的频繁发生。该命令的 no 命令为将延迟上报时间值恢复为默认值。

vsf mac-address persistent <timer always> no vsf mac-address persistent	配置 VSF 分裂后 VSF 组 MAC 地址保留时间。该命令的 no 命令为删除 VSF 组 MAC 地址保留时间的配置，即不保留。
vsf non-wait port-inactive no vsf non-wait port-inactive	快速检测 VSF 链路状态变化，用于需要快速发现 vsf 分裂的场景。该命令的 no 命令将恢复检测 VSF 链路状态为默认方式。

11.1.2.2 LACP MAD 配置

LACP MAD 配置任务序列：

1. 创建聚合组
2. 将端口加入聚合组，至少有一方为 active 方式
3. 配置快速检测（可选）
4. 使能 LACP MAD

1. 创建聚合组

命令	解释
全局配置模式	
port-group <port-group id> no Port-group <port-group id>	创建或删除 port-group

2. 将端口加入聚合组

命令	解释
全局配置模式	
interface ethernet <port-num>	进入端口模式。
端口配置模式	
port-group <port-group id> mode <active passive> no port-group <port-group id>	将端口加入或退出聚合组

3. 配置快速检测

命令	解释
端口配置模式	
lacp timeout <long short> no lacp timeout	配置或删除快速检测

4. 使能 LACP MAD

命令	解释
聚合端口配置模式	

vsf mad lacp <enable disable>	使能或去使能 port-group 上的 LACP MAD

11.1.2.3 BFD MAD 配置

BFD MAD 配置任务序列:

1. 创建用于 BFD MAD 的 vlan
2. 将用于进行 BFD MAD 的端口加入到相应 vlan 中
3. 为 BFD MAD 三层接口配置 IP 地址
4. 使能 BFD MAD 功能

1. 创建用于 BFD MAD 的 vlan

命令	解释
全局配置模式	
vlan <vlan-id> no vlan <vlan-id>	配置或删除 vlan

2. 将用于进行 BFD MAD 的端口加入到相应 vlan 中

命令	解释
vlan 配置模式	
switchport interface ethernet <port-num> no switchport interface ethernet <port-num>	将端口加入或移出 vlan。

3. 为 BFD MAD 三层接口配置 IP 地址

命令	解释
全局配置模式	
Interface vlan <vlan-id>	进入 vlan 接口配置模式
接口配置模式	
vsf mad ip address <ip-addr> <ip-mask> member <member-id> no vsf mad ip address <ip-addr> <ip-mask> member <member-id>	配置或删除三层接口上用于 BFD MAD 的 IP 地址

4. 使能 BFD MAD 功能

命令	解释
接口配置模式	
vsf mad bfd <enable disable>	使能或去使能 BFD MAD

11.1.3 VSF 典型案例

案例 1:

在独立运行模式下进行配置，令两台交换机形成 VSF，两台设备分的 VSF 成员编号分别为 1 和 2，为了让 vsf member2 成为 vsf master，配置 member 2 的成员优先级为 32，两台设备之间建立两个 vsf port-group，每个 vsf port-group 绑定一个万兆端口。

switch1 的 VSF 配置如下：

```
switch1#config
switch1(config)#vsf member 1
switch1(config)#vsf port-group 1
switch1(config-vsf-port1)#vsf port-group interface ethernet 1/1
switch1(config)#vsf port-group 2
switch1(config-vsf-port1)#vsf port-group interface ethernet 1/2
switch1(config)#exit
switch1(config)#switch convert mode vsf
```

switch2 的 VSF 配置如下：

```
switch2#config
switch1(config)#vsf member 2
switch1(config)#vsf priority 32
switch1(config)#vsf port-group 1
switch1(config-vsf-port1)#vsf port-group interface ethernet 2/1
switch1(config)#vsf port-group 2
switch1(config-vsf-port1)#vsf port-group interface ethernet 2/2
switch1(config)#exit
switch1(config)#switch convert mode vsf
```

案例 2:

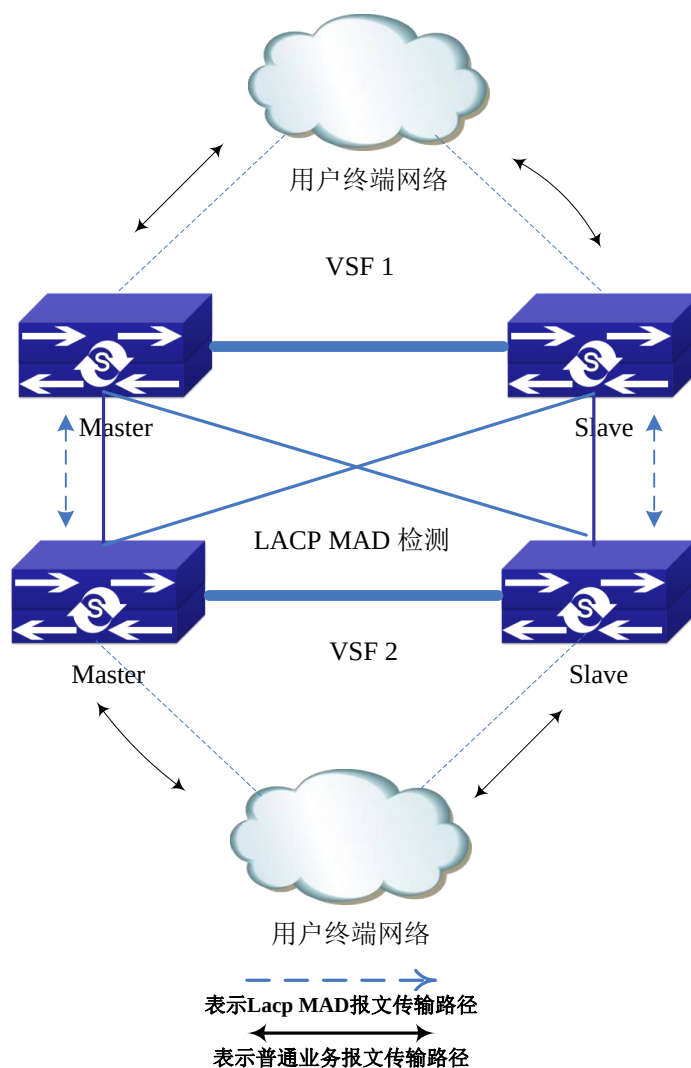


图 11-9 lacp mad 检测典型拓扑

按上图所示典型拓扑，在两个 vsf 之间使用 lacp mad 检测功能，vsf1 与 vsf2 的角色既为被检测设备，也互为中间设备。配置和前面基本相同。建议用户在各个设备之间交叉连接，避免 vsf1 分裂后不能做中间设备继续检测 vsf2。

假设 vsf1 与 vsf2 使用的 lacp 端口都为 ethernet 1/1/1, ethernet 1/1/2, ethernet 2/1/1, ethernet 2/1/2。

vsf1 的配置如下：

```
Switch(config)#vsf domain 1
```

*配置 vsf 域号，可以配置成其它值，但是不能与 vsf2 相同。

```
Switch(config)#port-group 1
```

```
Switch(config)#interface ethernet 1/1/1
```

```
Switch(config-if-ethernet1/1/1)#port-group 1 mode active
```

```
Switch(config)#interface ethernet 1/1/2
```

```
Switch(config-if-ethernet1/1/2)#port-group 1 mode active
```

```
Switch(config)#interface ethernet 2/1/1
```

```
Switch(config-if-ethernet2/1/1)#port-group 1 mode active
Switch(config)#interface ethernet 2/1/2
Switch(config-if-ethernet2/1/2)#port-group 1 mode active
Switch(config-if-ethernet2/1/2)#interface port-channel 1
Switch(config-if-port-channel1)#vsf mad lacp enable
```

vsf2 的配置如下：

```
Switch(config)#vsf domain 2
```

*配置 vsf 域号，可以配置成其它值，但是不能与 vsf1 相同。

其余配置和 vsf1 上的配置相同。

案例 3：



如上图所示，假设用于 bfd mad 检测的端口为 ethernet1/1/1, ethernet2/1/1, vlan 为 3000。Mad ip 地址的配置以 192.168.1.1 网段为例。

switch 的 BFD MAD 配置如下：

```
Switch(config)#vlan 3000
Switch(config-vlan3000)#interface ethernet 1/1/1
Switch(config-if-ethernet1/1/1)#switchport access vlan 3000
Switch(config-if-ethernet1/1/1)#interface ethernet 2/1/1
Switch(config-if-ethernet2/1/1)#switchport access vlan 3000
Switch(config-if-ethernet2/1/1)#interface vlan 3000
Switch(config-if-vlan3000)#vsf mad bfd enable
Switch(config-if-vlan3000)#vsf mad ip add 192.168.1.1 255.255.255.0 member 2
Switch(config-if-vlan3000)#vsf mad ip add 192.168.1.2 255.255.255.0 member 1
```

注意：此时整个 vsf 是一台设备，所以 BFD MAD 配置相当于只在一台设备上进行。

11.1.4 VSF 排错帮助

在进行 VSF 配置和使用过程中，出现命令行不可配置时，应注意如下事项：

- ☞ 是否处于正确的运行模式，有些命令行既可在独立运行模式下配置，又可在 VSF 运行模式下配置，而某些命令行只能最 VSF 运行模式下配置；
- 在 VSF 使用过程中不能形成 VSF，或出现其他异常时，应注意如下事项：
 - ☞ 首先查看物理连接是否正确。目前机架式 VSF 组只支持万兆口与逻辑口的绑定，需要查看连接的万兆端口是否是绑定的端口；

- ✎ Vsf member id 是否冲突，member id 冲突时，两设备无法形成 vsf。
- ✎ Vsf domain 号是否相同，只有 vsf domain 号相同的设备才有可能形成 vsf。
- ✎ 与逻辑 VSF 口绑定的物理端口上建议没有任何配置，尤其是类似于速率双工、带宽限制、安全认证、ACL 等配置。

在 VSF 运行模式下进行配置时，应注意如下事项：

- ✎ VSF 运行模式下，VSF 相关配置可以在各个成员上分别配置，但在成员上所做的配置，无法在成员上单独进行保存，但可以在 vsf master 上执行 write，此时所有成员上关于 vsf 的个性配置，分别各自保存在本机的 vsf.cfg 中。
- ✎ VSF 运行模式下，部分命令，如 VSF 域编号、成员优先级、member id 等仍然能够进行配置或修改，配置后 show run 显示为最新配置值，但需要在保存并重启后方能生效。
- ✎ VSF 运行模式下，为保证故障收敛时间，port group 的模式要求必须使用 active 模式。

对于 LACP MAD，在配置和使用过程中，出现疑问或异常时，应注意如下事项：

- ✎ 创建聚合组时，被检测设备与中间设备的聚合组必须保持一致
- ✎ 被检测设备和中间设备端口不能以 on 模式加入到用于 LACP MAD 的聚合组，且至少有一方为 active 方式
- ✎ 当被检测设备和中间设备都是 vsf 时，两 vsf 的 domain id(vsf 域号)不能相同

对于 BFD MAD，在配置和使用过程中，出现疑问或异常时，应注意如下事项：

- ✎ bfd mad 功能不能在 interface vlan 1 下使能
- ✎ bfd mad 功能与防环功能互斥(比如 stp,mstp 等等)，如果该 vsf 有配置防环功能，需要在用于做 bfd mad 检测的端口上都关闭防环功能，否则，可能导致检测失败。
- ✎ 如果该 vsf 上存在端口被配置成 trunk 口(包括 port-channel 口)，请确认用于 bfd mad 检测的 vlan 是否在该 trunk 内(trunk 口默认属于所有 vlan)，如果在，必须要在该端口下将用于 bfd mad 检测的 vlan 过滤掉，否则，可能导致出现环路。
- ✎ 如果该 vsf 上存在端口被配置成 hybrid 口，请确保该 hybrid 口不属于 bfd mad 检测的 vlan,否则，也可能导致出现环路。
- ✎ 在用于 bfd mad 检测的 vlan 接口以及该 vlan 中的所有端口，除了配置用于 bfd mad 检测的配置外，不能再配置其它配置。
- ✎ 如果不再使用 bfd mad 功能，除了将配置清除外，还要注意将用于 bfd mad 检测的连接线去掉。
- ✎ 如果 bfd mad 配置已经完成，后来又对其它端口，vlan 或者全局下的命令有修改，请确认以上几条是否满足。
- ✎ 建议用户选择一个在业务处理中基本不会用到的 vlan，作为 bfd mad 检测的专用 vlan。

第12章 IPV6 业务操作

12.1 DHCPv6

12.1.1 DHCPv6 简介

DHCPv6 是动态主机配置协议（DHCP）的 IPv6 版本，协议基本规范由 RFC3315 定义。DHCPv6 服务端可以向 DHCPv6 客户端分配 IPv6 地址以及 DNS 地址和域名等配置参数。相对于 IPv6 无状态地址自动配置协议，DHCPv6 属于一种有状态地址自动配置协议。在有状态地址配置过程中，DHCPv6 服务器分配一个完整的 IPv6 地址给主机，并提供 DNS 地址和域名等其他配置信息，这中间可能通过中继代理转交 DHCPv6 报文，而且最终服务器能把分配的 IPv6 地址和客户端的绑定关系记录在案，从而增强了网络的可管理性；DHCPv6 服务器也能提供无状态 DHCPv6 服务，即不分配 IPv6 地址，仅需向主机提供 DNS 地址和域名等其他配置信息，从而补足了 IPv6 无状态地址自动配置的缺陷；DHCPv6 协议还提供了 DHCPv6 前缀代理的扩展功能，上游路由器可以自动为下游路由器分派地址前缀，从而实现了层次化网络环境中 IPv6 地址的自动规划，解决互联网提供商（ISP）的 IPv6 网络部署问题。

DHCPv6 报文用于客户端、中继代理与服务器三者之间的通信，以完成 DHCPv6 的协议行为。DHCPv6 报文承载于 UDP 协议之上。DHCPv6 客户端使用 UDP 端口 547 向 DHCPv6 服务器或中继代理发送请求报文，DHCPv6 服务器和中继代理使用 UDP 端口 546 向 DHCPv6 客户端发送应答报文。DHCPv6 客户端使用所有 DHCP 中继代理和服务器多播地址(FF02::1:2)发送服务器发现(Solicit)或请求(Request)报文。协议实现的过程如下：

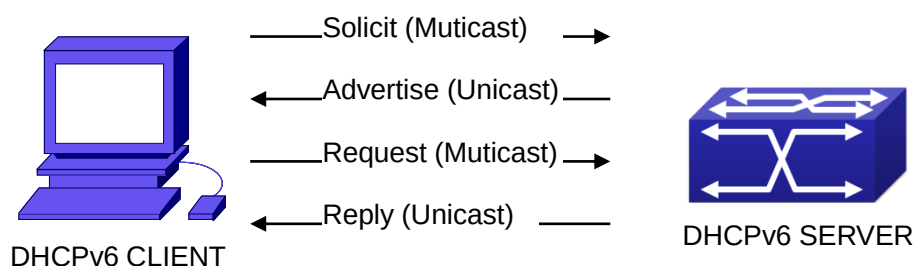


图 12-1 DHCPv6 协议交互过程

当 DHCPv6 客户端向位于同一链路的 DHCPv6 服务器请求分配 IPv6 地址以及其它配置参数时，客户端首先要定位和选择合适的服务器，然后才能向服务器请求分配地址和其它配置参数。

1. 在定位服务器时，客户端向所有 DHCP 中继代理和服务器多播地址 (FF02::1:2) 发出 SOLICIT 消息希望发现合适的 DHCPv6 服务器。
2. 任何能够满足该请求的服务器都会发送一个 ADVERTISE 消息给客户端，其中包

含可分配的配置信息以及服务器的身份标识(DUID)和该服务器的优先值信息。

- 3. 客户端可能接收到多个这样的通告消息，依据某种策略(如选择服务器优先值选项中优先值最高的服务器)选择其中一个服务器，然后向选定服务器发送 REQUEST 消息请求服务器分配通告的 IPv6 地址和其它配置参数。
- 4. 被选择的服务器回应以 REPLY 消息，其中包含确认的 IPv6 地址和其它配置参数，客户端收到 REPLY 消息提取地址和配置参数完成 DHCPv6 配置任务。

通过上面四个步骤，完成动态分配主机配置的协议过程。但如果 DHCPv6 服务器和 DHCPv6 客户机不相邻时，服务器是无法收到客户机发出的 DHCPv6 多播报文，因此服务器也不会给客户机发送任何 DHCPv6 报文，这时需要 DHCPv6 中继代理来转发这些 DHCPv6 报文，完成 DHCPv6 客户机和服务器之间的 DHCPv6 报文交互过程。

交换机实现了 DHCPv6 服务器、DHCPv6 中继的功能和 DHCPv6 前缀代理客户端功能。中继代理节点收到来自客户端的 DHCPv6 消息，重新封装成 Relay-forward 报文转发给 DHCPv6 服务器或下一个中继代理，服务器回应的 DHCPv6 消息被封装成 Relay-reply 报文交给 DHCPv6 中继代理节点，中继代理节点解除最外层中继封装后再转发给 DHCPv6 客户端或下一个中继代理，从而完成客户端与服务器之间的通信。

DHCPv6 前缀代理(Prefix Delegation)机制中，上游网络路由器(PE)启动 DHCPv6 服务器，下游网络路由器(CPE)启动 DHCPv6 前缀代理客户端。此时下游网络路由器不需要再手工指定用户侧链路的 IPv6 地址前缀，它只需要向上游网络路由器提出前缀分配申请，上游网络路由器便可以分配合适的地址前缀给下游路由器。二者之间的报文交互与 DHCPv6 地址分配的过程类似。然后下游路由器把获得的前缀(一般前缀长度小于 64)进一步自动细分成 64 前缀长度的子网网段，把细分的地址前缀再通过路由公告(RA)至与主机直连的用户链路上，实现主机的地址自动配置，从而完成整个系统层次的地址布局。

12.1.2 DHCPv6 服务器配置

DHCPv6 服务器配置任务序列如下：

- 1. 启动/关闭 DHCPv6 服务
- 2. 配置 DHCPv6 地址池
 - (1) 创建/删除 DHCPv6 地址池
 - (2) 配置 DHCPv6 地址池的参数
- 3. 在接口上启动 DHCPv6 服务器功能

1.启动/关闭 DHCPv6 服务

命令	解释
全局配置模式	
service dhcpv6 no service dhcpv6	启动 DHCPv6 服务。

2.配置 DHCPv6 地址池

(1) 创建/删除 DHCPv6 地址池

命令	解释
全局配置模式	
ipv6 dhcp pool <poolname> no ipv6 dhcp pool <poolname>	配置 DHCPv6 地址池。

(2) 配置 DHCPv6 地址池的参数

命令	解释
DHCPv6 地址池配置模式	
network-address <ipv6-pool-start-address> {<ipv6-pool-end-address> <prefix-length>} [eui-64] no network-address	配置地址池可分配的 IPv6 地址范围。
dns-server <ipv6-address> no dns-server <ipv6-address>	为 DHCPv6 客户机配置 DNS 服务器地址。
domain-name <domain-name> no domain-name <domain-name>	为 DHCPv6 客户机配置域名。
excluded-address <ipv6-address> no excluded-address <ipv6-address>	排除地址池中不用于动态分配的 IPv6 地址。
lifetime {<valid-time> infinity} {<preferred-time> infinity} no lifetime	配置 DHCPv6 地址池动态分配地址或前缀的生存期。

3. 在接口上启动 DHCPv6 服务器功能

命令	解释
接口配置模式	
ipv6 dhcp server <poolname> [preference<value>] [rapid-commit] [allow-hint] no ipv6 dhcp server <poolname>	在指定接口上启动 DHCPv6 服务器功能，并绑定使用的 DHCPv6 地址池。

12.1.3 DHCPv6 中继代理配置

DHCPv6 中继代理配置任务序列如下：

1. 启动/关闭 DHCPv6 服务
2. 在接口上配置 DHCPv6 中继代理

1.启动 DHCPv6 服务

命令	解释

全局配置模式	
service dhcpv6 no service dhcpv6	启动 DHCPv6 服务。

2. 在接口上配置 DHCPv6 中继代理

命令	解释
接口配置模式	
ipv6 dhcp relay destination {[<ipv6-address>] [interface {<interface-name> vlan <1-4096>}]} no ipv6 dhcp relay destination {[<ipv6-address>] [interface { <interface-name> vlan <1-4096>}]}	指定 DHCPv6 中继转发的目的地址；本命令的 no 操作为取消该项配置。

12.1.4 DHCPv6 前缀代理服务器端配置

DHCPv6 前缀代理服务器端配置任务序列如下：

1. 启动/关闭 DHCPv6 服务
2. 配置代理前缀池
3. 配置 DHCPv6 地址池
 - (1) 创建/删除 DHCPv6 地址池
 - (2) 配置 DHCPv6 地址池使用的代理前缀池
 - (3) 配置静态代理前缀绑定
 - (4) 配置 DHCPv6 地址池其它参数
4. 在接口上启动 DHCPv6 前缀代理服务器功能

1. 启动/关闭 DHCPv6 服务

命令	解释
全局配置模式	
service dhcpv6 no service dhcpv6	启动 DHCPv6 服务。

2. 配置代理前缀池

命令	解释
全局配置模式	
ipv6 local pool <poolname> <prefix/prefix-length> <assigned-length> no ipv6 local pool <poolname>	配置代理前缀池。

3. 配置 DHCPv6 地址池

(1) 创建/删除 DHCPv6 地址池

命令	解释
全局配置模式	
ipv6 dhcp pool <poolname> no ipv6 dhcp pool <poolname>	配置 DHCPv6 地址池。

(2) 配置 DHCPv6 地址池使用的代理前缀池

命令	解释
DHCPv6 地址池配置模式	
prefix-delegation pool <poolname> [lifetime {<valid-time> infinity} {<preferred-time> infinity}] no prefix-delegation pool <poolname>	指定 IDHCPv6 地址池使用的代理前缀池， 从该前缀池中分配可用的前缀给客户端。

(3) 配置静态代理前缀绑定

命令	解释
DHCPv6 地址池配置模式	
prefix-delegation <ipv6-prefix/prefix-length> <client-DUID> [iaid <iaid>] [lifetime {<valid-time> infinity} {<preferred-time> infinity}] no prefix-delegation <ipv6-prefix/prefix-length> <client-DUID> [iaid <iaid>]	指定 IPv6 前缀与某个前缀请求客户端静态绑定。

(4) 配置 DHCPv6 地址池其他参数

命令	解释
DHCPv6 地址池配置模式	
dns-server <ipv6-address> no dns-server <ipv6-address>	为 DHCPv6 客户机配置 DNS 服务器地址。
domain-name <domain-name> no domain-name <domain-name>	为 DHCPv6 客户机配置域名。

4. 在接口上启动 DHCPv6 服务器功能

命令	解释
接口配置模式	

ipv6 dhcp server <poolname> [preference <value>] [rapid-commit] [allow-hint] no ipv6 dhcp server <poolname>	在指定接口上启动 DHCPv6 服务器功能，并绑定使用的 DHCPv6 地址池。

12.1.5 DHCPv6 客户端配置

DHCPv6 客户端配置任务序列如下：

1. 启动/关闭 DHCPv6 服务
2. 在接口上启动 DHCPv6 客户端功能

1.启动/关闭 DHCPv6 服务

命令	解释
全局配置模式	
service dhcpv6 no service dhcpv6	启动 DHCPv6 服务。

2. 在接口上启动 DHCPv6 客户端功能

命令	解释
接口配置模式	
ipv6 dhcp client { rapid-commit <cr>} no ipv6 dhcp client	本交换机不支持此命令。

12.1.6 DHCPv6 前缀代理客户端配置

DHCPv6 前缀代理客户端配置任务序列如下：

3. 启动/关闭 DHCPv6 服务
4. 在接口上启动 DHCPv6 前缀代理客户端功能

1.启动/关闭 DHCPv6 服务

命令	解释
全局配置模式	
service dhcpv6 no service dhcpv6	启动 DHCPv6 服务。

2. 在接口上启动 DHCPv6 前缀代理客户端功能

命令	解释
接口配置模式	

<pre>ipv6 dhcp client pd <prefix-name> [rapid-commit] no ipv6 dhcp client pd</pre>	在指定接口上启动客户端前缀代理请求功能，获取的前缀与设置的通用前缀名关联。

12.1.7 DHCPv6 配置举例

案例 1:

在部署全 IPv6 校园网时，若使用专门的服务器进行 IPv6 地址的统一分配和管理，则可以使用路由交换设备的 DHCPv6 服务器功能进行 IPv6 地址分派。DHCPv6 服务器同时支持有状态与无状态 DHCPv6。

网络拓扑:

接入层使用二层接入设备 Switch1，连接宿舍楼栋的用户；一级汇聚层使用汇聚设备 Switch2 做 DHCPv6 中继代理；二级汇聚层使用汇聚设备 Switch3 做 DHCPv6 服务器，并与骨干网或更高的汇聚层设备相连；用户 PC 端一般装有 Windows Vista 操作系统，具有 DHCPv6 客户端。

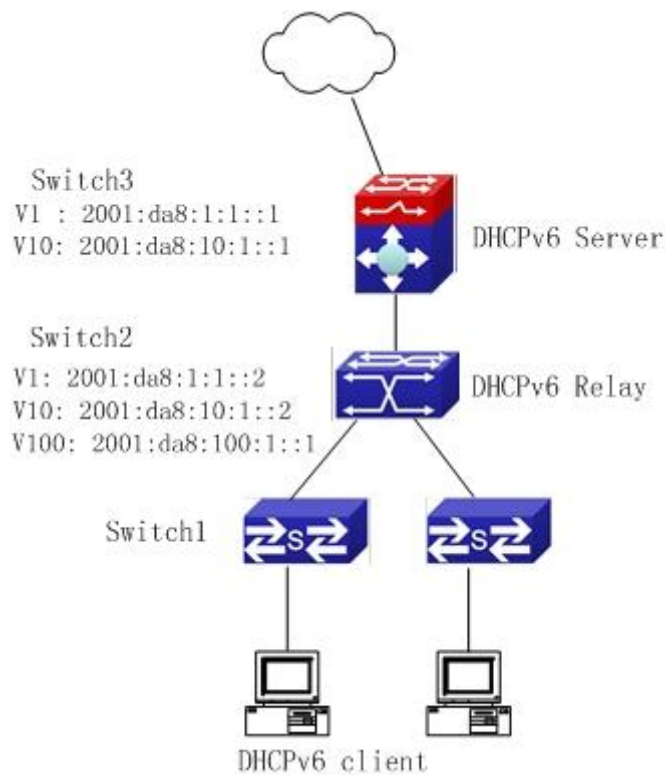


图 12-2 DHCPv6 配置举例示意图 1

使用提示:

Switch3 配置:

Switch3>enable

Switch3#config

```
Switch3(config)#service dhcpv6
Switch3(config)#ipv6 dhcp pool EastDormPool
Switch3(dhcpv6-EastDormPool-config)#network-address 2001:da8:100:1::1
2001:da8:100:1::100
Switch3(dhcpv6-EastDormPool-config)#excluded-address 2001:da8:100:1::1
Switch3(dhcpv6-EastDormPool-config)#dns-server 2001:da8::20
Switch3(dhcpv6-EastDormPool-config)#dns-server 2001:da8::21
Switch3(dhcpv6-EastDormPool-config)#domain-name dhcpv6.com
Switch3(dhcpv6-EastDormPool-config)#lifetime 1000 600
Switch3(dhcpv6-EastDormPool-config)#exit
Switch3(config)#interface vlan 1
Switch3(Config-if-Vlan1)#ipv6 address 2001:da8:1:1::1/64
Switch3(Config-if-Vlan1)#exit
Switch3(config)#interface vlan 10
Switch3(Config-if-Vlan10)#ipv6 address 2001:da8:10:1::1/64
Switch3(Config-if-Vlan10)#ipv6 dhcp server EastDormPool preference 80
Switch3(Config-if-Vlan10)#exit
Switch3(config)#
```

Switch2 配置:

```
Switch2>enable
Switch2#config
Switch2(config)#service dhcpv6
Switch2(config)#interface vlan 1
Switch2(Config-if-Vlan1)#ipv6 address 2001:da8:1:1::2/64
Switch2(Config-if-Vlan1)#exit
Switch2(config)#interface vlan 10
Switch2(Config-if-Vlan10)#ipv6 address 2001:da8:10:1::2/64
Switch2(Config-if-Vlan10)#exit
Switch2(config)#interface vlan 100
Switch2(Config-if-Vlan100)#ipv6 address 2001:da8:100:1::1/64
Switch2(Config-if-Vlan100)#no ipv6 nd suppress-ra
Switch2(Config-if-Vlan100)#ipv6 nd managed-config-flag
Switch2(Config-if-Vlan100)#ipv6 nd other-config-flag
Switch2(Config-if-Vlan100)#ipv6 dhcp relay destination 2001:da8:10:1::1
Switch2(Config-if-Vlan100)#exit
Switch2(config)#
```

案例 2:

运营商在部署层次化的 IPv6 网络时，不再需要为网络下游的路由交换设备手工配置 IPv6 地址，可以通过前缀代理机制，实现层次化 IPv6 网络的自动配置：

1. 配置上游路由交换设备为 DHCPv6 前缀代理服务器，维护一个保存已分配前缀与下游设备 DUID 关联信息的本地数据库；
2. 配置下游路由交换设备为前缀代理客户端，通过与前缀代理服务器进行类似 DHCPv6 地址分配相似的交互，获得长度小于 64 的 IPv6 地址前缀；
3. 获得前缀的下游接入设备与用户 PC 处于同一链路的接口通过路由公告(RA)告知接入用户所在链路的前缀信息，用户的 PC 机进一步通过无状态地址自动配置获取 IPv6 地址；同时，该接口配置无状态 DHCPv6 服务器，为 DHCPv6 client 提供 DNS、Domain name 信息。

网络拓扑：

下游路由交换设备使用接入设备 Switch1，与上游路由交换设备相连的接口配置前缀代理客户端，与用户 PC 处于同一链路的接口配置无状态 DHCPv6 服务器，为 PC 提供无状态信息（如 DNS、domain name 等），并开启无状态地址自动配置的路由公告；上游路由交换设备使用汇聚设备 Switch2，配置前缀代理服务器，开启有状态地址自动配置的路由公告；用户 PC 端一般装有 Windows Vista 操作系统，具有 DHCPv6 客户端。

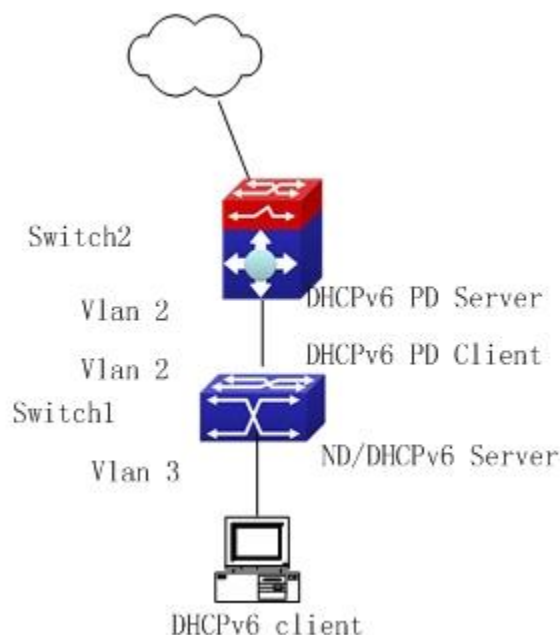


图 12-3 DHCPv6 配置举例示意图 2

使用提示：

Switch2 配置

```
Switch2>enable
```

```
Switch2#config
```

```
Switch2(config)#interface vlan 2
```

```
Switch2(Config-if-Vlan2)#ipv6 address 2001:da8:1100::1/64
```

```
Switch2(Config-if-Vlan2)#exit
```

```
Switch2(config)#service dhcpv6
```

```
Switch2(config)#ipv6 local pool client-prefix-pool 2001:da8:1800::/40 48
Switch2(config)#ipv6 dhcp pool dhcp-pool
Switch2(dhcpv6-dhcp-pool-config)#prefix-delegation pool client-prefix-pool 1800 600
Switch2(dhcpv6-dhcp-pool-config)#exit
Switch2(config)#interface vlan 2
Switch2(Config-if-Vlan2)#ipv6 dhcp server dhcp-pool
Switch2(Config-if-Vlan2)#exit
```

Switch1 配置

```
Switch1>enable
Switch1#config
Switch1(config)#service dhcpv6
Switch1(config)#interface vlan 2
Switch1(Config-if-Vlan2)#ipv6 dhcp client pd prefix-from-provider
Switch1(Config-if-Vlan2)#exit
Switch1(config)#interface vlan 3
Switch1(Config-if-Vlan3)#ipv6 address prefix-from-provider 0:0:0:1::1/64
Switch1(Config-if-Vlan3)#exit
Switch1(config)#ipv6 dhcp pool foo
Switch1(dhcpv6-foo-config)#dns-server 2001:4::1
Switch1(dhcpv6-foo-config)#domain-name www.ipv6.org
Switch1(dhcpv6-foo-config)#exit
Switch1(config)#interface vlan 3
Switch1(Config-if-Vlan3)#ipv6 dhcp server foo
Switch1(Config-if-Vlan3)#ipv6 nd other-config-flag
Switch1(Config-if-Vlan3)#no ipv6 nd suppress-ra
Switch1(Config-if-Vlan3)#exit
```

12.1.8 DHCPv6 排错帮助

DHCPv6 客户机无法获得 IPv6 地址及其它网络参数, 在确保 DHCPv6 客户机硬件、线缆等没有问题的前提下, 检查可能存在的情况和建议的解决方法:

- ☞ 首先检查全局 DHCPv6 服务是否已启动, 若没有启动, 请启动全局 DHCPv6 服务与相关接口上的 DHCPv6 功能;
- ☞ 若 DHCPv6 客户机和服务器不在同一个物理网络中, 检查中间负责转发 DHCPv6 报文的路由器是否具备 DHCPv6 中继功能。若中间路由器不具备 DHCPv6 中继功能, 建议替换掉中间的路由器或更新版本, 使其具备 DHCPv6 中继功能;
- ☞ 用户比较容易遇到的现象是 DHCPv6 客户机连接在启动了 DHCPv6 服务器功能的交换机上, 却获得不到 IPv6 地址。遇到这种情况请首先确定 DHCPv6 客户机是否直连在启

动 DHCPv6 服务器的接口上,若是直连,则判断该接口 VLAN 的 IPv6 地址是否与本接口上启动的 DHCPv6 服务器地址池在同一个网段,如没有则请添加该网段至地址池;若不是直连,而且通过了其它启动了 DHCPv6 中继代理的三层交换机接入到 DHCPv6 服务器,则请判断 DHCPv6 客户机所在链路的交换机接口是否配置了 IPv6 地址,若没有配置 IPv6 地址,建议配置该接口 IPv6 地址。若该接口配置了地址,则进一步判断接口的 IPv6 地址是否在 DHCPv6 服务器地址池的网段之内,如没有则请添加该网段至地址池。

12.2 DHCPv6 option37, 38

12.2.1 DHCPv6 option37, 38 介绍

DHCPv6 (Dynamic Host Configuration Protocol for IPv6, 支持 IPv6 的动态主机配置协议)是针对 IPv6 编址方案设计的,为主机分配 IPv6 前缀、IPv6 地址和其他网络配置参数的协议。

当 DHCPv6 客户端希望从位于不同链路的 DHCPv6 服务器请求地址和配置参数时,客户端需要通过 DHCPv6 中继代理才能与服务器进行通信。中继代理节点收到来自客户端的 DHCPv6 消息,重新封装成 Relay-forward 报文转发给不在同一链路的服务器,服务器回应的 DHCPv6 消息被封装成 Relay-reply 报文交给 DHCPv6 中继代理节点,中继代理节点拆除中继封装还原成 DHCPv6 消息再转交给 DHCPv6 客户端,从而完成客户端与服务器的通信。

但 DHCPv6 中继代理使用上仍然存在一些问题,如:服务器如何给某些特定用户分配固定范围的 IP 地址?如何避免不法 DHCPv6 客户端伪造 DHCPv6 报文的客户端 MAC 地址字段发起 IP 地址耗尽攻击?如何避免不法 DHCPv6 客户端利用其它合法客户端 MAC 地址发起拒绝服务攻击?为此,IETF 提出了 rfc4649 和 rfc4580,即 DHCPv6 option 37 和 option 38 来解决这些问题。

DHCPv6 的 option 37 和 option 38 选项和 DHCP 中的 option 82 选项非常类似,当 DHCPv6 客户端经过 DHCPv6 中继代理发送请求报文到 DHCPv6 服务器时,若 DHCPv6 中继代理支持中继代理信息 option 37 和 option 38 选项,则由 DHCPv6 中继代理将 option 37 和 option 38 选项添加到请求报文中。对于服务器返回的应答报文,option 37 和 option 38 选项已经没有实际的意义,服务器的应答报文中剥离其中的 option 37 和 option 38 选项。因此 option 37 和 option 38 选项的应用对客户端是透明的。

DHCPv6 服务器借助 option 37 和 option 38 可以对 DHCPv6 客户端和 DHCPv6 中继设备的身份信息进行验证,并通过配置的分配策略灵活进行客户端的地址分配和管理,同时利用选项中包含的客户端信息可以有效预防上面的 DHCPv6 攻击,如伪造 DHCPv6 报文的客户端 MAC 地址字段发起的 IP 地址耗尽攻击。由于服务器这时可以识别多个请求报文来自同一个接入交换机端口,通过策略性限制分配地址个数可以防止地址耗尽。但 rfc4649 和

rfc4580 没有具体规定 DHCPv6 服务器如何使用 option 37 和 option 38 的信息，这可以由用户根据自己的需要灵活定制。

12.2.2 DHCPv6 option37， 38 配置任务序列

1. DHCPv6 snooping option 基本功能配置
2. DHCPv6 relay option 基本功能配置
3. DHCPv6 server option 基本功能配置

1.DHCPv6 snooping option 基本功能配置

命令	解释
全局配置模式	
ipv6 dhcp snooping remote-id option no ipv6 dhcp snooping remote-id option	设置本命令允许 DHCPv6 SNOOPING 支持 option 37 选项功能，本命令的 no 操作关闭 DHCPv6 SNOOPING 的 option 37 选项功能。
ipv6 dhcp snooping subscriber-id option no ipv6 dhcp snooping subscriber-id option	设置本命令允许 DHCPv6 SNOOPING 支持 option 38 选项功能，本命令的 no 操作关闭 DHCPv6 SNOOPING 的 option 38 选项功能。
ipv6 dhcp snooping remote-id policy {drop keep replace} no ipv6 dhcp snooping remote-id policy	本命令用于设置系统对于接收到的含有 option 37 选项的 DHCPv6 报文的重转发策略，其中 drop 模式表示如果报文中含有 option 37 选项，则系统丢弃该 DHCPv6 报文不作处理；keep 模式表示系统保持现有报文中的 option 37 选项不变转发给服务器处理；replace 模式表示系统使用自己的 option 37 选项替换现有报文中的 option 37 选项，然后转发给服务器处理。本命令的 no 操作设置 option 37 选项 DHCPv6 报文的重转发策略为 replace。
ipv6 dhcp snooping subscriber-id policy {drop keep replace} no ipv6 dhcp snooping subscriber-id policy	本命令用于设置系统对于接收到的含有 option 38 选项的 DHCPv6 报文的重转发策略，其中 drop 模式表示如果报文中含有 option 38 选项，则系统丢弃该 DHCPv6 报文不作处理；keep 模式表示系统保持现有报文中的 option 38 选项不变转发给服务器处理；replace 模式表示系统使用自己的 option 38 选项替换现有报文中的 option 38 选项，然后转发给服务器处理。本命令的 no 操作设置 option 38 选项 DHCPv6 报文

	的重转发策略为 replace。
ipv6 dhcp snooping subscriber-id select (sp sv pv spv) delimiter WORD (delimiter WORD) no ipv6 dhcp snooping subscriber-id select delimiter	配置用户配置选项来生成 subscriber-id, no 命令恢复为最初的默认配置即 VLAN 名加端口名的形式。
端口配置模式	
ipv6 dhcp snooping remote-id <remote-id> no ipv6 dhcp snooping remote-id	本命令用于设置在接收的 DHCPv6 请求报文中添加 option 37 选项的形式, <remote-id>为用户自定义的 option 37 中 remote-id 的内容, 它是一个长度小于 128 的字符串。本命令的 no 操作恢复 option 37 中 remote-id 的内容为默认的 enterprise-number 和 vlan MAC 地址。
ipv6 dhcp snooping subscriber-id <subscriber-id> no ipv6 dhcp snooping subscriber-id	本命令用于设置在接收的 DHCPv6 请求报文中添加 option 38 选项的形式, <subscriber-id>为用户自定义的 option 38 中 subscriber-id 的内容, 它是一个长度小于 128 的字符串。本命令的 no 操作恢复 option 38 中 subscriber-id 的内容为默认的 vlan 名加物理端口名形式, 如“Vlan2+Ethernet1/2”。

2. DHCPv6 relay option 基本功配置

命令	解释
全局配置模式	
ipv6 dhcp relay remote-id option no ipv6 dhcp relay remote-id option	设置本命令允许交换机中继支持 option 37 选项功能, 本命令的 no 操作关闭交换机中继的 option 37 选项功能。
ipv6 dhcp relay subscriber-id option no ipv6 dhcp relay subscriber-id option	设置本命令允许交换机中继支持 option 38 选项功能, 本命令的 no 操作关闭交换机中继的 option 38 选项功能。
ipv6 dhcp relay remote-id delimiter WORD no ipv6 dhcp relay remote-id delimiter	配置用户配置选项来生成 remote-id, no 命令恢复为最初的默认配置即 enterprise number 加 vlan MAC 形式。
ipv6 dhcp relay subscriber-id select	配置用户配置选项来生成

(sp sv pv spv) delimiter WORD (delimiter WORD) no ipv6 dhcp relay subscriber-id select delimiter	subscriber-id, no 命令恢复为最初的默认配置即 vlan 名加端口名的形式。
三层接口配置模式	
ipv6 dhcp relay remote-id <remote-id> no ipv6 dhcp relay remote-id	本命令用于设置从接口接收的 DHCPv6 请求报文添加 option 37 选项的形式, <remote-id> 为用户自定义的 option 37 的 remote-id 内容, 它是一个长度小于 128 的字符串。本命令的 no 操作恢复 option 37 的 remote-id 选项的形式为默认的 enterprise-number 和 vlan MAC 地址。
ipv6 dhcp relay subscriber-id <subscriber-id> no ipv6 dhcp relay subscriber-id	本命令用于设置从接口接收的 DHCPv6 请求报文添加 option 38 选项的形式, <subscriber-id> 为用户自定义的 option 38 的 subscriber-id 内容, 它是一个长度小于 128 的字符串。本命令的 no 操作恢复 option 38 的 subscriber-id 选项的形式为默认的 vlan 名加物理端口名形式, 如“Vlan2+Ethernet1/2”。

3. DHCPv6 server option 基本功能配置

命令	解释
全局配置模式	
ipv6 dhcp server remote-id option no ipv6 dhcp server remote-id option	本命令用于设置 DHCPv6 服务器支持对 option 37 选项的识别。本命令的 no 操作使系统不支持 option 37 选项。
ipv6 dhcp server subscriber-id option no ipv6 dhcp server subscriber-id option	本命令用于设置 DHCPv6 服务器支持对 option 38 选项的识别。本命令的 no 操作使系统不支持 option 38 选项。

ipv6 dhcp use class no ipv6 dhcp use class	本命令用于设置 DHCPv6 服务器在进行地址分配时支持使用 DHCPv6 class，本命令的 no 操作使服务器在进行地址分配时不支持使用 DHCPv6 class，但是却并不删除已经配置的有关 DHCPv6 class 的信息。
ipv6 dhcp class <class-name> no ipv6 dhcp class <class-name>	本命令用于定义一个 DHCPv6 class 并进入 DHCPv6 class 配置模式，no 命令用于删除这个 DHCPv6 class。
接口配置模式	
ipv6 dhcp server select relay-forw no ipv6 dhcp server select relay-forw	本命令用于设置 DHCPv6 服务器支持对报文中存在多个 option 37 选项或者 option38 选项时对其进行选择，选择最里层 relay-forw 中的 option 37 选项和 option38 选项。本命令的 no 操作恢复默认设置，即选择原始报文中的 option 37 选项和 option38 选项。
IPv6 DHCP Class 配置模式	
{remote-id [*]<remote-id>[*] subscriber-id [*]<subscriber-id>[*]} no {remote-id [*]<remote-id>[*] subscriber-id [*]<subscriber-id>[*]}	本命令在 ipv6 dhcp class 配置模式下配置匹配该 class 的 option37 选项和 option38 选项。
DHCPv6 地址池配置模式	
class <class-name> no class <class-name>	本命令在 DHCPv6 地址池配置模式下将 class 关联到地址池中，并进入地址池中 class 的配置模式，用 no 命令删除这种关联
address range <start-ip> <end-ip> no address range <start-ip> <end-ip>	本命令在 DHCPv6 地址池配置模式下用来为 DHCPv6 服务器地址池中的一个 DHCPv6 class 设置一个地址范围，no

	命令用来移除这个地址范围。 不支持 prefix/plen 形式。

12.2.3 DHCPv6 option37， 38 举例

12.2.3.1 DHCPv6 Snooping option37， 38 举例

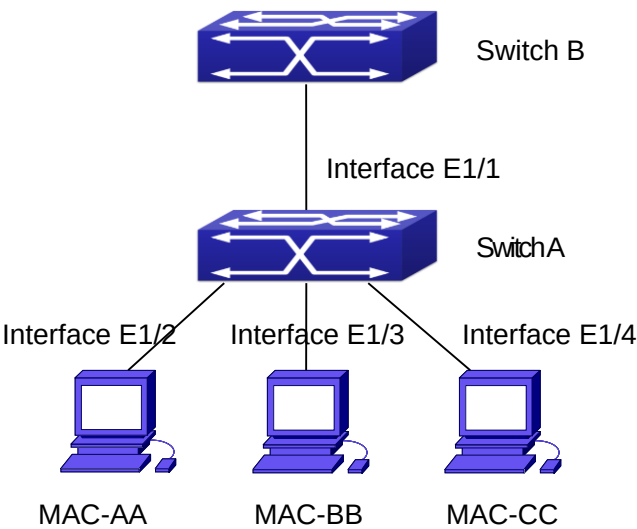


图 12-4 DHCPv6 Snooping option 示意图

如上图，Mac-AA、Mac-BB 和 Mac-CC 设备为正常用户，连接在交换机非信任端口 1/2、1/3 和 1/4 上，其通过 DHCPv6 Client 获得 IP 2010::2、2010::3 和 2010::4；DHCPv6 Server 连接在交换机的信任端口 1/1 上。配置 3 个地址分配策略(CLASS)，CLASS1 匹配 option 38，CLASS2 匹配 option 37，CLASS3 用于匹配 option 37 和 option 38。在地址池 EastDormPool 中，匹配 CLASS1 的请求将分配 2001:da8:100:1::2 到 2001:da8:100:1::30 的地址，匹配 CLASS2 的请求将分配 2001:da8:100:1::31 到 2001:da8:100:1::60 的地址，匹配 CLASS3 的请求将分配 2001:da8:100:1::61 到 2001:da8:100:1::100 的地址；在交换机 switch A 上开启 dhcpv6 snooping 功能同时配置 option 37、38 功能。

Switch A 配置：

```
SwitchA(config)#ipv6 dhcp snooping remote-id option
SwitchA(config)#ipv6 dhcp snooping subscriber-id option
SwitchA(config)#int e 1/1
SwitchA(config-if-ethernet1/1)#ipv6 dhcp snooping trust
SwitchA(config-if-ethernet1/1)#exit
SwitchA(config)#interface vlan 1
```



```
SwitchA(config-if-vlan1)#ipv6 address 2001:da8:100:1::1
SwitchA(config-if-vlan1)#exit
SwitchA(config)#interface ethernet 1/1-4
SwitchA(config-if-port-range)#switchport access vlan 1
SwitchA(config-if-port-range)#exit
SwitchA(config)#
```

Switch B 配置:

```
SwitchB(config)#service dhcpv6
SwitchB(config)#ipv6 dhcp server remote-id option
SwitchB(config)#ipv6 dhcp server subscriber-id option
SwitchB(config)#ipv6 dhcp pool EastDormPool
SwitchB(dhcpv6-eastdormpool-config)#network-address 2001:da8:100:1::2
2001:da8:100:1::1000
SwitchB(dhcpv6-eastdormpool-config)#dns-server 2001::1
SwitchB(dhcpv6-eastdormpool-config)#domain-name dhcpv6.com
SwitchB(dhcpv6-eastdormpool-config)# excluded-address 2001:da8:100:1::2
SwitchB(dhcpv6-eastdormpool-config)#exit
SwitchB(config)#
SwitchB(config)#ipv6 dhcp class CLASS1
SwitchB(dhcpv6-class-class1-config)#remote-id 00-03-0f-00-00-01 subscriber-id
vlan1+Ethernet1/1
SwitchB(dhcpv6-class-class1-config)#exit
SwitchB(config)#ipv6 dhcp class CLASS2
SwitchB(dhcpv6-class-class2-config)#remote-id 00-03-0f-00-00-01 subscriber-id
vlan1+Ethernet1/2
SwitchB(dhcpv6-class-class2-config)#exit
SwitchB(config)#ipv6 dhcp class CLASS3
SwitchB(dhcpv6-class-class3-config)#remote-id 00-03-0f-00-00-01 subscriber-id
vlan1+Ethernet1/3
SwitchB(dhcpv6-class-class3-config)#exit
SwitchB(config)#ipv6 dhcp pool EastDormPool
SwitchB(dhcpv6-eastdormpool-config)#class CLASS1
SwitchB(dhcpv6-pool-eastdormpool-class-class1-config)#address range
2001:da8:100:1::3 2001:da8:100:1::30
SwitchB(dhcpv6-pool-eastdormpool-class-class1-config)#exit
SwitchB(dhcpv6-eastdormpool-config)#class CLASS2
```

```
SwitchB(dhcpv6-pool-eastdormpool-class-class2-config)#address range
2001:da8:100:1::31 2001:da8:100:1::60
SwitchB(dhcpv6-eastdormpool-config)#class CLASS3
SwitchB(dhcpv6-pool-eastdormpool-class-class3-config)#address range
2001:da8:100:1::61 2001:da8:100:1::100
SwitchB(dhcpv6-pool-eastdormpool-class-class3-config)#exit
SwitchB(dhcpv6-eastdormpool-config)#exit
SwitchB(config)#interface vlan 1
SwitchB(config-if-vlan1)#ipv6 address 2001:da8:100:1::2/64
SwitchB(config-if-vlan1)#ipv6 dhcp server EastDormPool
SwitchB(config-if-vlan1)#exit
SwitchB(config)#
```

12.2.3.2 DHCPv6 Relay option37, 38 举例

案例 1:

在部署全 IPv6 校园网时，若使用专门的服务器进行 IPv6 地址的统一分配和管理，则可以使用路由交换设备的 DHCPv6 服务器功能进行 IPv6 地址分派。DHCPv6 服务器同时支持有状态与无状态 DHCPv6。

网络拓扑:

接入层使用二层接入设备 Switch1，连接宿舍楼栋的用户；一级汇聚层使用汇聚设备 Switch2 做 DHCPv6 中继代理；二级汇聚层使用汇聚设备 Switch3 做 DHCPv6 服务器，并与骨干网或更高的汇聚层设备相连；用户 PC 端一般装有 Windows Vista 操作系统，具有 DHCPv6 客户端。

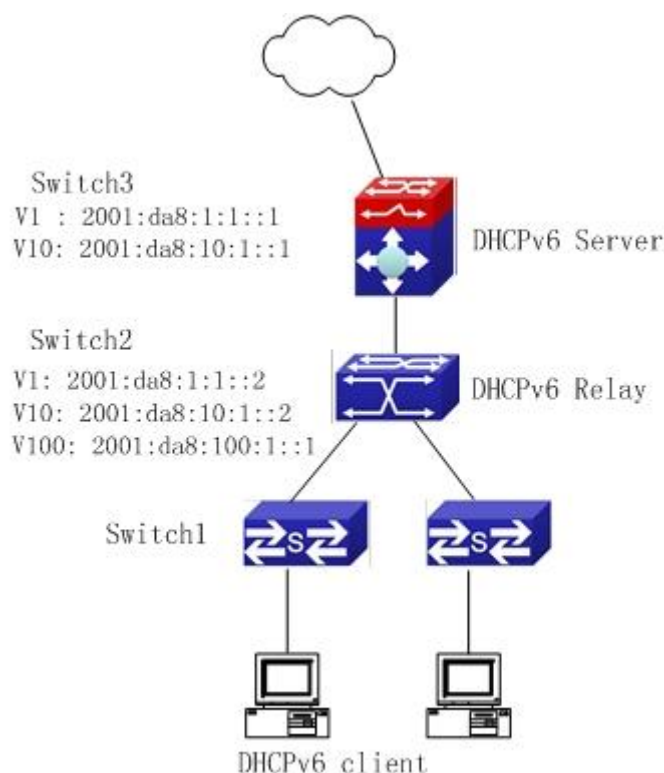


图 12-5 DHCPv6 relay option 示意图

Switch2 配置:

```
S2(config)#service dhcpv6
S2(config)#ipv6 dhcp relay remote-id option
S2(config)#ipv6 dhcp relay subscriber-id option
S2(config)#vlan 10
S2(config-vlan10)#int vlan 10
S2(config-if-vlan10)#ipv6 address 2001:da8:1::2/64
S2(config-if-vlan10)#ipv6 dhcp relay destination 2001:da8:10:1::1
S2(config-if-vlan10)#exit
S2(config)#
```

12.2.4 DHCPv6 option37, 38 排错帮助

- ☞ DHCPv6 客户端发送的请求报文是组播报文，只有本 VLAN 内的设备才能接收到，所以如果想让 DHCPv6 server 收到来自客户端的报文，必须让 DHCPv6 客户端与 DHCPv6 server 在同一个 VLAN 内，否则需要使用 DHCPv6 relay。
- ☞ Snooping option37,38 可以对已经含有 option37,38 的 dhcpv6 请求报文可以进行如下操作中的一个：用自己的 option37,38 替代报文中原来的 option37,38；丢弃该含有 option37,38 的报文；不添加也不丢弃，转发。所以，如果按照 option37,38 获取的地址不正确，或者获取不到地址，请检查第二台设备上 snooping option37,38 的策略配置。
- ☞ DHCPv6 server 默认从客户端发送的报文中获取 option37,38，如果没有，再从 relay

发送的报文中获取 option37,38。

- ☞ DHCPv6 server 只检查第一个 DHCPv6 relay 有没有添加 option37,38，不再检查后续的 relay 有没有添加 option37,38。即中继报文中只有最里层的 relay-forw 选项中的 option37,38 有效。

12.3 ND 绑定

12.3.1 概述

ND 是 IPV6 协议中的邻居发现协议，其工作原理同 ARP 类似，因此我们采取同 ARP 绑定相同的方法处理利用 ND 进行绑定和攻击的手段。

12.3.2 ND 绑定配置

配置 ND 绑定配置序列如下：

4. 关闭 ND 自动更新功能
5. 关闭 ND 自动学习与更新功能
6. 将动态的 ND 转化为静态 ND 的功能

1. 关闭 ND 自动更新功能

命令	解释
全局配置模式和接口配置模式	
ipv6 nd-security updateprotect no ipv6 nd-security updateprotect	关闭和启动 ND 的自动更新功能。

2. 关闭 ND 自动学习与更新功能

命令	解释
全局配置模式和接口配置模式	
ipv6 nd-security learnprotect no ipv6 nd-security learnprotect	关闭和启动 ND 的自动学习与更新功能。

3. 将动态的 ND 转化为静态 ND 的功能

命令	解释
全局配置模式和接口配置模式	
ipv6 nd-security convert	将动态 ND 转化静态。

12.3.3 ND 绑定举例

请参考 ARP 绑定举例。

12.4 RIPng

12.4.1 RIPng 介绍

RIP协议最早在ARPANET网络中上使用，专门用于小型简单网络中。RIPng协议是基于Bellman-Ford算法的距离向量路由协议，是用于IPv6的RIP协议。运行距离向量协议的网络设备定期向相邻设备发送两种信息：

- 到达目的网络所经过的跳数，即使用的度（metric），或者通过网络的数量。
- 下一跳是什么，或者达到目的网络要使用的方向（向量）。

距离向量三层交换机定期向相邻的三层交换机发送它们的整个路由选择表。三层交换机在从相邻三层交换机接收到的信息的基础之上建立自己的路由选择信息表。然后，将信息传递到它的相邻三层交换机。结果是路由选择表是在第二手信息的基础上建立的，距离代价超过15跳的路由将被视为不可达。

RIPng协议是可选路由协议，它是基于UDP的协议，使用RIPng的每个主机在UDP的521端口上发送和接收数据报。所有运行RIPng协议的三层交换机，每隔30秒向所有邻居三层交换机发送路由表更新信息。如果180秒内没有收到来自对端的信息，那么认为该设备崩溃或相连的网络不可达。但到该三层交换机的路由还将在路由表中保持120秒，然后才被删除。

由于运行RIPng协议的三层交换机使用第二手信息建立路由表，因此至少会遇到一个问题——无穷记数问题。对于一个运行RIPng路由协议的网络，当某条RIPng路由变为不可达，RIPng三层交换机通常不会立刻发送路由更新报文，而是等待到达周期更新时间间隔（每30秒）时才发送有该路由信息的更新数据报。如果在没有收到更新报文之前，邻居向该三层交换机发送了一个包含邻居三层交换机自身路由表信息的数据报，那么会引起“无穷记数”现象，即出现到不可达三层交换机的路由选择度量固定递增的现象。这大大影响了路由选择、路由汇聚时间。

为了避免“无穷记数”现象，RIPng协议提供了“水平分割”和“触发更新”等机制来解决路由循环问题。“水平分割”的原理是避免向网关发送从该网关学习到的路由，它包括“简单水平分割”——删除从邻居网关学习到的、将发送给该邻居网关的路由，和“逆向毒性水平分割”——不但在更新包中删除上述路由，而且将这些路由的代价设置为无穷。“触发更新”机制定义无论网关何时改变路由度量，都立即将更新数据报并以广播形式发送出去，而不等待达到30秒更新时间间隔。

RIPng协议目前只有版本1一个版本：RFC2080中介绍了RIPng协议。RIPng采用发送组播数据报的方式发送路由更新数据报（组播地址为FF02::9）。

每个运行RIPng协议的三层交换机都有一个路由数据库，数据库中包含了该三层交换机所有可达目的地的路由项，并以此建立路由表。当RIPng三层交换机向其相邻设备发送

路由更新数据报时，路由更新数据报中包含了该三层交换机依据路由数据库建立的整个路由表。因此，对于较大的网络系统，每台三层交换机需要传输和处理的路由数据量很大、负担重，从而大大影响网络性能。

同时，RIPng 协议支持将其它 IPv6 路由协议发现的路由信息引入到路由表中。

RIPng 协议运行过程描述如下：

1. 启动 RIPng，以组播形式向其相邻三层交换机发送请求数据报，相邻设备收到请求数据报后，响应该请求，并回送包含本地路由信息的响应数据报。
2. 三层交换机接收到响应数据报后，修改本地路由表，同时向相邻设备发送触发更新数据报，广播路由更新信息。相邻三层交换机接收到触发更新数据报之后，向其相邻的三层交换机发送触发更新数据报。经过一连串触发更新报的广播之后，各三层交换机都得到并保持最新的路由信息。

同时，RIPng 三层交换机每隔 30 秒向其相邻设备广播本地路由表。相邻设备在接收到数据报之后，对本地路由进行维护，选择出最佳路由并向其各自的设备广播更新信息，使更新的路由最终达到全局有效。另外，RIPng 采用超时机制对过时的路由进行超时处理，即三层交换机在一定时间间隔内（invalid 定时器间隔）没有收到来自某邻居的周期更新数据报，则认为来自该三层交换机的路由为无效路由，然后该路由再在路由表中保留一定时间间隔（garbage collect 定时器间隔），最后删除该路由。

由于 IPv6 网络还在发展中，有时网络中存在不支持 IPv6 的网络环境，这就需要通过隧道进行 IPv6 的操作，所以我们的 RIPng 支持在配置隧道上的配置和运行，以 IPV4 单播的方式穿过不支持 IPv6 的网络。

12.4.2 RIPng 配置

RIPng 配置任务序列：

1. 启动 RIPng 协议（必须）
 - (1) 启动/关闭 RIPng 协议
 - (2) 配置运行 RIPng 协议的网段
2. 配置 RIPng 协议参数（可选）
 - (1) 配置 RIPng 发包机制
 - 1) 配置 RIPng 数据报的定点发送
 - (2) 配置 RIPng 路由参数
 - 1) 配置引入路由（缺省路由权值、配置 RIPng 中引入其它协议的路由）
 - 2) 配置路由偏移
 - 3) 配置应用路由过滤
 - 4) 配置水平分割
3. 配置 RIPng 协议其它参数
 - (1) 配置 RIPng 更新、超时、抑制等计时器时间
4. 删除 RIPng 路由表中的特定路由

5. 配置 RIPng 路由聚合

- (1) 配置 IPv6 路由器模式的聚合路由
- (2) 配置 IPv6 接口配置模式的聚合路由
- (3) 显示 IPv6 聚合路由信息

6. 配置 RIPng 引入不同进程 OSPFv3 路由

- (1) 启动 RIPng 引入不同进程 OSPFv3 路由功能
- (2) 显示和调试 RIPng 引入不同进程 OSPFv3 路由功能相关信息

1. 启动 RIPng 协议

在 DCNOS 中运行 RIPng 路由协议的基本配置很简单，通常只需打开 RIPng 开关、并且配置运行 RIPng 的接口，即按 RIPng 缺省配置发送和接收 RIPng 数据报。

命令	解释
全局配置模式	
[no] router IPv6 rip	打开 RIPng 协议; 本命令的 no 操作关闭 RIPng 协议。
接口配置模式	
[no] IPv6 router rip	设定运行 RIPng 协议的接口; 本命令的 no 操作为设定接口不运行 RIPng 协议。

2. 配置 RIPng 协议参数

(1) 配置 RIPng 发包机制

1) 配置 RIPng 数据报的定点发送

命令	解释
路由器配置模式	
[no] neighbor <IPv6-address> <ifname>	指定需要定点发送的邻居路由器的 IPv6 Link-local 地址和接口名; 本命令的 no 操作取消指定的路由器。
[no] passive-interface <ifname>	指示 RIPng 三层交换机在指定的接口上阻塞 RIPng 组播，只能向配置了 neighbor 的三层交换机之间发送 RIPng 数据包; 本命令的 no 操作取消该功能。

(2) 配置 RIPng 路由参数

1) 配置引入路由（缺省路由权值、配置 RIPng 中引入其它协议的路由）

命令	解释
路由器配置模式	

default-metric <value> no default-metric	设定引入路由的缺省路由权值；本命令的 no 操作恢复缺省设置 1。
[no]redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>]	在 RIPng 数据报中引入从其它路由协议引入的路由；本命令的 no 操作取消引入的相应协议的路由。
[no]default-information originate	产生一条默认路由到 RIPng 协议中；no 操作取消该特性。

2) 配置路由偏移

命令	解释
路由器配置模式	
[no] offset-list <access-list-number access-list-name> {in out} <number > [<ifname>]	设定接口发送和接收 RIPng 数据报时给路由的度量一个偏移量；本命令的 no 操作移去偏移列表。

3) 配置应用路由过滤和路由聚合

命令	解释
路由器配置模式	
[no] distribute-list {< access-list-number access-list-name > prefix<prefix-list-name>} {in out} [<ifname>]	设定接口发送和接收 RIPng 数据报时过滤路由；本命令的 no 操作不配置路由过滤。
[no]aggregate-address <IPv6-address>	配置路由聚合，no 命令取消路由聚合。

4) 配置水平分割

命令	解释
接口配置模式	
IPv6 rip split-horizon [poisoned]	配置接口发送数据报时进行水平分割操作，POISONED 表示带逆向毒化。
no IPv6 rip split-horizon	取消水平分割操作。

3. 配置 RIPng 协议其它参数

(1) 配置 RIPng 更新、超时、抑制等计时器时间

命令	解释
路由器配置模式	
timers basic <update> <invalid> <garbage> no timers basic	调整 RIPng 计时器更新、期满、垃圾收集的时间；本命令的 no 操作回复缺省设置。

4. 删除 RIPng 路由表中的特定路由

命令	解释
特权配置模式	
clear IPv6 rip route {<IPv6-address> kernel static connected rip ospf isis bgp all}	本命令删除 RIPng 路由表中的特定路由。

5. 配置 RIPng 路由聚合

(1) 配置 IPv6 全局聚合路由

命令	解释
路由器配置模式	
ipv6 rip aggregate-address X:X::X:X/M no ipv6 rip aggregate-address X:X::X:X/M	配置或取消 IPv6 全局聚合路由。

(2) 配置 IPv6 接口聚合路由

命令	解释
接口配置模式	
ipv6 rip aggregate-address X:X::X:X/M no ipv6 rip aggregate-address X:X::X:X/M	配置或取消 IPv6 接口聚合路由。

(3) 显示 IPv6 聚合路由信息

命令	解释
特权模式和配置模式	
show ipv6 rip aggregate	显示 IPv6 聚合路由信息，如聚合的接口，度量，聚合的路由数，被聚合的次数。

6. 配置 RIPng 引入不同进程 OSPFv3 路由

(1) 启动 RIPng 引入不同进程 OSPFv3 路由功能

命令	解释
Router ipv6 rip 配置模式	
redistribute ospf [<process-tag>] [metric<value>] [route-map <word>] no redistribute ospf [<process-tag>]	启动或者关闭 RIPng 引入其他 OSPFv3 进程路由功能。

(2) 显示和调试 RIPng 引入不同进程 OSPFv3 路由功能相关信息

命令	解释
特权配置模式	
show ipv6 rip redistribute	显示 RIPng 进程引入其它外部路由的配置信息。
特权模式	
debug ipv6 rip redistribute message send no debug ipv6 rip redistribute message send debug ipv6 rip redistribute route receive no debug ipv6 rip redistribute route receive	打开或关闭 RIPng 引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 RIPng 进程收到 nsm 发来的路由信息的调试开关。

12.4.3 RIPng 典型案例

12.4.3.1 RIPng 典型案例

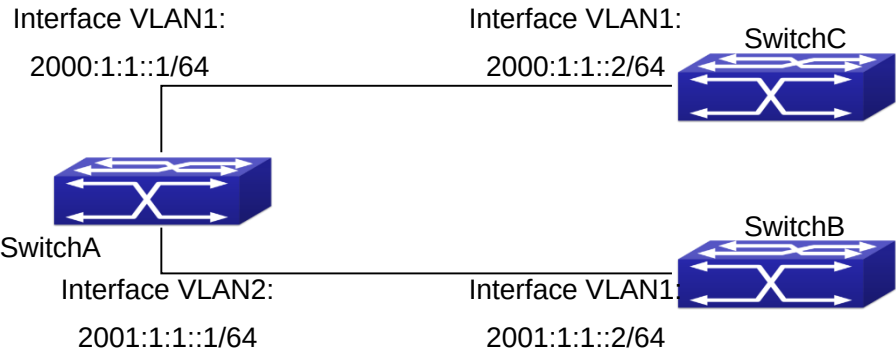


图 12-6 RIPng 案例

图 3-1 为三台三层交换机组成的一个网络，三层交换机 SwitchA 三层交换机 SwitchB 和三

层交换机 SwitchC 分别通过以太网口 VLAN2 和 VLAN1 相连,三台三层交换机都运行 RIPng 路由协议。设三层交换机 SwitchA (VLAN1: 2001:1:1::1/64, VLAN2: 2001:1:1::1/64) 只与三层交换机 SwitchB (VLAN1: 2001:1:1::2/64) 交流三层交换机更新信息,而不向三层交换机 SwitchC (VLAN1: 2001:1:1::2/64) 交流三层交换机更新信息。

三层交换机 SwitchA、SwitchB、SwitchC 的配置分别如下:

a) 三层交换机 SwitchA:

启动 RIPng 协议

```
SwitchA(config)#router IPv6 rip
```

```
SwitchA(config-router)#exit
```

配置以太网口 VLAN1 的 IPv6 地址和接口运行 RIPng

```
SwitchA#config
```

```
SwitchA(config)# interface Vlan1
```

```
SwitchA(config-if-Vlan1)# IPv6 address 2000:1:1::1/64
```

```
SwitchA(config-if-Vlan1)#IPv6 router rip
```

```
SwitchA(config-if-Vlan1)#exit
```

配置以太网口 VLAN2 的 IP 地址和接口运行 RIPng

```
SwitchA(config)# interface Vlan2
```

```
SwitchA(config-if-Vlan2)# IPv6 address 2001:1:1::1/64
```

```
SwitchA(config-if-Vlan2)#IPv6 router rip
```

```
SwitchA(config-if-Vlan2)#exit
```

配置接口 VLAN1 不向交换机 SwitchC 发送 RIPng 报文

```
SwitchA(config)#
```

```
SwitchA(config-router)#passive-interface Vlan1
```

```
SwitchA(config-router)#exit
```

b) 三层交换机 SwitchB:

启动 RIP 协议

```
SwitchB(config)#router IPv6 rip
```

```
SwitchB(config-router-rip)#exit
```

配置以太网口 VLAN1 的 IP 地址和接口运行 RIPng。

```
SwitchB#config
```

```
SwitchB(config)# interface Vlan1
```

```
SwitchB(config-if)# IPv6 address 2001:1:1::2/64
```

```
SwitchB(config-if)#IPv6 router rip
```

```
SwitchB(config-if)#exit
```

c) 三层交换机 SwitchC:

启动 RIP 协议

```
SwitchC(config)#router IPv6 rip
SwitchC(config-router-rip)#exit
配置以太网口 VLAN1 的 IP 地址和接口运行 RIPng。
SwitchC#config
SwitchC(config)# interface Vlan1
SwitchC(config-if)# IPv6 address 2000:1:1::2/64
SwitchC(config-if)#IPv6 router rip
SwitchC(config-if)#exit
```

12.4.3.2 RIPng 路由聚合功能典型案例

如下图的应用环境：

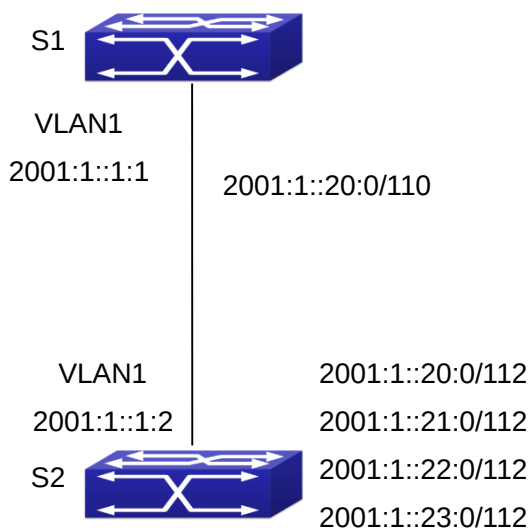


图 12-7 RIPng 路由聚合典型应用环境

在上述网络拓扑图中，S2通过接口vlan1与S1相连，S2另外有四条子网路由，分别是：2001:1::20:0/112，2001:1::21:0/112，2001:1::22:0/112，2001:1::23:0/112。S2支持路由聚合，在S2的接口vlan1上配置聚合路由2001:1::20:0/110后，通过vlan1给S1发送路由信息时，将四条子网路由聚合成一条聚合路由2001:1::20:0/110，发送给S1，而不将子网发送给邻居。减小了S1的路由表规模，节省了内存。

S1配置序列：

```
S1(config)#router ipv6 rip
S1(config)# interface Vlan1
S1(config-if-Vlan1)#IPv6 address 2001:1::1:1/112
S1(config-if-Vlan1)#IPv6 router rip
```

S2配置序列：

```
S2(config)#router ipv6 rip
```

```
S2(config)#interface vlan 1
S2(config-if-Vlan1)#IPv6 address 2001:1::1:2/112
S2(config-if-Vlan1)#IPv6 router rip
S2(Config-if-Vlan1)#ipv6 rip agg 2001:1::20:0/110
```

12.4.4 RIPng 排错帮助

在配置、使用 RIPng 协议时，可能会由于物理连接、配置错误等原因导致 RIPng 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误和 IP Forwarding 命令开启；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后，先启动 RIPng 协议（使用 router IPv6 rip 命令）并且配置接口（使用 IPv6 router rip 命令）再在相应接口配置 RIPng 协议参数；
- ☞ 接着，注意 RIPng 协议的自身特点——RIPng 三层交换机每隔 30 秒向它所有邻居三层交换机发送路由表更新信息，如果 180 秒内没有收到来自某三层交换机的信息，那么认为该三层交换机崩溃或相连的网络不可达，到该三层交换机的路由还将在路由表中保持 120 秒然后被删除。因此，如果删除某个 RIPng 路由，需等待 300 秒后才能保证 RIPng 数据库中除去此路由项。

如果仍无法解决 RIP 路由问题，那么请使用 debug IPv6 rip 调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.5 OSPFv3

12.5.1 OSPFv3 介绍

OSPFv3（Open Shortest Path First）协议即“开放最短路径优先协议”第三版，是 OSPF 协议的 IPv6 版本。它是一种基于链路状态的自治系统内部的动态路由协议，它通过路由器或三层交换机之间交换链路状态信息来组成一个链路状态数据库，然后基于这个数据库用最短路径优先算法生成路由表。

自治系统（AS）是一个自我管理的互连网络。在大型网络中，例如 Internet，极大的互连网络被分解为自治系统。连接到 Internet 上的大型公司网络是独立的自治系统，因为 Internet 上的其他主机并不由它来管理，而且它和 Internet 三层交换机并不共享内部路由选择信息。

每个链路状态三层交换机可提供与其邻居三层交换机的拓扑结构的信息：

- 三层交换机所连接的网段（链路）
- 连接链路的状态

链路状态信息在网络上泛洪（flooding），目的是使所有的三层交换机可以接收到第一手信息。链路状态三层交换机并不会广播包含在它们的路由表内的所有信息，相反链路状态三层交换机将发送关于已经改动的链路状态信息。链路状态三层交换机将向它们的邻居发送呼叫信息建立邻接关系，然后在邻接三层交换机之间发送链路状态通告（LSA）。邻接三层交换机将LSA复制到它们的路由选择表中，并传递这些信息到网络的剩余部分。这个过程称为泛洪（flooding）。这样，实现了向网络发送第一手信息，为网络建立更新路由的准确映射。链路状态路由选择协议使用称为代价的方法（而不是使用跳）来选择路径。代价是自动或人工赋值的。根据链路状态协议的算法，代价可以计算数据包必须穿越的跳数目、链路带宽、链路上的当前负载，甚至可以由管理员加入的权重来评价。

- 1) 当一个链路状态三层交换机进入链路状态互连网络时，它发送一个呼叫数据包（HELLO）以了解其邻居，并建立邻接关系。
- 2) 邻居用关于它们所连接的链路以及相关的代价度的信息进行应答。
- 3) 起始的三层交换机用这个信息来建立它的路由选择表。
- 4) 然后，作为定期更新的一部分。三层交换机向它的邻接三层交换机发送链路状态数据包。这个LSA包括了那个三层交换机的链路及相关代价。
- 5) 每个邻接三层交换机复制该LSA数据包，并且将LSA传递到下一个邻居（即泛洪）。
- 6) 因为三层交换机并没有在向前泛洪LSA之前重新计算路由选择数据库，因此收敛时间大大减少了。

链路状态路由选择协议的一个主要优点就是路由无穷记数不可能形成，原因是链路状态协议的路由器独立建立它们自己的路由选择信息表。另一个优点是，在链路状态互连网络中收敛非常快，原因是一旦路由选择拓扑出现变动，则更新在互连网络上迅速泛洪。这些优点又释放了三层交换机的资源，因为对不好的路由信息所花费的处理能力和带宽消耗都很少。

OSPFv3协议的特点：OSPFv3协议支持各种规模的网络，最多可以支持几百台三层交换机；路由拓扑结构变化后OSPFv3能够立即发送链路状态更新数据包，收敛迅速；OSPFv3根据收集到的链路状态采用最短路径算法计算路由，保证了无自环路由；OSPFv3将自治系统划分为多个域，减小了数据库尺寸、减少了占用网络带宽、降低了计算负担（根据三层交换机在自治系统中所处的位置可以划分为域内三层交换机、域边界三层交换机、自治系统边界三层交换机和骨干三层交换机）；OSPFv3支持负载分担，支持到同一目的地址的多条等价路由；OSPFv3支持4级路由机制（按域内路由、域间路由、第一类外部路由和第二类外部路由顺序分级处理路由）；OSPFv3协议支持IP子网、支持其他路由协议的路由的引入；OSPFv3支持组播发送数据包。

每个OSPFv3三层交换机都维护着一份描述整个自治系统拓扑结构的数据库。每一台三层交换机都搜集本地状态的信息，如可用接口信息，可达邻居信息，然后通过使用链路状态通告（即向外发送链路状态信息），本三层交换机与其他OSPFv3三层交换机交换链路状态信息，从而形成描述整个自治系统的链路状态数据库。根据链路状态数据库，各三层交换机构建一棵以自己为根的最短路径树，这棵树给出了到自治系统中各节点的路由。如果存在两台或两台以上的三层交换机（即多路访问网络），该网络上要选出“指定三层交换机”和“备份指定三层交换机”，指定三层交换机负责将网络的链路状态播出去，引入这一概念，有助于

减少在多路访问网络上各三层交换机之间的数据流量。

OSPFv3协议要求将自治系统的网络划分成区域来管理，即将自治系统划分为0域（骨干域）和非0域，区域间传送的路由信息被进一步抽象概括，从而减少了整个网络的实用带宽。OSPFv3使用4类不同的路由，按优先顺序来说分别是区域内路由、区域间路由、第一类外部路由和第二类外部路由。区域内和区域间路由描述的是自治系统内部的网络结构，而外部路由则描述了应该如何选择到自治系统以外的目的地。第一类外部路由对应于OSPFv3从其它内部路由协议所引入的信息，这些路由的开销和OSPFv3自身路由的开销具有可比性；第二类外部路由对应于OSPFv3从外部路由协议所引入的信息，它们的开销远大于OSPFv3自身的路由开销，所以在计算路由开销时，不考虑OSPFv3自身的路由开销。

OSPFv3的区域以Backbone（骨干区域）作为核心，该区域标识为0域，所有的其他区域都必须与0域在逻辑上相连，0域必须保证连续。为此，骨干区域上特别引入了虚连接的概念以保证即使在物理上分割的该区域仍然在逻辑上具有连通性。在同一区域内的所有三层交换机的配置要保持一致。

如上所述，LSA只能在邻接三层交换机之间进行传递，OSPFv3协议包括七类LSA：链路LSA、域内前缀LSA、路由器LSA、网络LSA、域间前缀LSA、域间路由器LSA和自治系统外部LSA。路由器LSA由OSPFv3域内的每个三层交换机产生，并发送给域内所有其它邻接三层交换机；网络LSA是由多路访问网络所在OSPFv3域的指定三层交换机产生的（为减少多路访问网络上各三层交换机之间的数据流量，多路访问网络中需要选出“指定三层交换机”和“备份指定三层交换机”，由指定三层交换机负责将网络的链路状态播出去），并发送给域内所有其它邻接三层交换机；域间前缀LSA和域间路由器LSA由OSPFv3域边界三层交换机产生，并在域边界三层交换机之间传送；自治系统外部LSA由自治系统外部边界三层交换机产生，并在整个自治系统内传递。链路LSA由链路上的三层交换机产生，并且发送给链路上的其它三层交换机。域内前缀LSA由域内每个链路上的指定路由器产生，并且泛洪到整个域内。

对于以外部链路状态通告为主的自治系统，为减少拓扑数据库的尺寸，OSPFv3允许将某些域配置成STUB域。路由器LSA、网络LSA、域间前缀LSA、链路LSA、域内前缀LSA允许被通告到STUB域。STUB域内必须使用缺省路由，STUB域的域边界三层交换机通过域间前缀LSA来通告到STUB域的缺省路由；这些缺省路由只在STUB内泛滥，不泛滥到STUB域之外。每个STUB域对应一条缺省路由，STUB域到AS外部目的地的路由只依赖该域的缺省路由。

OSPFv3协议路由的简单计算过程如下：

- 1) 每个支持 OSPFv3 协议的三层交换机都维护着一个描述整个自治系统拓扑结构的链路状态数据库（即 LS 数据库）。每个三层交换机根据自己周围的网络拓扑结构生成一条链路状态通告（即路由器 LSA），通过相互之间发送链路状态更新包（LSU）将各自的 LSA 发送给网络中的其它三层交换机。这样，每台三层交换机都收到了其它三层交换机的 LSA，所有 LSA 在一起便形成了链路状态数据库。

- 2) 由于一条 LSA 是对该三层交换机周围网络拓扑结构的描述, 那么 LS 数据库则是对整个网络的拓扑结构的描述。三层交换机很容易根据 LS 数据库建立一张带权有向图, 显然自治系统内的所有三层交换机都将得到一张相同的网络拓扑图。
- 3) 每台三层交换机都使用最短路径优先 SPF 算法计算出一棵以自己为根的最短路径树, 该树给出了到自治系统中各个节点的路由, 外部路由信息为叶子节点, 外部路由可根据广播它的三层交换机进行标记以记录关于自治系统的额外信息。由此可见, 各个三层交换机得到的路由表是不同的。

OSPFv3 协议是 IETF 组织开发的, 目前使用的是 OSPFv3 版本是参照 RFC2328 和 RFC2740 中描述的内容实现的。

由于 IPv6 网络还在发展中, 有时网络中存在不支持 IPv6 的网络环境, 这就需要通过隧道进行 IPv6 的操作, 所以我们的 OSPFv3 支持在配置隧道上的配置和运行, 以 IPv4 单播的方式穿过不支持 IPv6 的网络。

12.5.2 OSPFv3 配置

OSPFv3 配置任务序列:

1. 启动 OSPFv3 协议 (必须)
 - (1) 启动/关闭 OSPFv3 协议 (必须)
 - (2) 配置运行 OSPFv3 的三层交换机的 router-id
 - (3) 配置运行 OSPFv3 的网络范围 (可选)
 - (4) 在接口使能 OSPFv3 (必须)
2. 配置 OSPFv3 辅助参数 (可选)
 - (1) 配置 OSPFv3 发包机制参数
 - 1) 配置 OSPFv3 接口为只收不发
 - 2) 配置接口发送数据包的代价
 - 3) 配置 OSPFv3 发包定时器参数 (广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器)
 - (2) 配置 OSPFv3 引入路由参数
 - 1) 配置引入外部路由的缺省参数 (缺省类型、缺省标记值、缺省代价值)
 - 2) 配置在 OSPFv3 中引入其它协议的路由
 - (3) 配置 OSPFv3 引入其他 OSPFv3 进程路由功能
 - 1) 启动 OSPFv3 引入其他 OSPFv3 进程路由功能
 - 2) 显示相关配置信息
 - 3) 调试
 - (4) 配置 OSPFv3 协议其它参数
 - 1) 配置 OSPFv3 路由协议优先级

- 2) 配置 OSPFv3 STUB 域及缺省路由的代价
 - 3) 配置 OSPFv3 虚链路
 - 4) 配置接口在选举指定三层交换机 DR 中的优先级
3. 关闭 OSPFv3 协议

1. 启动 OSPFv3 协议

在三层交换机上运行 OSPFv3 路由协议的基本配置也很简单，通常只需打开 OSPFv3 开关、在接口使能 OSPFv3 即可，OSPFv3 协议的参数都使用缺省值。如需修改 OSPFv3 协议参数值，参看 2.配置 OSPFv3 辅助参数。

命令	解释
全局配置模式	
[no] router IPv6 ospf <tag>	本命令初始化 ospfv3 路由进程并且进入 ospfv3 模式配置 ospfv3 路由进程。no 操作终止相应的进程。（必须）
OSPFv3 协议配置模式	
router-id <router_id> no router-id	为 ospfv3 进程配置路由器 ID。no 操作恢复 ID 到 0.0.0.0（必需）。
[no] passive-interface<ifname>	将一个接口设置为只收不发；本命令的 no 操作取消配置。
接口配置模式	
[no] IPv6 router ospf {area <area-id> [instance-id <instance-id> tag <tag> [instance-id <instance-id>]] tag <tag> area <area-id> [instance-id <instance-id>]}	在接口上使能 ospfv3 路由；本命令的 no 操作取消此配置。

2. 配置 OSPFv3 辅助参数

(1) 配置 OSPFv3 发包机制参数

- 1) 配置 OSPF 接口为只收不发
- 2) 配置接口发送数据包的代价

命令	解释
接口配置模式	
IPv6 ospf cost <cost> [instance-id <id>] no IPv6 ospf cost [instance-id <id>]	指定接口运行 OSPFv3 协议所需的代价；本命令的 no 操作恢复代价的缺省值。

3) 配置 OSPFv3 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）

命令	解释
接口配置模式	
IPv6 ospf hello-interval <time> [instance-id <id>] no IPv6 ospf hello-interval [instance-id <id>]	配置接口周期发送 HELLO 数据包的时间间隔；本命令的 no 操作恢复为缺省值。
IPv6 ospf dead-interval <time> [instance-id <id>] no IPv6 ospf dead-interval [instance-id <id>]	配置认定相邻三层交换机失效的时间间隔；本命令的 no 操作恢复为缺省值。
IPv6 ospf transit-delay <time> [instance-id <id>] no IPv6 ospf transit-delay [instance-id <id>]	设置在接口上传送链路状态广播的时延值；本命令的 no 操作恢复为缺省值。
IPv6 ospf retransmit <time> [instance-id <id>] no IPv6 ospf retransmit [instance-id <id>]	配置接口与邻接三层交换机之间传送链路状态通告时的重传间隔；本命令的 no 操作恢复为缺省值。

(2) 配置 OSPFv3 引入路由参数

配置在 OSPFv3 中引入其它协议的路由

命令	解释
OSPF 协议配置模式	
[no]redistribute {kernel connected static rip isis bgp} [metric<value>] [metric-type {1 2}][route-map<word>]	引入其他协议发现路由和静态路由作为外部路由信息；本命令的 no 操作取消引入的外部路由信息。

(3) 配置 OSPFv3 引入其他 OSPFv3 进程路由功能

1) 启动 OSPFv3 引入其他 OSPFv3 进程路由功能

命令	解释
Router ipv6 ospf 配置模式	

redistribute ospf [<process-id> [metric<value>] [metric-type {1 2}][route-map<word>] no redistribute ospf [<process-id> [metric<value>] [metric-type {1 2}] [route-map<word>]	启动或者关闭 OSPFv3 引入其他 OSPFv3 进程路由功能。

2) 显示相关配置信息

命令	解释
特权模式或者配置模式	
show ipv6 ospf [<process-id> redistribute	显示 OSPFv3 进程引入其它外部路由的配置信息。

3) 调试

命令	解释
特权模式	
debug ipv6 ospf redistribute message send no debug ipv6 ospf redistribute message send debug ipv6 ospf redistribute route receive no debug ipv6 ospf redistribute route receive	打开或关闭 OSPFv3 进程引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 OSPFv3 进程收到 nsm 发来的路由信息的调试开关。

(4) 配置 OSPFv3 协议其它参数

1) 配置 OSPF v3 STUB 域及缺省路由的代价

2) 配置 OSPFv3 虚链路

命令	解释
OSPFV3 协议配置模式	
timers spf <spf-delay> <spf-holdtime> no timers spf	配置 OSPFv3 的 SPF 定时器；本命令的 no 操作恢复为缺省值。

area <id> stub [no-summary] no area <id> stub [no-summary] area <id> default-cost <cost> no area <id> default-cost area <id> virtual-link A.B.C.D [instance-id <instance-id> INTERVAL] no area <id> virtual-link A.B.C.D [INTERVAL]	配置 OSPFv3 区域下的参数(STUB 域, 和虚链路); 本命令的 no 操作恢复为缺省值。

4) 配置接口在选举指定三层交换机 DR 中的优先级

命令	解释
接口配置模式	
IPv6 ospf priority <priority> [instance-id <id>] no IPv6 ospf priority [instance-id <id>]	配置接口在选举“指定三层交换机”时的优先级; 本命令的 no 操作为恢复优先级缺省值。

3. 关闭 OSPFv3 协议

命令	解释
全局配置模式	
no router IPv6 ospf [<tag>]	关闭 OSPFv3 路由协议。

12.5.3 OSPFv3 案例

案例一：OSPFv3自治系统

以五台三层交换机为例组成一个OSPFv3自治系统，OSPF区域划分见下图：

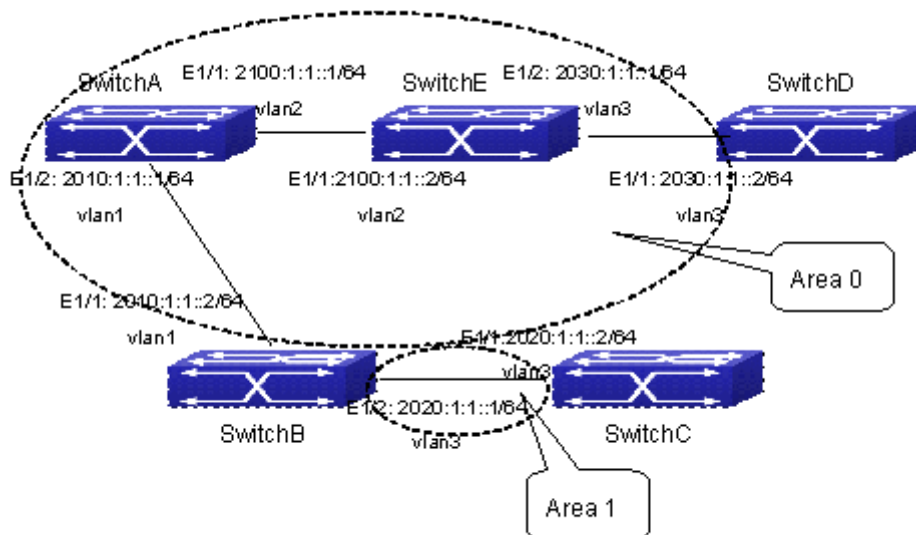


图 12-8 OSPFv3 自治系统网络拓扑示意图

三层交换机Switch1- Switch5的配置分别如下：

三层交换机Switch1:

启动OSPFv3协议，配置路由器ID

```
Switch1(config)#router IPv6 ospf
```

```
Switch1 (config-router)#router-id 192.168.2.1
```

配置接口vlan1的IPv6地址和所属的OSPFV3域

```
Switch1#config
```

```
Switch1(config)# interface vlan 1
```

```
Switch1(config-if-vlan1)# IPv6 address 2010:1:1::1/64
```

```
Switch1(config-if-vlan1)# IPv6 router ospf area 0
```

```
Switch1(config-if-vlan1)#exit
```

配置接口vlan2的IP地址和所属的OSPFV3域

```
Switch1(config)# interface vlan 2
```

```
Switch1(config-if-vlan2)# IPv6 address 2100:1:1::1/64
```

```
Switch1(config-if-vlan2)# IPv6 router ospf area 0
```

```
Switch1 (config-if-vlan2)#exit
```

```
Switch1(config)#exit
```

```
Switch1#
```

三层交换机Switch2:

启动OSPFv3协议，配置路由器ID

```
Switch2(config)#router IPv6 ospf
```

```
Switch2 (config-router)#router-id 192.168.2.2
```

配置接口vlan1和vlan2的IPv6地址及所属的OSPFV3域。

```
Switch2#config
Switch2(config)# interface vlan 1
Switch2(config-if-vlan1)# IPv6 address 2010:1:1::2/64
Switch2(config-if-vlan1)# IPv6 router ospf area 0
Switch2(config-if-vlan1)#exit
Switch2(config)# interface vlan 3
Switch2(config-if-vlan3)# IPv6 address 2020:1:1::1/64
Switch2(config-if-vlan3)# IPv6 router ospf area 1
Switch2(config-if-vlan3)#exit
Switch2(config)#exit
Switch2#
```

三层交换机Switch3:

启动OSPFv3协议，配置路由器ID

```
Switch3(config)#router IPv6 ospf
Switch3(config-router)#router-id 192.168.2.3
```

配置接口vlan3的IPv6地址及所属的OSPFV3域。

```
Switch3#config
Switch3(config)# interface vlan 3
Switch3(config-if-vlan3)# IPv6 address 2020:1:1::2/64
Switch3(config-if-vlan3)# IPv6 router ospf area 1
Switch3(config-if-vlan3)#exit
Switch3(config)#exit
Switch3#
```

三层交换机Switch4:

启动OSPFv3协议，配置路由器ID

```
Switch4(config)#router IPv6 ospf
Switch4(config-router)#router-id 192.168.2.4
```

配置接口vlan3的IPv6地址及所属的OSPFV3域。

```
Switch4#config
Switch4(config)# interface vlan 3
Switch4(config-if-vlan3)# IPv6 address 2030:1:1::2/64
Switch4(config-if-vlan3)# IPv6 router ospf area 0
Switch4(config-if-vlan3)#exit
Switch4(config)#exit
Switch4#
```

三层交换机Switch5:

启动OSPFv3协议，配置路由器ID

```
Switch5(config)#router IPv6 ospf
```

```
Switch5(config-router)#router-id 192.168.2.5
配置接口vlan2的IPv6地址及所属的OSPFV3域。
Switch5#config
Switch5(config)# interface vlan 2
Switch5(config-if-vlan2)# IPv6 address 2100:1:1::2/64
Switch5(config-if-vlan2)# IPv6 router ospf area 0
Switch5(config-if-vlan2)#exit
配置接口vlan3的IPv6地址及所属的OSPFV3域
Switch5(config)# interface vlan 3
Switch5(config-if-vlan3)# IPv6 address 2030:1:1::1/64
Switch5(config-if-vlan3)# IPv6 router ospf area 0
Switch5(config-if-vlan3)#exit
Switch5(config)#exit
Switch5#
```

12.5.4 OSPFv3 排错帮助

在配置、使用 OSPFv3 协议时，可能会由于物理连接、配置错误等原因导致 OSPFV3 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后在各接口上配置不同网段的 IPv6 地址；
- ☞ 然后，先启动 OSPFv3 协议（使用 router IPv6 ospf 命令）再在相应接口配置所属 OSPFV3 域；
- ☞ 接着，注意 OSPFv3 协议的自身特点——OSPFv3 骨干域（0 域）必须保证是连续的，如果不连续使用虚连接（virtual link）接来保证，所有非 0 域只能通过 0 域与其他非 0 域相连，不允许非 0 域直接相连；边界三层交换机是指该三层交换机的一部分接口属于 0 域，而另外一部分接口属于非 0 域；对于广播网等多访问网，需要选举指定三层交换机 DR；对于每个 ospfv3 进程必须配置不是 0.0.0.0 的路由器 ID。

12.6 MBGP4+

12.6.1 MBGP4+简介

MBGP4+是多协议 BGP(Multiprotocol Border Gateway Protocol)为 IPv6 进行的扩展，关于 BGP 协议的介绍请参看本手册的 BGP 协议部分。与 RIPng 和 OSPFv3 不同，由于 BGP 本身的灵活性，BGP 对 IPv6 没有单独的协议，而是在原有的 BGP 上进行了地址族的

扩展，MBGP4+对 BGP 的扩展主要体现在：

- a. 配置的邻居地址可以是 IPv6 地址；
- b. 增加 IPv6 unicast 地址族的配置。

12.6.2 MBGP4+配置

MBGP4+配置任务序列：

1. 配置 IPv6 邻居
2. 配置使能 IPv6 地址族
3. 配置 MBGP4+引入不同进程 OSPFv3 路由
 - (1) 启动 MBGP4+引入不同进程 OSPFv3 路由功能
 - (2) 显示和调试 MBGP4+引入不同进程 OSPFv3 路由功能相关信息

1. 配置 IPv6 邻居

命令	解释
BGP 协议配置模式	
neighbor <X:X::X:X> remote-as <as-id>	配置 IPv6 邻居。

2. 配置使能 IPv6 地址族

命令	解释
BGP 协议配置模式	
address-family IPv6 unicast	进入 IPv6 单播地址族。
BGP 协议地址族配置模式	
neighbor <X:X::X:X> activate no neighbor <X:X::X:X> activate	配置 IPv6 邻居使能/不使能该地址族。
exit-address-family	退出地址族配置模式。

3. 配置 MBGP4+引入不同进程 OSPFv3 路由

- (1) 启动 MBGP4+引入不同进程 OSPFv3 路由功能

命令	解释
Router ipv6 bgp 配置模式	
redistribute ospf [<process-tag>] [route-map<word>] no redistribute ospf [<process-tag>]	启动或者关闭 MBGP4+引入其他 OSPFv3 进程路由功能。

- (2) 显示和调试 MBGP4+引入不同进程 OSPFv3 路由功能相关信息

命令	解释
特权和配置模式	

show ipv6 bgp redistribute	显示 bgp4+进程引入其它外部路由的配置信息。
特权模式	
debug ipv6 bgp redistribute message send no debug ipv6 bgp redistribute message send debug ipv6 bgp redistribute route receive no debug ipv6 bgp redistribute route receive	打开或关闭 MBGP4+引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 MBGP4+进程收到 nsm 发来的路由信息的调试开关。

12.6.3 MBGP4+ 案例

SwitchB、SwitchC和SwitchD位于AS200中，SwitchA位于AS100中。SwitchA和SwitchB共享一个相同的网络段2001::/96。而SwitchB和SwitchD彼此物理上不相邻。

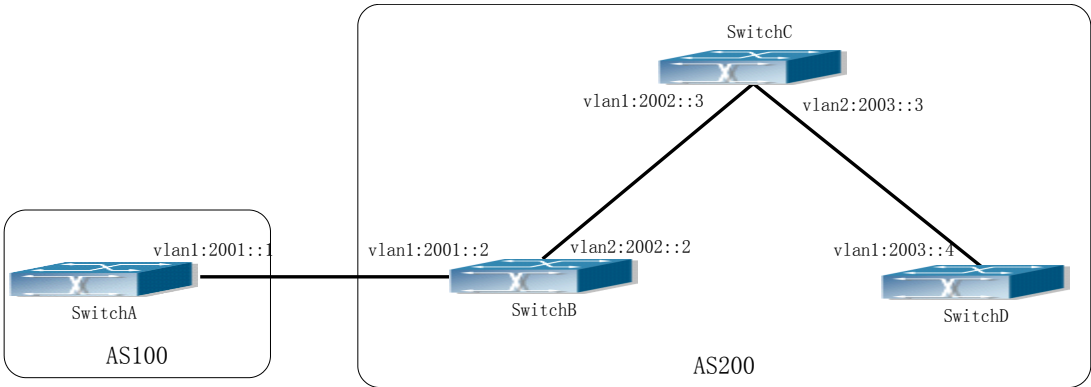


图 12-9 BGP 网络拓扑图

```
则SwitchA的配置如下：
SwitchA(config)#router bgp 100
SwitchA(config-router)#bgp router-id 1.1.1.1
SwitchA(config-router)#neighbor 2001::2 remote-as 200
SwitchA(config-router)#address-family IPv6 unicast
SwitchA(config-router-af)#neighbor 2001::2 activate
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router)#exit
SwitchA(config)#
```

SwitchB的配置如下：

```
SwitchB(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
SwitchB(config-router)#neighbor 2001::1 remote-as 100
SwitchB(config-router)#neighbor 2002::3 remote-as 200
SwitchB(config-router)#neighbor 2003::4 remote-as 200
SwitchB(config-router)#address-family IPv6 unicast
SwitchB(config-router-af)#neighbor 2001::1 activate
SwitchB(config-router-af)#neighbor 2002::3 activate
SwitchB(config-router-af)#neighbor 2003::4 activate
SwitchB(config-router-af)#exit-address-family
SwitchB(config-router)#exit
SwitchB(config)#
```

SwitchC的配置如下：

```
SwitchC(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
SwitchC(config-router)#neighbor 2002::2 remote-as 200
SwitchC(config-router)#neighbor 2003::4 remote-as 200
SwitchC(config-router)#address-family IPv6 unicast
SwitchC(config-router-af)#neighbor 2002::2 activate
SwitchC(config-router-af)#neighbor 2003::4 activate
SwitchC(config-router-af)#exit-address-family
SwitchC(config-router)#exit
```

RouteD的配置如下：

```
SwitchD(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
SwitchD(config-router)#neighbor 2003::3 remote-as 200
SwitchD(config-router)#neighbor 2002::2 remote-as 200
SwitchD(config-router)#address-family IPv6 unicast
SwitchD(config-router-af)#neighbor 2002::2 activate
SwitchD(config-router-af)#neighbor 2003::3 activate
SwitchD(config-router-af)#exit-address-family
SwitchD(config-router)#exit
```

此时SwitchB和SwitchA之间的连接是EBGP，和SwitchC、SwitchD之间的连接是IBGP。

SwitchB和SwitchD之间不必物理上相连就可以进行BGP的连接，但前提是两个交换机必须

有到达对方的路由。该路由可以通过静态路由或是IGP获得。

12.6.4 MBGP4+排错帮助

同 4.5BGP 排错帮助。

12.7 IPv6 黑洞路由

12.7.1 黑洞路由介绍

黑洞路由实际是一种特殊的静态路由，顾名思义，就是目的地址为该网段的数据报文到达设备之后，将被丢弃。

12.7.2 IPv6 黑洞路由配置任务

1. 启动 IPv6 功能
2. 配置IPv6黑洞路由

1. 启动IPv6功能

命令	解释
全局配置模式	
ipv6 enable	在交换机上启动 IPv6 功能。

2. 配置IPv6黑洞路由

命令	解释
全局配置模式	
ipv6 route <ipv6-prefix prefix-length> null0 [<precedence>] no ipv6 route <ipv6-prefix prefix-length> null0	配置 IPv6 静态黑洞路由。本命令的 no 操作为删除配置的 IPv6 黑洞路由。

12.7.3 黑洞路由举例

案例：IPv6 黑洞路由功能。

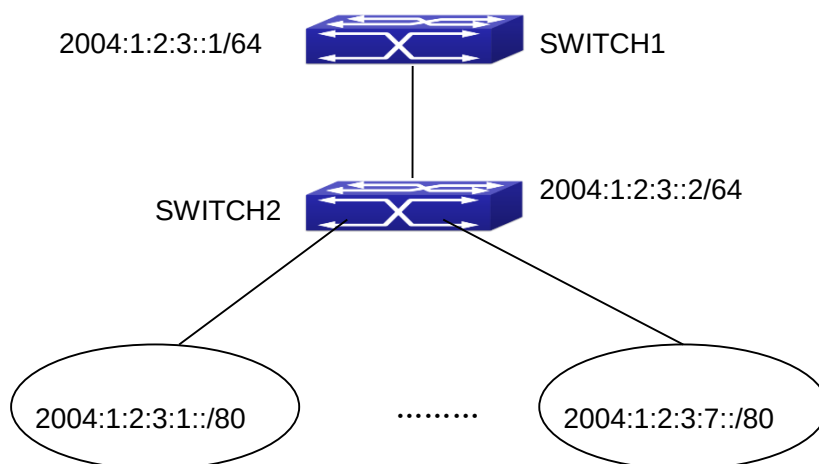


图 12-10 配置 IPv6 黑洞路由的示例图

如图所示，交换机SWITCH2配置了8个VLAN三层接口用于用户接入，分别是2004:1:2:3:1/80~2004:1:2:3:7/80，同时SWITCH2配置了一条缺省路由指向路由器SWITCH1，SWITCH1配置回程路由2004:1:2:3::/64指回到SWITCH2。正常情况下这个网络不会出现问题，但是SWITCH2的一个三层接口down掉之后，例如到网段2004:1:2:3:1/80的三层接口down掉了，有数据包发往某VLAN1时，但数据到达SWITCH2之后，没有到这个网段的路由，那么会根据缺省路由送到SWITCH1，SWITCH1在接受到IP报文之后，查询路由表，匹配上直连路由，然后又送到SWITCH2，如此往复，就在SWITCH2和SWITCH1之间循环转发，形成路由环路。为了解决这个问题，可以在SWITCH2上配置一条

12.7.4 黑洞路由排错帮助

在配置、使用黑洞路由功能时，可能会因为掩码长度不合适，以及管理距离不合适而造成无法正常使用。因此，用户应注意以下要点：

使用IPv6黑洞路由时，请首先确保全局启动了IPv6功能。

- ☞ 掩码长度最好比正常的路由的掩码长度长，这样可以确保当掩码长度长的正常路由有效的时候不干扰其工作。
- ☞ 当掩码长度与其他路由一样时，建议把黑洞路由的distance调的低一点。

如果使用检查都尝试仍无法解决的问题，那么请使用 `show ip route database`，以及 `show ip route fib`，`show l3` 等命令，然后将这些信息拷贝下来，发送给本公司技术服务中心。

12.8 IPv6 组播协议

12.8.1 PIM-DM6

12.8.1.1 PIM-DM6 介绍

PIM-DM6 (Protocol Independent Multicast, Dense Mode, 协议独立组播—密集模式) 即与协议无关组播-密模式的IPv6版本, 属于密集模式的组播路由协议, 适用于小型网络, 在这种网络环境下, 组播组的成员相对比较密集。与用于IPv4版本的PIM-DM版本相比较, 除了所使用的地址为IPv6地址以外, 没有其它的区别。因此, 在本章中对于PIM-DM和PIM-DM6不加区别, 凡是使用PIM-DM而不加说明的都是指PIM-DM的IPv6版本。

由于IPv6网络还在发展中, 有时网络中存在不支持IPv6组播的网络环境, 这就需要通过隧道进行IPv6组播的操作, 所以我们的PIM-DM6支持在配置隧道上的配置, 以IPV4单播的方式穿过不支持IPv6组播的网络。

PIM-DM 的工作过程可以概括为: 邻居发现、扩散—剪枝过程、嫁接阶段。

1. 邻居发现

PIM-DM 路由器刚开始启动时, 需要使用Hello报文来发现邻居。运行PIM-DM的各网络节点之间使用Hello报文保持联系。PIM-DM Hello报文是周期性发送的。

2. 扩散—剪枝过程 (Flooding&Prune)

PIM-DM 假设网络上的所有主机都准备接收组播数据。当某组播源S开始向组播组G发送数据时, 在路由器接收到组播报文后, 首先根据单播路由表进行RPF检查, 如果检查通过, 路由器创建一个 (S, G) 表项, 然后将组播报文向网络上所有下游PIM-DM节点转发

(Flooding)。如果没有通过RPF检查, 即组播报文是从错误的接口输入, 则将该报文丢弃。经过这个过程, 在PIM-DM组播域内, 每个节点都会创建一个 (S, G) 表项。如果下游节点没有组播组成员, 则向上游节点发剪枝 (Prune) 消息, 通知上游节点不用再转发该组播组数据。上游节点收到剪枝消息后, 就将相应的接口从其组播转发表项 (S, G) 对应的输出接口列表中删除, 这就建立了一个以源S为根的SPT (Shortest Path Tree, SPT) 树。剪枝过程最先由叶子路由器发起。

3. RPF检查

PIM-DM采用RPF检查, 利用现存的单播路由表构建一棵从数据源始发的组播转发树。当一个组播包到达时, 路由器首先判断到达路径的正确性。如果到达接口是单播路由指示的通往组播源的接口, 就认为这个组播包是从正确路径而来; 否则, 将组播包作为冗余报文丢弃。作为路径判断依据的单播路由信息可以来源于任何一种单播路由协议, 如RIP、OSPF 等发现的路由信息, 而不依赖于特定的单播路由协议。

4. Assert机制

如果处于一个LAN网段上的两台组播路由器A和B, 各自有到组播源S的接收途径, 它们

在接收到组播源S发出的组播数据报文以后，都会向LAN上转发该组播报文，这时，下游节点组播路由器C就会收到两份相同的组播报文。路由器检测到这种情况后，需要通过Assert机制来选定一个唯一的转发者。通过发送Assert报文，选出一条最优的转发路径，如果两条或两条以上路径的优先级和开销相同，则选择IP地址大的节点作为该（S，G）项的上游邻居，由它负责该（S，G）组播报文的转发。

5. 嫁接（Graft）

当被剪枝的下游节点需要恢复到转发状态时，该节点使用嫁接报文通知上游节点恢复组播数据转发。

6. PIM-DM6的状态刷新

为了减少PIM-DM6周期性的扩散-剪枝，PIM-DM6协议最新版本提供了一个选项（option），它将从组播源的第一跳路由器开始，周期性的向下面广播（S，G）状态刷新消息以完成状态刷新。当下游PIM路由器收到（S，G）状态刷新消息，它就将剪枝计时器复位，终止剪枝计时器超时，从而即保证了网络的时效性，避免了周期性的扩散-剪枝过程。

12.8.1.2 PIM-DM6 配置任务序列

- 1、启动 PIM-DM（必须）
- 2、配置静态组播表项（可选）
- 3、配置 PIM-DM 辅助参数（可选）
 - (1) 配置 PIM-DM 接口参数
 - 1) 配置 PIM-DM hello 报文间隔时间
 - 2) 配置 PIM-DM state-refresh 报文间隔时间
 - 3) 配置边缘接口
 - 4) 配置管理边界
- 4、关闭 PIM-DM 协议

1. 启动 PIM-DM 协议

在三层交换机上运行 PIM-DM 路由协议的基本配置很简单，需全局配置模式下打开 PIM 组播开关，然后在相应接口下打开 PIM-DM 开关即可。

命令	解释
全局配置模式	
ipv6 pim multicast-routing	使各个接口上的 PIM-DM 协议进入使能状态（但真正在接口上开始 PIM-DM 协议，还需下面的命令）。

然后在接口上打开 PIM-SM 开关

命令	解释
接口配置模式	

ipv6 pim dense-mode	启动本接口 PIM-DM 协议。（必须）

2. 配置静态组播表项

命令	解释
全局配置模式	
ipv6 mroute <X:X::X:X> <X:X::X:X> <ifname> <.ifname> no ipv6 mroute <X:X::X:X> <X:X::X:X> [<ifname> <.ifname>]	配置静态组播表项，其 no 命令删除静态组播表项或其出接口。

3. 配置 PIM-DM 辅助参数

（1）配置 PIM-DM 接口参数

1）配置 PIM-DM hello 报文间隔时间

命令	解释
接口配置模式	
ipv6 pim hello-interval <interval> no ipv6 pim hello-interval	配置接口 PIM-DM hello 报文间隔时间；本命令的 no 操作恢复为缺省值。

2）配置 PIM-DM state-refresh 报文间隔时间

命令	解释
接口配置模式	
ipv6 pim state-refresh origination-interval no ipv6 pim state-refresh origination-interval	配置接口 PIM-DM state-refresh 报文间隔时间；本命令的 no 操作恢复为缺省值。

3）配置边缘接口

命令	解释
接口配置模式	
ipv6 pim bsr-border no ipv6 pim bsr-border	配置接口为 PIM-DM6 边缘接口，边缘接口上，STATE REFRESH 相关消息不向该接口发送也不从该接口接收，连接的网络被认为都是该接口的直连网络。本命令的 no 操作取消该配置。

4）配置管理边界

命令	解释
接口配置模式	

ipv6 pim scope-border <500-599> <acl_name> no ipv6 pim scope-border	配置 PIM-DM6 管理边界和使用的 ACL。组播数据不向 SCOPE-BORDER 扩散，默认的情形下，认为 ffx0::/13 的范围为管理组范围，若配置了 ACL，则 ACL permit 的范围为管理组范围。acl_name 应当是 IPV6 标准 ACL 名。本命令的 no 操作取消该配置。

4. 关闭 PIM-DM 协议

命令	解释
接口配置模式	
no ipv6 pim dense-mode	在接口上关闭 PIM-DM 协议。
全局配置模式	
no ipv6 pim multicast-routing	全局关闭 PIM-DM 协议。

12.8.1.3 PIM-DM6 典型案例

如下图，将SwitchA，switchB的以太网接口加入到相应的VLAN中，并在各VLAN接口上启动PIM-DM协议。如果VLAN内同时配置了IGMP SNOOPING，则组播流量能精确到端口，而不在整个VLAN内泛洪。

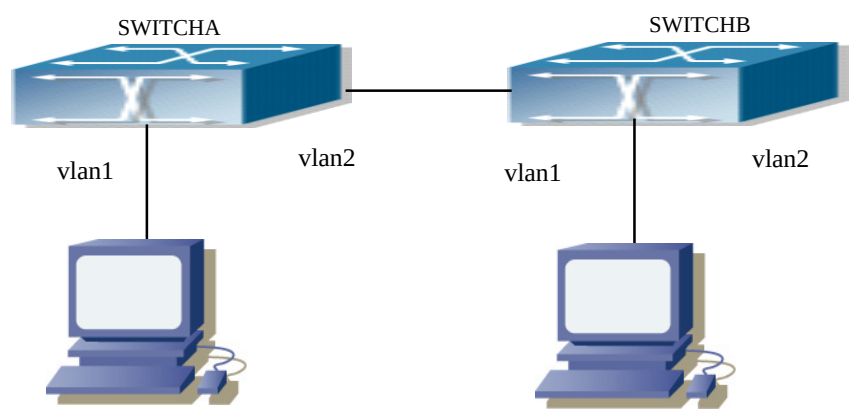


图 12-11 PIM-DM 典型环境

SwitchA 和SwitchB配置步骤如下：

(1) 配置SwitchA:

```
Switch (config)#ipv6 pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ipv6 address 2000:10:1:1::1/64
Switch(Config-if-Vlan1)# ipv6 pim dense-mode
Switch(Config-if-Vlan1)#exit
```



```
Switch (config)#interface vlan2
Switch(Config-if-Vlan2)# ipv6 address 2000:12:1:1::1/64
Switch(Config-if-Vlan2)# ipv6 pim dense-mode
```

(2) 配置SwitchB:

```
Switch (config)#ip pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-if-Vlan1)# ipv6 address 2000:12:1:1::2/64
Switch(Config-if-Vlan1)# ipv6 pim dense-mode
Switch(Config-if-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-if-Vlan2)# ipv6 address 2000:20:1:1::1/64
Switch(Config-if-Vlan2)# ipv6 pim dense-mode
```

12.8.1.4 PIM-DM6 排错帮助

在配置、使用 PIM-DM 协议时，可能会由于物理连接、配置错误等原因导致 PIM-DM 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误；
- ☞ 保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 保证全局配置模式下打开 PIM 协议(使用 ipv6 pim multicast-routing)
- ☞ 在接口上启动 PIM-DM 协议（使用 ipv6 pim dense-mode 命令）。

组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性；如果使用检查仍无法解决 PIM-DM 的问题，那么请使用 debug ipv6 pim 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.8.2 PIM-SM6

12.8.2.1 PIM-SM6 介绍

PIM-SM6（Protocol Independent Multicast，Sparse Mode）即与协议无关组播-稀疏模式的IPv6版本，属于稀疏模式的组播路由协议，主要用于组成员分布相对分散、范围较广、大规模的网络。与用于IPv4版本的PIM-SM版本相比较，除了所使用的地址为IPv6地址以外，没有其它的区别。因此，在本章中对于PIM-SM和PIM-SM6不加区别，凡是使用PIM-SM而不加说明的都是指PIM-SM的IPv6版本。与密集模式的扩散—剪枝不同，PIM-SM协议假定所有的主机都不需要接收组播数据包，只有主机明确指定需要时，PIM-SM路由器才向它转发组播数据包。

PIM-SM通过设置汇聚点RP（Rendezvous Point）和自举路由器BSR（Bootstrap Router），向所有PIM-SM路由器通告组播信息，并利用路由器的加入/剪枝信息，建立起基

于RP的共享树RPT（RP-rooted shared tree）。从而减少数据报文和控制报文占用的网络带宽，降低路由器的处理开销。组播数据沿着共享树流到该组播组成员所在的网段，当数据流量达到一定程度，组播数据流可以切换到基于源的最短路径树SPT，以减少网络延迟。

PIM-SM不依赖于特定的单播路由协议，而是使用现存的单播路由表进行RPF检查。

由于IPv6网络还在发展中，有时网络中存在不支持IPv6组播的网络环境，这就需要通过隧道进行IPv6组播的操作，所以我们的PIM-SM6支持在配置隧道上的配置，以IPv4单播的方式穿过不支持IPv6组播的网络。

1. PIM-SM 工作原理

PIM-SM的工作过程主要有：邻居发现、RP共享树（RPT）的生成、组播源注册、SPT切换等。其中，邻居发现机制与PIM-DM相同，这里不再介绍。

(1) RP 共享树（RPT）的生成

当主机加入一个组播组G时，与该主机直接相连的叶子路由器通过IGMP报文了解到有组播组G的接收者，就为组播组G计算出对应的汇聚点RP，然后向朝着RP方向的上一级节点发送加入组播组的消息（join 消息）。从叶子路由器到RP之间途经的每个路由器都会在转发表中生成（*, G）表项，表示由任意源发出的，发送至组播组G的，都适用于该表项。当RP收到发往组播组G的报文后，报文就会沿着已经建立好的路径到达叶子路由器，进而到达主机。这样就生成了以RP为根的RPT。

(2) 组播源注册

当组播源S向组播组G发送了一个组播报文时，与其直接相连的PIM-SM组播路由器接收到该报文以后，就负责将该组播报文封装成注册报文，单播给对应的RP。如果一个网段上有多个PIM-SM组播路由器，这时候将由指定路由器DR（Designated Router）负责发送该组播报文。

(3) SPT 切换

当组播路由器发现从RP发来的目的地址为G的组播报文的速率超过了阈值时，组播路由器就向朝着源S的上一级节点发送加入消息，导致RPT向SPT的切换。

2. PIM-SM 配置前准备工作

(1) 配置候选 RP

在PIM-SM网络中，可以存在多个RP（候选RP），每个候选RP（Candidate-RP, C-RP）负责转发目的地址在一定范围内的组播报文。配置多个候选RP可以实现RP负载分担。候选RP之间没有主次之分，所有的组播路由器收到BSR通告的候选RP消息后，根据相同的算法计算出与某一组播组对应的RP。

注意，一个RP可以为多个组播组服务，也可以为所有组播组服务。每个组播组在任意时刻，只能唯一地对应一个RP，不能同时对应多个RP。

(2) 配置 BSR

BSR 是PIM-SM 网络里的管理核心，它负责收集候选RP发来的信息，并把它们广播出去。

一个网络内部只能有一个BSR，但可以配置多个候选BSR（Candidate-BSR, C-BSR）。这样，一旦某个BSR发生故障后，能够切换到另外一个。C-BSR通过自动选举产生BSR。

12.8.2.2 PIM-SM6 配置任务序列

- 1、启动 PIM-SM（必须）
- 2、配置静态组播表项（可选）
- 3、配置 PIM-SM 辅助参数（可选）
 - (1) 配置 PIM-SM 接口参数
 - 1) 配置 PIM-SM hello 报文间隔时间
 - 2) 配置 PIM-SM hello 报文 holdtime 时间
 - 3) 配置 PIM-SM 邻居访问列表
 - 4) 配置接口为 PIM-SM6 边缘接口
 - 5) 配置接口为 PIM-SM6 管理边界
 - (2) 配置 PIM-SM 全局参数
 - 1) 配置交换机作为候选 BSR
 - 2) 配置交换机作为候选 RP
 - 3) 配置静态 RP
 - 4) 配置内核组播路由缓存时间
- 4、关闭 PIM-SM 协议

1. 启动 PIM-SM 协议

在三层交换机上运行 PIM-SM 路由协议的基本配置很简单，需全局配置模式下打开 PIM 组播开关，然后在相应接口下打开 PIM-SM 开关即可。

命令	解释
全局配置模式	
[no] ipv6 pim multicast-routing	使各个接口上的 PIM-SM 协议进入使能状态(但真正在接口上开始 PIM-SM 协议，还需下面的命令)，本命令的 no 操作关闭所有接口上的 PIM-SM 协议。(必须)

然后在接口上打开 PIM-SM 开关

命令	解释
接口配置模式	
[no] ipv6 pim sparse-mode [passive]	启动本接口 PIM-SM 协议，本命令的 no 操作关闭本接口 PIM-SM 协议。(必须)

2. 配置静态组播表项

命令	解释
全局配置模式	

ipv6 mroute <X:X::X:X> <X:X::X:X> <ifname> <.ifname> no ipv6 mroute <X:X::X:X> <X:X::X:X> [<ifname> <.ifname>]	配置静态组播表项，其 no 命令删除静态组播表项或其出接口。

3. 配置 PIM-SM 辅助参数

(1) 配置 PIM-SM 接口参数

1) 配置 PIM-SM hello 报文间隔时间

命令	解释
接口配置模式	
ipv6 pim hello-interval < interval> no ipv6 pim hello-interval	配置接口 PIM-SM hello 报文间隔时间；本命令的 no 操作恢复为缺省值。

2) 配置 PIM-SM hello 报文 holdtime 时间

命令	解释
接口配置模式	
ipv6 pim hello-holdtime <value> no ipv6 pim hello-holdtime	配置接口 PIM-SM hello 报文中的 holdtime 域的值；本命令的 no 操作恢复为缺省值。

3) 配置 PIM-SM 邻居访问列表

命令	解释
接口配置模式	
ipv6 pim neighbor-filter <access-list-name> no ipv6 pim neighbor-filter <access-list-name>	配置邻居访问列表。如果被列表过滤，如果已经同此邻居建立连接，则此连接马上被切断，如果没有建立连接，则这个连接不能建立。

4) 配置边缘接口

命令	解释
接口配置模式	
ipv6 pim bsr-border no ipv6 pim bsr-border	配置接口为 PIM-SM6 边缘接口，边缘接口上，BSR 相关消息不向该接口发送也不从该接口接收，连接的网络被认为都是该接口的直连网络。本命令的 no 操作取消该配置。

5) 配置管理边界

命令	解释
接口配置模式	

ipv6 pim scope-border <500-599> <acl_name> no ipv6 pim scope-border	配置 PIM-DM6 管理边界和使用的 ACL。组播数据不向 SCOPE-BORDER 扩散，默认的情形下，认为 ff00::/13 的范围为管理组范围，若配置了 ACL，则 ACL permit 的范围为管理组范围。acl_name 应当是 IPV6 标准 ACL 名。本命令的 no 操作取消该配置。

(2) 配置 PIM-SM 全局参数

1) 配置交换机作为候选 BSR

命令	解释
全局配置模式	
ipv6 pim bsr-candidate {vlan <vlan_id> <ifname> tunnel <1-50>} [hash-mask-length] [priority] no ipv6 pim bsr-candidate {vlan <vlan_id> <ifname> tunnel <1-50>} [<hash-mask-length>] [<priority>]	该命令为全局候选 BSR 配置命令，用于配置 PIM-SM 候选 BSR 的信息，以用于同其它候选 BSR 竞争 BSR 路由器；本命令的 no 操作为取消候选 BSR 的配置。

2) 配置交换机作为候选 RP

命令	解释
全局配置模式	
ipv6 pim rp-candidate {vlan<vlan-id> loopback<index> <ifname>} [<group range>] [<priority>] no ipv6 pim rp-candidate	该命令为全局候选 RP 配置命令，用于配置 PIM-SM 候选 RP 的信息，以用于同其它候选 RP 竞争 RP 路由器；本命令的 no 操作为取消候选 RP 的配置。

3) 配置静态 RP

命令	解释
全局配置模式	
ipv6 pim rp-address <rp-address> [<group-range>] no ipv6 pim rp-address <rp-address> {all <group-range>}	该命令为全局或某个组播地址范围的组播组配置静态 RP，本命令的 no 操作为取消静态 RP 的配置。

4) 配置内核组播路由缓存时间

命令	解释

全局配置模式	
ipv6 multicast unresolved-cache aging-time <value> no ipv6 multicast unresolved-cache aging-time	配置内核组播路由缓存时间；本命令的 no 操作恢复为缺省值。

4.关闭 PIM-SM 协议

命令	解释
接口配置模式	
no ipv6 pim sparse-mode	关闭 PIM-SM 协议。
全局配置模式	
no ipv6 pim multicast-routing	全局关闭 PIM-SM 协议。

12.8.2.3 PIM-SM6 典型案例

如下图所示，将switchA，switchB，switchC，switchD的以太网接口加入到相应VLAN中，并在各VLAN接口上启动PIM-SM协议。如果VLAN内同时配置了IGMP SNOOPING，则组播流量能精确到端口，而不在整个VLAN内泛洪。

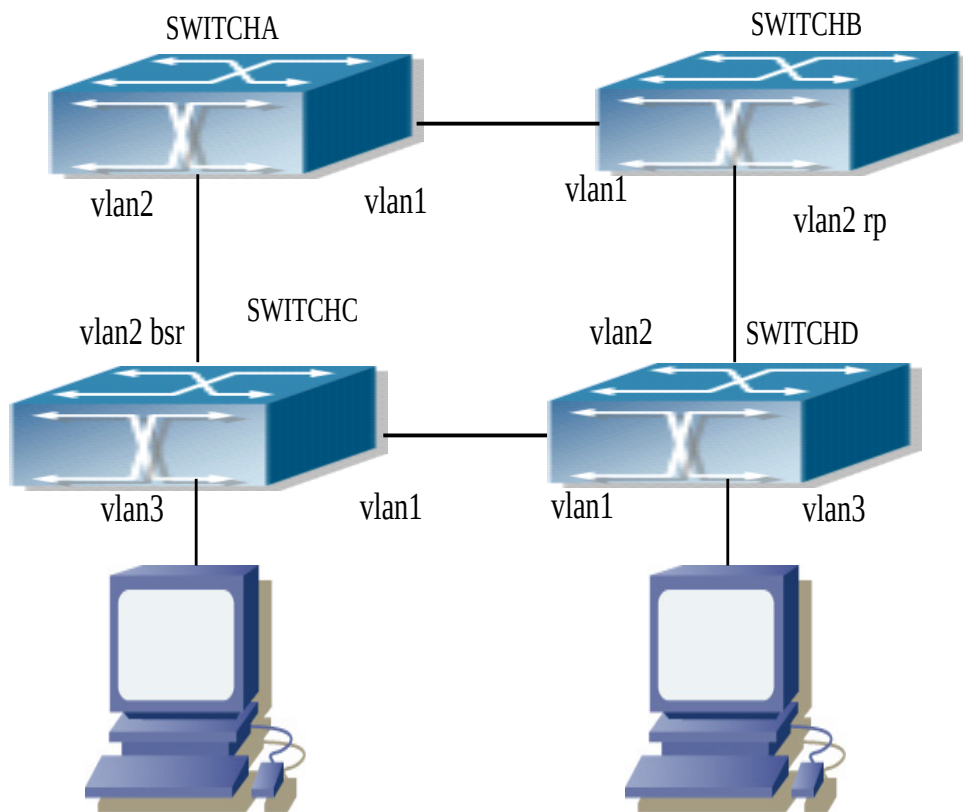


图 12-12 PIM-SM 典型环境

SwitchA 和 SwitchB, switchC, switchD 配置步骤如下:

(1) 配置 SwitchA:

```
Switch(config)#ipv6 pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:12:1:1::1/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:13:1:1::1/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
```

(2) 配置 SwitchB:

```
Switch(config)#ipv6 pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:12:1:1::2/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
```

```
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:24:1:1::2/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)# exit
Switch (config)# ipv6 pim rp-candidate vlan2
```

(3) 配置 SwitchC:

```
Switch(config)#ipv6 pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:34:1:1::3/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:13:1:1::3/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch (config)#interface vlan 3
Switch(Config-If-Vlan3)# ipv6 address 2000:30:1:1::1/64
Switch(Config-If-Vlan3)# ipv6 pim sparse-mode
Switch(Config-If-Vlan3)# exit
Switch (config)# ipv6 pim bsr-candidate vlan2 30 10
```

(4) 配置 SwitchD:

```
Switch(config)#ipv6 pim multicast-routing
Switch (config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:34:1:1::4/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch (config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:24:1:1::4/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch (config)#interface vlan 3
Switch(Config-If-Vlan3)# ipv6 address 2000:40:1:1::1/64
Switch(Config-If-Vlan3)# ipv6 pim sparse-mode
```

12.8.2.4 PIM-SM6 排错帮助

在配置、使用 PIM-SM 协议时，可能会由于物理连接、配置错误等原因导致 PIM-SM

协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性；
- ☞ PIM-SM 协议需要有 rp 和 bsr 的支持，所以首先使用 `show ipv6 pim bsr-router`，看是否有 bsr 信息，如果不存在，则需要查看是否有通向 bsr 的单播路由；
- ☞ 使用 `show ipv6 pim rp-hash` 命令查看 rp 信息是否正确，如果没有 rp 信息，也要检查单播路由。

如果使用检查仍无法解决 PIM-SM 的问题，那么请使用 `debug ipv6 pim/ debug ipv6 pim bsr` 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.8.3 ANYCAST RP v6

12.8.3.1 ANYCAST RP v6 介绍

基于 PIM 协议的 Anycast RP v6 是一种为了让 RP 失效能够尽快恢复而提供冗余的技术，Anycast-RP 地址通常是配置在 loopback 接口上的地址。

Anycast RP v6 的核心思想是在整个网络上配置的 RP 地址，存在于多台组播路由器上（通常的情况是在各个提供 ANYCAST RP 的设备上都使用 LOOPBACK 接口，并配置 RP 地址在这个接口上，使用最长的掩码），而单播路由算法将决定了 PIM 路由器总是能找到离自己最近的 RP，因此对于比较大的网络提供了更短更快的找到 RP 的路径。而一旦正在使用的某一个 RP 失效，单播路由算法将保证 PIM 路由器很快找到新的 RP 路径，从而使组播服务迅速恢复。多个 RP 会出现新的问题，即如果组播源和接收者注册到不同的 RP，会导致有的接收者不能接收到组播源数据（注册报文显然只会青睐离自己的最近的 RP）。因此，为了在各 RP 间保持联络，Anycast RP 规定离组播源最近的 RP 在收到注册报文时应将源注册信息转发通知给其他 RP，保证所有 RP 上的加入者都可以找到该组播源。

基于 PIM 协议的 Anycast RP 具体实现方法是：在每个配置了 Anycast RP 的交换机上维护一份 ANYCAST RP 的列表，并使用另一个地址作为他们之间互相识别的标志，在一台 Anycast RP 设备收到注册消息后，以标志自己的地址作为源地址向其它 Anycast RP 设备发送注册消息，以达到让其它设备也知道本源的目的。

12.8.3.2 ANYCAST RP v6 配置任务

- 1.启动 ANYCAST RP v6 功能
- 2.配置 ANYCAST RP v6

1.启动 ANYCAST RP v6 功能

命令	解释
全局配置模式	

ipv6 pim anycast-rp	启动 ANYCAST RP 功能。（必须）
no ipv6 pim anycast-rp	no 操作关闭全局 ANYCAST RP 功能。

2.配置 ANYCAST RP v6

(1) 配置候选 RP

命令	解释
全局配置模式	
ipv6 pim rp-candidate {vlan<vlan-id> loopback<index> <ifname>} [<A:B::C:D>] [<priority>] no ipv6 pim rp-candidate	目前 PIM6-SM 已允许配置 Loopback 接口作为候选 RP。（必须） 需要注意的是，ANYCAST RP 协议可以配置 Loopback 接口或普通三层 VLAN 接口作为候选 RP。为使网络中 PIM 路由器找到 RP 位置，应将候选 RP 接口加入到路由中。 no 操作取消本路由器候选 RP 配置。

(2) 配置 self-rp-address (本 RP 通信地址)

命令	解释
全局配置模式	
ipv6 pim anycast-rp self-rp-address A:B::C:D no ipv6 pim anycast-rp self-rp-address	在本路由器（作为 RP）上配置本路由器的 self-rp-address。该地址唯一标志本路由器，用于与其他 RPs 之间的通讯。（必须） self-rp-address 具体作用有两个方面： 1 当本路由器（作为 RP）收到从 DR 单播过来的注册报文，需要将该注册报文向网络中其他 RP 转发，使得网络中所有 RP 获得源（S,G）状态。转发时，本路由器修改注册报文源地址为 self-rp-address。 2 当本路由器（作为 RP）收到从其他 RP 单播转发过来的注册报文，如注册报文目的地址为本路由器 self-rp-address，则建立（S,G）状态并反方向发送注册中止报文，注册中止报文目的地址即注册报文的源地址。 注意：self-rp-address 应为本路由器上某个三层接口地址，但在配置时我们允许此接口当前不存在。self-rp-address 是唯一的。 no 操作取消本路由器用于与其他 RP 通信的 self-rp-address。

(3) 配置 other-rp-address (其他 RP 通信地址)

命令	解释
全局配置模式	
<pre> ipv6 pim anycast-rp <anycast-rp-addr> <other-rp-addr> no ipv6 pim anycast-rp <anycast-rp-addr> <other-rp-addr> </pre>	在本路由器（作为 RP）上配置 anycast-rp-addr 地址。该单播地址实际上就是在网络中多个 RP 上配置的 RP 地址，对应候选 RP 接口(或者 Loopback 接口)的地址。 anycast-rp-addr 具体作用有： 1 允许配置多个 anycast-rp-addr 地址，只有与当前配置的候选 RP 接口地址相同的 anycast-rp-addr 地址生效。此时对应该 anycast-rp-addr 地址的 other-rp-address 也才会生效。 2 配置时我们允许 anycast-rp-addr 对应的接口当前不存在。 在本路由器（作为 RP）上配置与本路由器通讯的其他 RP 的 other-rp-address。该单播地址标志其他 RP，用于与本地路由器之间的通讯。 other-rp-address 具体作用有两个方面： 1 当本路由器（作为 RP）收到从 DR 单播过来的注册报文，需要将该注册报文向网络中其他 RP 转发，使得网络中所有 RP 获得源（S，G）状态。转发时，本路由器修改注册报文目的地址为 other-rp-address。 2 对应一个 anycast-rp-addr 可以配置多个 other-rp-address，当收到 DR 单播过来的注册报文，依次向这些其他 RP 转发。 no 操作取消与本路由器通信的某个 other-rp-address。

12.8.3.3 ANYCAST RP v6 典型案例

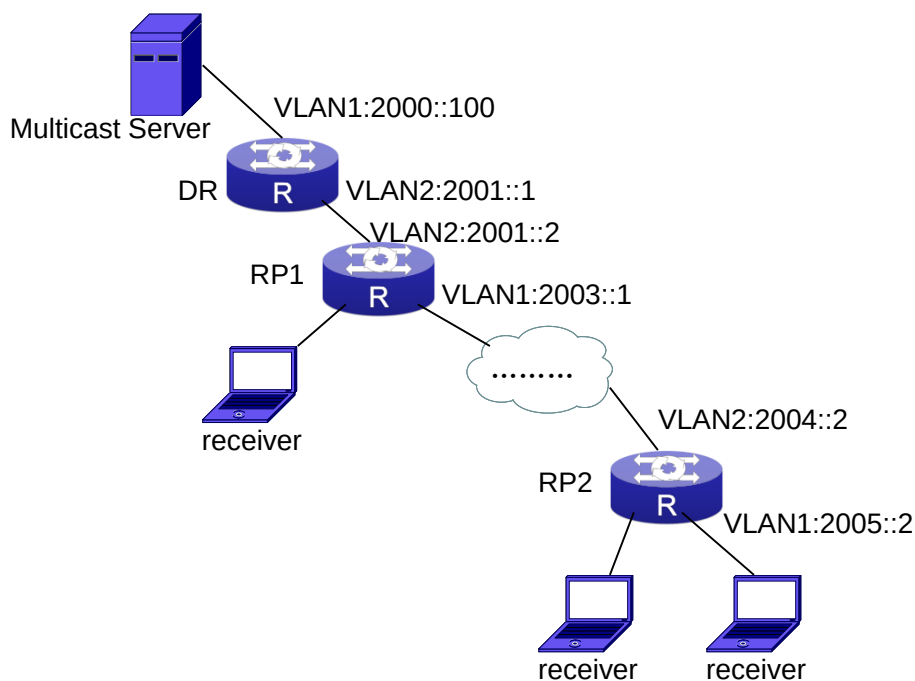


图 12-13 路由器 ANYCAST RP v6 功能

配置步骤如下：

RP1 配置：

```
Switch#config
Switch(config)#interface loopback 1
Switch(Config-if-Loopback1)#ipv6 address 2006::1/128
Switch(Config-if-Loopback1)#exit
Switch(config)#ipv6 pim rp-candidate loopback1
Switch(config)#ipv6 pim bsr-candidate vlan 1
Switch(config)#ipv6 pim multicast-routing
Switch(config)#ipv6 pim anycast-rp
Switch(config)#ipv6 pim anycast-rp self-rp-address 2003::1
Switch(config)#ipv6 pim anycast-rp 2006::1 2004::2
```

RP2 配置：

```
Switch#config
Switch(config)#interface loopback 1
Switch(Config-if-Loopback1)#ipv6 address 2006::1/128
Switch(Config-if-Loopback1)#exit
Switch(config)#ipv6 pim rp-candidate loopback1
Switch(config)#ipv6 pim multicast-routing
Switch(config)#ipv6 pim anycast-rp
```

```
Switch(config)#ipv6 pim anycast-rp self-rp-address 2004::2
Switch(config)#ipv6 pim anycast-rp 2006::1 2003::1
```

需要注意，为发布 loopback 接口路由，如使用 MBGP4+协议，可使用 network 命令；如使用 RIPng 协议，可使用 route 命令。

12.8.3.4 ANYCAST RP v6 排错帮助

在配置、使用 ANYCAST RP v6 功能时，可能会由于物理连接、配置错误等原因导致 ANYCAST RP 未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证 PIM-SM6 协议正确运行
- ☞ 保证全局配置模式下 ANYCAST RP 打开。
- ☞ 保证全局配置模式下 self-rp-address 配置正确
- ☞ 保证全局配置模式下 other-rp-address 配置正确
- ☞ 保证各接口路由由正确添加，包括作为 RP 的 loopback 接口也要正确加入路由
- ☞ 使用 show ipv6 pim anycast rp status 命令查看 ANYCAST RP 配置信息是否正确

如果使用检查仍无法解决 ANYCAST RP 的问题，那么请使用 debug ipv6 pim anycast-rp 调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.8.4 PIM-SSM6

12.8.4.1 PIM-SSM6 介绍

源指定组播(PIM-SSM6)是一种区别于传统组播的新的业务模式，它使用组播组地址和组播源地址来同时标志一个组播会话。SSM保留了传统PIM-SM6模式中的主机显示加入组播组的高效性，但是跳过了PIM-SM6模式中的共享树和RP规程。SSM6直接建立由(S,G)标志的spt树，其中G表示组播组的地址，S表示发向组播组G的特定组播源的地址。SSM6的一个（S，G）对也被称为一个频道。SSM特别适用于点到多点的组播服务，如网络体育频道，网络新闻频道等业务。在缺省情况下，SSM6组播组地址范围限制在IPV6地址范围：ff3x::/32内，但是可以根据需要对地址进行扩展。

PIM-DM6的环境中也支持PIM-SSM6。

12.8.4.2 PIM-SSM6 配置任务序列

命令	解释
全局配置模式	
ipv6 pim ssm {default range <access-list-number >}	该命令配置pim ssm 组播组地址范围；本命令的no 操作删除配置的pim ssm 组播组

no ipv6 pim ssm

12.8.4.3 PIM-SSM6 典型案例

如下图所示，将switchA，switchB，switchC，switchD的以太网接口加入到相应VLAN中，并在各VLAN接口上启动PIM-SM6协议，或者PIM-DM6协议，全局启动PIM-SSM6。本例以PIM-SM6协议为例。

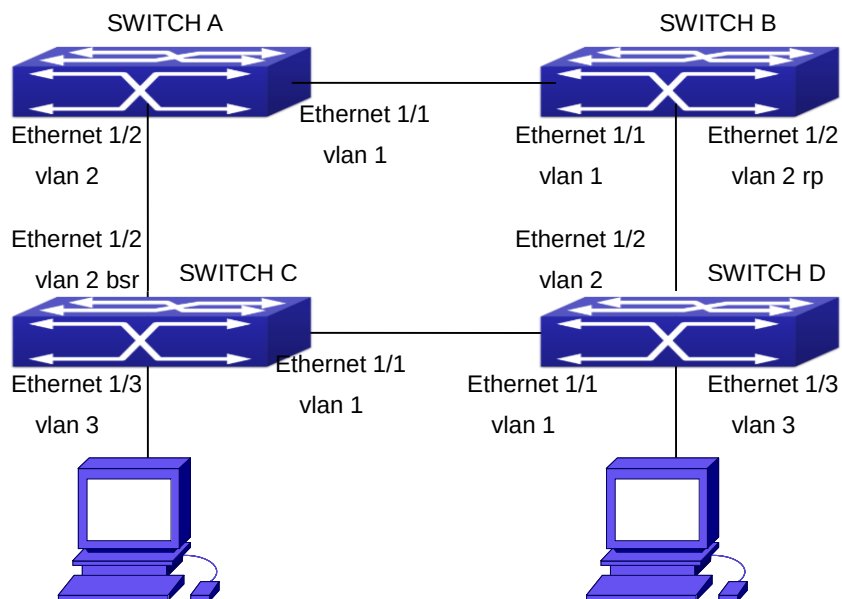


图 12-14 PIM-SSM 典型环境

switchA 和 switchB，switchC，switchD 配置步骤如下：

(1) 配置 switchA:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:12:1:1::1/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:13:1:1::1/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch(config)#ipv6 access-list 500 permit ff1e::1/64
Switch(config)#ip multicast ssm range 500
```

(2) 配置 switchB:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:12:1:1::2/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address2000:24:1:1::2/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)# exit
Switch(config)# ipv6 pim rp-candidate vlan2
Switch(config)#ipv6 access-list 500 permit ff1e::1/64
Switch(config)#ip multicast ssm range 500
```

(3) 配置 SwitchC:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:34:1:1::3/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:13:1:1::3/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
Switch(Config-If-Vlan2)#exit
Switch(config)#interface vlan 3
Switch(Config-If-Vlan3)# ipv6 address 2000:30:1:1::1/64
Switch(Config-If-Vlan3)# ipv6 pim sparse-mode
Switch(Config-If-Vlan3)# exit
Switch(config)# ipv6 pim bsr-candidate vlan2 30 10
Switch(config)#ipv6 access-list 500 permit ff1e::1/64
Switch(config)#ip multicast ssm range 500
```

(4) 配置 SwitchD:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#interface vlan 1
Switch(Config-If-Vlan1)# ipv6 address 2000:34:1:1::4/64
Switch(Config-If-Vlan1)# ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-If-Vlan2)# ipv6 address 2000:24:1:1::4/64
Switch(Config-If-Vlan2)# ipv6 pim sparse-mode
```

```
Switch(Config-If-Vlan2)#exit
Switch(config)#interface vlan 3
Switch(Config-If-Vlan3)# ipv6 address 2000:40:1:1::1/64
Switch(Config-If-Vlan3)# ipv6 pim sparse-mode
Switch(Config-If-Vlan3)#exit
Switch(config)#ipv6 access-list 500 permit ff1e::1/64
Switch(config)#ip multicast ssm range 500
```

12.8.4.4 PIM-SSM6 排错帮助

在配置、使用 PIM-SSM6 协议时，可能会由于物理连接、配置错误等原因导致 PIM-SSM6 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误；
- ☞ 保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 保证全局配置模式下 PIM6 协议打开（使用 ipv6 pim multicast-routing）；
- ☞ 保证接口上配置 PIM-SM6（使用 ipv6 pim sparse-mode）；
- ☞ 保证全局模式下配置 SSM6；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性。

如果使用检查仍无法解决问题，那么请使用 debug ipv6 pim event/debug ipv6 pim packet 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.8.5 IPv6 DCSCM

12.8.5.1 IPv6 DCSCM 介绍

IPV6 DCSCM（Destination Control and Source Control Multicast）技术主要包含三个方面，即组播信源可控、组播用户可控及面向服务优先级的策略组播。

IPV6 DCSCM 的组播信源可控技术主要采取以下的方式进行：

1. 在边缘交换机上，如果配置的源受控组播，只有指定源发出的指定组的组播数据才能通过。
2. 对于处于 PIM-SM 核心地位的 RP 交换机，对于指定源及指定组以外的 REGISTER 信息，直接发送 REGISTER_STOP，而不允许建立表项。（该任务在 PIM-SM 模块中实现）。

IPV6 DCSCM 技术的组播用户可控技术的实现基于对用户发出的 MLD report 报文的控制，因此进行控制的模块是 MLD snooping 和 MLD 模块，其控制逻辑包括以下三种，即根据发送报文的 VLAN+MAC 地址进行控制，根据发送报文的 IP 地址进行控制和根据报文进入的

端口进行控制。其中MLD snooping可以同时使用上述三种方式进行控制，而MLD模块由于处于三层，仅针对发送报文的IP地址进行控制。

IPV6 DCSCM 技术的面向服务优先级的策略组播采用下述的方式：对于限定范围的组播数据，在接入端就设置用户指定的优先级，使得数据在干道端口（TRUNK）上以更高的优先级发送，从而在整个网络上保证其以用户指定的优先级传送。

12.8.5.2 IPv6 DCSCM 配置任务序列

1. 源受控的配置
2. 目的受控的配置
3. 组播策略的配置

1. 源受控的配置

源受控的配置分三个部分，首先是全局启动源受控，全局启动源受控的命令如下：

命令	解释
全局配置模式	
ipv6 multicast source-control(必须) no ipv6 multicast source-control	全局启动源控制，本命令的 no 操作全局关闭源控制。值得注意的是，全局启动源控制后，所有组播报文都默认丢弃。所有源控制配置都要在全局启动后才可以进行，而只有关闭所有已经配置的规则后，才可以全局关闭源控制。

其次是配置源控制的规则，它使用与 ACL 相同的方式配置，使用 8000—8099 的 ACL 号，每个规则号最多可配置 10 个规则，值得注意的是，这些规则是有顺序的，先配置的在最前面，一旦所配置的规则得到匹配，其后面的规则将不会起作用，所以全局允许的规则一定要配置在最后。其命令如下

命令	解释
全局配置模式	
[no] ipv6 access-list <8000-8099> {deny permit} {{<source/M>} {host-source <source-host-ip>} any-source} {{<destination/M> } {host-destination <destination-host-ip>} any-destination}	用于配置源受控使用的规则，该规则只有应用到指定的端口上时才可以生效。使用其 NO 形式可以删除指定的规则。

最后是把配置的规则配置到指定的端口上。

注意：由于配置的规则要占据硬件的表项，配置过多的规则可能导致由于底层表项满而配置失败，因此建议用户尽可能使用简单的规则。配置命令如下：

命令	解释
端口配置模式	
[no] ipv6 multicast source-control access-group <8000-8099>	用于把源受控使用的规则配置到端口上, 其 NO 形式取消该配置。

2. 目的受控的配置

目的受控的配置与源受控配置相似, 也是分三个步骤。

首先要全局打开目的受控, 由于目的受控要防止未获授权的用户接收到组播数据, 因此配置全局目的受控后, 交换机对收到的组播数据不再广播, 因此, 应当避免在启动了目的受控的交换机上同一 VLAN 内连接两台或以上的其它三层交换机。其配置命令如下:

命令	解释
全局配置模式	
multicast destination-control(必须)	全局启动 IPV4 和 IPV6 目的控制, 本命令的 no 操作全局关闭目的控制。所有其它的配置都只有在全局启动后才能生效。

其次是配置目的控制规则, 它与源控制相似, 只是使用 9000—10099 的 ACL 号。

命令	解释
全局配置模式	
[no] ipv6 access-list <9000-10099> {deny permit} {{<source/M>}{host-source <source-host-ip>}}any-source {{<destination/M>}{host-destination <destination-host-ip>}}any-destination	用于配置源受控使用的规则, 该规则只有应用到指定的源 IP 或 VLAN-MAC、端口上时才可以生效。使用其 NO 形式可以删除指定的规则

最后要把该规则配置到指定的源 IP、源 VLAN MAC 或指定端口上。这里需要注意的是, 由于上面所说的情况, 只有在启动 MLD-SNOOPING 的情况下才可以全面使用这些规则, 而如果没有启动 MLD-SNOOPING, 则只有源 IP 的规则才能在 MLD 协议使用。其配置命令如下:

命令	解释
端口配置模式	
[no] ipv6 multicast destination-control access-group <9000-10099>	用于把目的受控使用的规则配置到端口上, 其 NO 形式取消该配置。
全局配置模式	
[no] ipv6 multicast destination-control <1-4094> <macaddr> access-group <9000-10099>	用于把目的受控使用的规则配置到指定的 VLAN-MAC 上, 其 NO 形式取消该配置。

[no] ipv6 multicast destination-control <IPADDRESS/M> access-group <9000-100999>	用于把目的受控使用的规则配置到指定的源 IPv6 地址/掩码上，其 NO 形式取消该配置。

3. 组播策略的配置

组播策略使用为指定的组播数据指定优先级的方式达到保证特定用户需求的效果，值得注意的是组播数据只有在干道端口上传输才能保证数据一直得到重点照顾。其配置很简单，只有一个命令，即为指定的组播设定优先级，命令如下：

命令	解释
全局配置模式	
[no] ipv6 multicast policy <IPADDRESS/M> <IPADDRESS/M> cos <priority>	配置组播策略，为特定范围的源、组指定优先级，范围为<0-7>。

12.8.5.3 IPv6 DCSCM 典型案例

1. 源受控

为了防止一台边缘交换机随意发布组播数据，我们在边缘交换机上配置，只有在端口 Ethernet1/4 的交换机才允许发送组播数据，且数据的组必须是 ff1e::1。而上连端口 Ethernet1/25 可以不受限制的转发组播数据，则我们可以进行如下配置。

```
Switch(config)#ipv6 access-list 8000 permit any-source ff1e::1
Switch(config)#ipv6 access-list 8001 permit any any
Switch(config)#ipv6 multicast source-control
Switch(config)#interface Ethernet1/4
Switch(Config-If-Ethernet1/4)#ipv6 multicast source-control access-group 8000
Switch(config)#interface Ethernet1/25
Switch(Config-If-Ethernet1/25)#ipv6 multicast source-control access-group 8001
```

2. 目的受控

我们要限制地址在 fe80::203:fff:fe01:228a/64 网段的用户不能加入 ff1e::1/64 的组，我们可以做如下配置：

首先在其所在的 VLAN(这里认为是 VLAN2)内要启动 MLD snooping。

```
Switch(config)#ipv6 mld snooping
Switch(config)#ipv6 mld snooping vlan 2
```

然后配置相关的目的控制访问列表，并配置指定的 IPv6 地址使用该访问列表。

```
Switch(config)#ipv6 access-list 9000 deny any ff1e::1/64
Switch(config)#ipv6 access-list 9000 permit any any
Switch(config)#ipv6 multicast destination-control
Switch(config)#ipv6 multicast destination-control fe80::203:fff:fe01:228a/64 access-group 9000
```

这样，该网段的用户就只能加入 2ff1e::1/64 以外的组了。

3. 组播策略

服务器 2008::1 正在组 ff1e::1 上发布重要的组播数据，我们可在其接入交换机上配置如下：

```
Switch(config)#ipv6 multicast policy 2008::1/128 ff1e::1/128 cos 4
```

这样该组播流在通过本交换机的 TRUNK 端口通向其它交换机时，将拥有值为 4 的优先级（通常这已经比较高了，更高的可能是协议数据了，若设置更高的优先级，在该组播数据太多时，有可能造成交换机协议行为的不正常）。

12.8.5.4 IPv6 DCSCM 排错帮助

IPv6 DCSCM 模块本身作用与 ACL 相似，发生问题通常与不当的配置有关，请您详细阅读上面的说明。

注：12GT、12GB、4GX24TX 板卡不支持组播源受控。

12.8.6 MLD

12.8.6.1 MLD 介绍

MLD(Multicast Listener Discovery)是服务于IPv6组播的组播组成员（接受者）发现协议，类似于IPv4组播应用中的IGMP协议，其中MLD协议版本1类似于IGMP协议的版本2，MLD协议版本2类似于IGMP协议的版本3，我们目前实现了MLDv2。

参与IPv6 组播的主机可以在任意位置、任意时间、成员总数不受限制地加入或退出组播组。组播交换机不需要也不可能保存所有主机的成员关系，它只是通过MLD 协议了解每个接口连接的网段上是否存在某个组播组的接收者，即组成员。而主机方只需要保存自己加入了哪些组播组。

MLD 在主机与路由器之间是不对称的：主机需要响应组播交换机的MLD查询报文，即以成员资格报告报文响应；交换机周期性发送成员资格查询报文，然后根据收到的响应报文确定某个特定组在自己所在子网上是否有主机加入，并且当收到主机的退出组的报告时，发出特定组的查询，以确定某个特定组是否已无成员存在。

MLD协议的协议报文可以分为三种类型，即Query, Report和Done（对应于IGMPv2的Leave）。与IGMPV2一样，其Query报文包括了普通查询和特定组查询两种。其普通查询使用所有主机的组播地址FF02::1作为目的地址，而组地址填充为0，其特定查询使用其组地址作为目的地址。MLD的组播地址使用130、131、132作为数据类型，分别表示上述的三种报文。其它的逻辑与IGMPv2也基本相同。

MLDv2增加了V2的report类型，其组播地址使用FF02::16,并使用143作为数据类型，其它逻辑与IGMPV3相同。

12.8.6.2 MLD 配置任务序列

- 1、启动 MLD（必须）
- 2、配置 MLD 辅助参数（可选）
 - (1) 配置 MLD 组 group 参数
 - 1) 配置 MLD 组过滤条件
 - 2) 配置 MLD 加入组
 - 3) 配置 MLD 加入静态组
 - (2) 配置 MLD 查询参数
 - 1) 配置 MLD 发送查询报文的时间间隔
 - 2) 配置 MLD 查询的最大反应时间
 - 3) 配置 MLD 查询的超时时间
 - (3) 配置 MLD 版本
- 3、关闭 MLD 协议

1. 启动 MLD 协议

在三层交换机上没有启动 MLD 协议的专门命令，在相应接口下启动任何一种 IPv6 组播协议，则 MLD 自动随之启动。

命令	解释
全局配置模式	
ipv6 pim multicast-routing	启动全局 IPv6 组播协议，是启动 MLD 协议的必要前提，相应命令的 no 操作关闭 ipv6 组播协议以及 MLD 协议。（必须）

命令	解释
接口配置模式	
ipv6 pim dense-mode ipv6 pim sparse-mode	启动 MLD 协议，相应命令的 no 操作关闭 MLD 协议。（必须）

2. 配置 MLD 辅助参数

- (1) 配置 MLD 组 group 参数
 - 1) 配置 MLD 组过滤条件
 - 2) 配置 MLD 加入组
 - 3) 配置 MLD 加入静态组

命令	解释
接口配置模式	
ipv6 mld access-group <acl_name> no ipv6 mld access-group	配置接口对 MLD 组的过滤条件；本命令的 no 操作取消过滤条件。

ipv6 mld join-group <address> no ipv6 mld join-group <address> ipv6 mld join-group <address> mode <include exclude> source <.X:X::X:X> no ipv6 mld join-group <X:X::X:X> source <.X:X::X:X>	配置接口加入某个 MLD 组或组源；本命令的 no 操作取消加入。
ipv6 mld static-group <group_address> [source <source_address>] no ipv6 mld static-group <group_address> [source <source_address>]	配置接口加入某个 IGMP 静态组或组源；本命令的 no 操作取消加入。

(2) 配置 MLD 查询参数

- 1) 配置 MLD 发送查询报文的时间间隔
- 2) 配置 MLD 查询的最大响应时间
- 3) 配置 MLD 查询的超时时间

命令	解释
接口配置模式	
ipv6 mld query-interval <time_val> no ipv6 mld query-interval	配置周期发送 MLD 查询消息的时间间隔；本命令的 no 操作恢复缺省值。
ipv6 mld query-max-response-time <time_val> no ipv6 mld query-max-response-time	配置接口对 MLD 查询的最大反应时间；本命令的 no 操作恢复缺省值。
ipv6 mld query-timeout <time_val> no ipv6 mld query-timeout	配置接口对 MLD 查询的超时时间；本命令的 no 操作恢复缺省值。

(3) 配置 MLD 版本

命令	解释
全局配置模式	
ipv6 mld version <version> no ipv6 mld version	配置接口上 MLD 版本；本命令的 no 操作恢复为缺省值。

3. 关闭 MLD 协议

命令	解释
接口配置模式	
no ipv6 pim dense-mode no ipv6 pim sparse-mode no ipv6 pim multicast-routing (全局配置模式)	关闭 MLD 协议。

12.8.6.3 MLD 典型案例

如下图，将SwitchA 和SwitchB 的以太网端口加入相应的VLAN，并在各VLAN接口上启动PIM6。

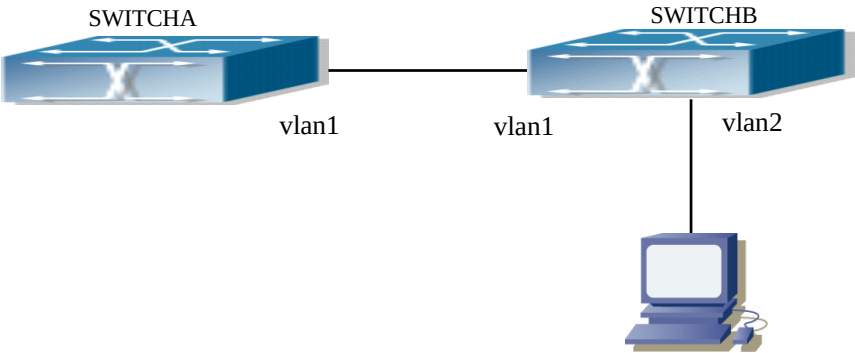


图 12-15 MLD 网络拓扑示意图

SwitchA 和 SwitchB 配置步骤如下：

(1) 配置 SwitchA:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#ipv6 pim rp-address 3FFE::1
Switch(config)#interface vlan 1
Switch(Config-If-Vlan1)#ipv6 address 3FFE::1/64
Switch(Config-If-Vlan1)#ipv6 pim sparse-mode
```

(2) 配置 SwitchB:

```
Switch(config)#ipv6 pim multicast-routing
Switch(config)#ipv6 pim rp-address 3FFE::1
Switch(config)#interface vlan1
Switch(Config-If-Vlan1)#ipv6 address 3FFE::2/64
Switch(Config-If-Vlan1)#ipv6 pim sparse-mode
Switch(Config-If-Vlan1)#exit
Switch(config)#interface vlan2
Switch(Config-If-Vlan2)#ipv6 address 3FFA::1/64
```

```
Switch(Config-If-Vlan2)#ipv6 pim sparse-mode  
Switch(Config-If-Vlan2)#ipv6 mld query-timeout 150
```

12.8.6.4 MLD 排错帮助

在配置、使用 MLD 协议时，可能会由于物理连接、配置错误等原因导致 MLD 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 然后，确保在接口上启动一种组播协议；
- ☞ 注意保证同网段上各路由器定时器时间是一致的，通常我们建议您使用默认的设置；
- ☞ 组播协议需使用单播路由进行 RPF 检查，因此必须首先确保单播路由的正确性。

如果使用检查仍无法解决 MLD 的问题，那么请使用 `debug ipv6 mld event/packet` 等调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

12.8.7 MLD Snooping

12.8.7.1 MLD Snooping 介绍

MLD（Multicast Listener Discovery）组播接听者发现协议，用于实现IPv6的组播。MLD被支持组播的网络设备（如路由器）用来进行组播接听者发现，也被想加入某组播组的接听者用来通知路由器接收某个组播地址的数据包，而这些都是通过MLD消息交换来完成的。路由器首先利用一个可寻址到所有接听者的组播地址（即ff02::1）发送一条MLD组播接听者查询（Multicast listener Query）消息。若一个接听者希望加入某组播地址，它就利用该组播地址的地址回应一条MLD接听者报告（Multicast listener Report）消息。

MLD Snooping即MLD侦听。交换机通过MLD Snooping来限制组播流量的泛滥，只把组播流量转发给与组播设备相连的端口。交换机侦听组播路由器和接听者之间的MLD消息，根据侦听结果维护组播转发表，而交换机根据组播转发表来决定组播包的转发。

交换机实现了MLD Snooping功能，并且支持MLD v2，这样，用户可以用交换机实现IPv6组播。

12.8.7.2 MLD Snooping 配置任务

- 1.启动MLD Snooping功能
- 2.配置MLD Snooping

- 1.启动MLD Snooping功能

命令	解释
全局配置模式	
ipv6 mld snooping no ipv6 mld snooping	启动全局MLD Snooping功能。 no 操作关闭全局MLD Snooping功能。

2.配置MLD Snooping

命令	解释
全局配置模式	
ipv6 mld snooping vlan <vlan-id> no ipv6 mld snooping vlan <vlan-id>	启动指定VLAN的MLD Snooping功能。 no 操作 关 闭 指 定 VLAN 上 的 MLD Snooping。
ipv6 mld snooping vlan <vlan-id> limit {group <g_limit> source <s_limit>} no ipv6 mld snooping vlan <vlan-id> limit	设置MLD Snooping可加入组的个数和每个组中源个数的最大值。 no 操作恢复默认值。
ipv6 mld snooping vlan <vlan-id> l2-general-querier no ipv6 mld snooping vlan <vlan-id> l2-general-querier	将该VLAN设为二层普通查询者，每一个网段推荐配置一个二层普通查询者。 no 操作取消二层普通查询者设置。
ipv6 mld snooping vlan <vlan-id> mrouter-port interface <interface -name> no ipv6 mld snooping vlan <vlan-id> mrouter-port interface <interface -name>	设置指定VLAN内的静态mrouter端口。 no 操作取消mrouter端口设置。
ipv6 mld snooping vlan <vlan-id> mrouter-port learnpim6 no ipv6 mld snooping vlan <vlan-id> mrouter-port learnpim6	打开指定VLAN根据pimv6报文学习mrouter-port的功能；本命令的 no 操作为关闭指定VLAN根据pimv6报文学习mrouter-port的功能。
ipv6 mld snooping vlan <vlan-id> mrpt <value> no ipv6 mld snooping vlan <vlan-id> mrpt	设置mrouter端口存活时间， no 操作恢复默认值。
ipv6 mld snooping vlan <vlan-id> query-interval <value> no ipv6 mld snooping vlan <vlan-id> query-interval	设置查询间隔， no 操作恢复默认值。
ipv6 mld snooping vlan <vlan-id> immediate-leave no ipv6 mld snooping vlan <vlan-id> immediate-leave	设置指定VLAN的MLD Snooping具有快速离开组播组功能， no 操作取消immediate-leave设置。
ipv6 mld snooping vlan <vlan-id> query-mrsp <value>	设置查询的最大响应时间， no 操作恢复默认值。

<code>no ipv6 mld snooping vlan <vlan-id> query-mrsp</code>	
<code>ipv6 mld snooping vlan <vlan-id> query-robustness <value> no ipv6 mld snooping vlan <vlan-id> query-robustness</code>	设置查询鲁棒值，no操作恢复默认值。
<code>ipv6 mld snooping vlan <vlan-id> suppression-query-time <value> no ipv6 mld snooping vlan <vlan-id> suppression-query-time</code>	设置抑制查询时间值，no操作恢复默认值。
<code>ipv6 mld snooping vlan <vlan-id> static-group <X:X::X:X> [source <X:X::X:X>] interface [ethernet port-channel] <IFNAME> no ipv6 mld snooping vlan <vlan-id> static-group <X:X::X:X> [source <X:X::X:X>] interface [ethernet port-channel] <IFNAME></code>	设置静态组源，no 操作取消设置的静态组源

12.8.7.3 MLD Snooping 典型案例

案例1：MLD Snooping功能。

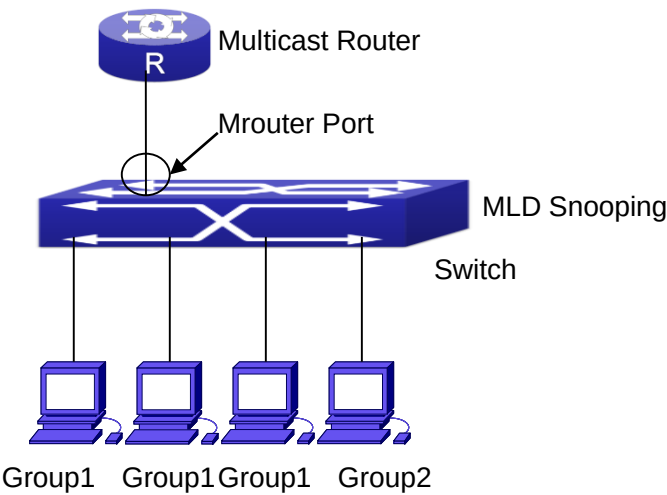


图 12-16 打开交换机 MLD Snooping 功能图

如图所示，switch上配置VLAN 100包含端口1、2、6、10、12。四台主机分别连在端口2、6、10、12上，组播路由器连在端口1上。假设我们需要在VLAN 100上配置MLD Snooping，

缺省情况下，交换机的全局MLD Snooping功能和各VLAN上的MLD Snooping功能都不打开。因此，需要打开全局下的MLD Snooping功能，同时在VLAN 100上打开MLD Snooping，还需要设置VLAN 100的端口1为mrouter端口。

配置步骤如下：

```
switch#config
```

```
switch (config)#ipv6 mld snooping
```

```
switch (config)#ipv6 mld snooping vlan 100
```

```
switch (config)#ipv6 mld snooping vlan 100 mrouter-port interface ethernet 1/1
```

组播配置：

假设有两个组播服务器Multicast Server 1和Multicast Server 2，其中组播服务器1上提供节目1、节目2，组播服务器2上提供节目3，分别使用组地址Group1，Group2和Group 3。四台主机上同时运行组播应用软件，连在端口2、6上的两台主机播放节目1，连在端口10上的主机播放节目2，连在端口12上的主机播放节目3。

MLD Snooping侦听结果：

VLAN 100上MLD Snooping建立的组播表显示：其中端口1、2、6在（Multicasting Server 1， Group1）中，端口1、10在（Multicast Server 1, Group 2）中，端口1、12在（Multicast Server2, Group 3）中。

四台主机都能正常地收到自己感兴趣的节目，端口2、6不会收到节目2和节目3的流量，端口10不会收到节目1和节目3的流量，端口12不会收到节目1和节目2的流量。

案例2：MLD L2-general-querier

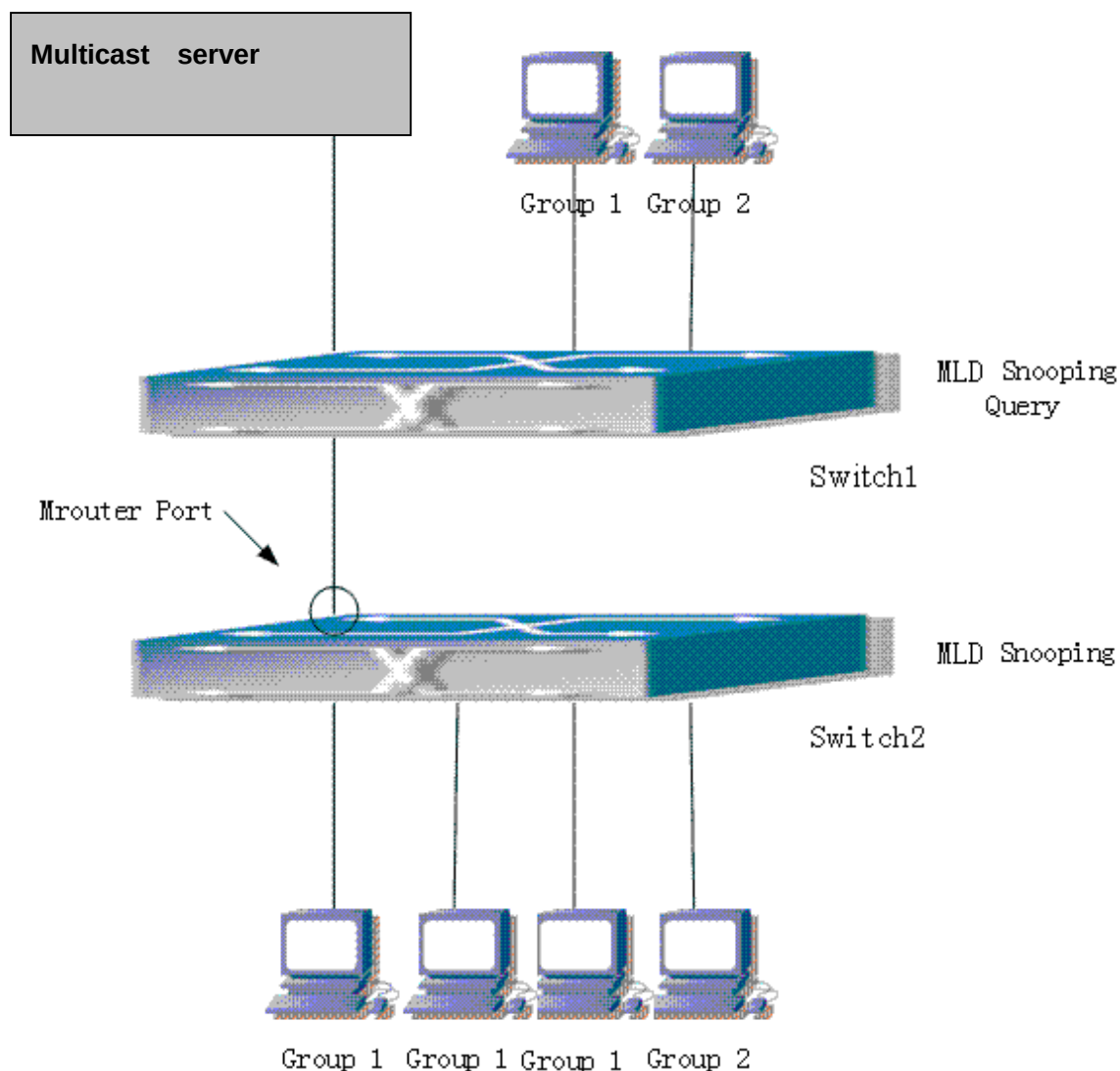


图 12-17 交换机作为 MLD Querier 功能图

switch B的配置与案例1中交换机相同，交换机switch 1取代案例1中的Multicast Router。假设其上配置VLAN 60包含端口1、2、10、12。端口1连接组播服务器，端口2连接switch2。为了定期发Query，需要打开全局下的MLD Snooping功能，同时需要执行mld snooping vlan 60 l2-general-querier，将VLAN 60设置成为二层普通查询者。

配置步骤如下：

```
switchA#config
switchA(config)#ipv6 mld snooping
switchA(config)#ipv6 mld snooping vlan 60
switchA(config)#ipv6 mld snooping vlan 60 l2-general-querier
switchB#config
switchB(config)#ipv6 mld snooping
switchB(config)#ipv6 mld snooping vlan 100
switchB(config)#ipv6 mld snooping vlan 100 mrouter interface ethernet 1/1
```

组播配置：

同案例1。

MLD Snooping侦听结果：

与案例1相似。

案例 3： 与三层组播协议混合运行

我们将案例 1 中的 SWITCH 替换为 ROUTER，配置与原 SWITCH 配置相同，组播配置与 MLD snooping 侦听结果与案例 1 相同。在 ROUTER 上全局启动 IPv6 三层组播(PIM-SM6)，同时在 VLAN 100 上启动 PIM SM6。(与上层组播路由器使用相同 PIM 模式)

配置步骤如下：

```
switch#config
switch(config)#ipv6 pim multicast-routing
switch(config)#interface vlan 100
switch(config-if-vlan100)#ipv6 pim sparse-mode
```

此时由于存在三层组播协议，MLD Snooping 不再下发表项，只负责以下几件事情：

☞ 删除二层组播表项

向三层提供端口查询函数，参数为 VLAN，S，G

当三层 MLD 关闭时，重新下发二层组播表项

通过观察三层 IP6MC 表项，发现三层组播出入表项仍然可以精确到端口，保证了 MLD snooping 与三层组播协议的混合运行。

12.8.7.4 MLD Snooping 排错帮助

在配置、使用MLD Snooping功能时，可能会由于物理连接、配置错误等原因导致MLD Snooping未能正常运行。因此，用户应注意以下要点：

- ☞ 应该保证物理连接的正确无误
- ☞ 保证全局配置模式下MLD Snooping打开（使用ipv6 mld snooping）
- ☞ 保证全局配置模式下VLAN上配置MLD Snooping（使用ipv6 mld snooping vlan <vlan-id>）
- ☞ 确保同一网段内配置了一个VLAN作为二层普通查询者，或者确保配置了静态mrouter
- ☞ 使用show ipv6 mld snooping vlan <vid>命令查看MLD Snooping信息是否正确

12.9 IPv6 安全 RA

12.9.1 IPv6 安全 RA 简介

在 IPv6 网络中，一般的网络拓补结构是由路由器、二层交换机、IPv6 主机构成。通常路由器公告 RA，包括链路前缀、链路 MTU 等信息，IPv6 主机收到 RA 后，生成链路地址，并将默认路由指向发送 RA 的路由器，从而可以进行 IPv6 网络通信。如果恶意的 IPv6 主机发送 RA，使正常的 IPv6 用户将默认路由指向恶意的 IPv6 主机用户，那么就可截获别的用户信息，影响网络安全。同时正常的用户获得另外的地址，使自己无法链接网络。所以我们要实现安全 RA 功能，在交换机的端口通过命令配置拒绝接收恶意的 RA 报文，这样在一定程度上防止恶意 RA 的转发，可以避免影响网络的正常工作。

12.9.2 IPv6 安全 RA 配置任务序列

1. 全局启动 IPv6 安全 RA
2. 端口启动 IPv6 安全 RA
3. 显示和调试 IPv6 安全 RA 相关信息

1. 全局启动 IPv6 安全 RA

命令	解释
全局配置模式	
ipv6 security-ra enable no ipv6 security-ra enable	全局模式启动和关闭 IPv6 安全 RA。

2. 端口启动 IPv6 安全 RA

命令	解释
端口配置模式	
ipv6 security-ra enable no ipv6 security-ra enable	端口模式启动和关闭 IPv6 安全 RA。

3. 显示和调试 IPv6 安全 RA 相关信息

命令	解释
特权配置模式	
debug ipv6 security-ra no debug ipv6 security-ra	打开 IPv6 安全 RA 模块的调试信息，本命令的 NO 操作关闭 IPv6 安全 RA 调试信息输出。
show ipv6 security-ra [interface <interface-list>]	显示非信任端口和全局安全 RA 是否启动。

12.9.3 IPv6 安全 RA 典型案例

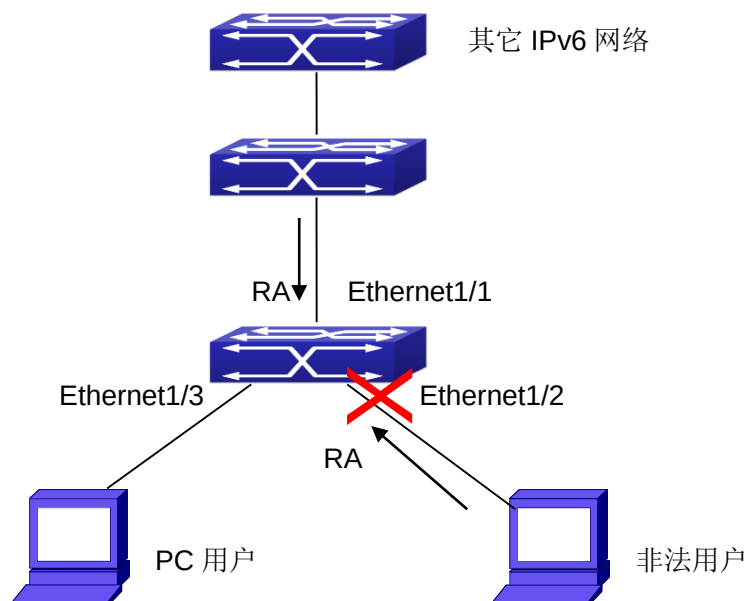


图 12-18 IPv6 安全 RA 案例示意图

说明：图中如果非法用户公告 RA，正常的 PC 用户将收到 RA，将默认路由指向恶意的 IPv6 主机用户，并改变自己的地址，自己将无法链接网络。我们要在交换机的 1/2 端口设置安全 RA，那么非法用户的 RA 报文将不会影响到正常的用户。

SWITCH 配置任务序列：

```
Switch#config
```

```
Switch(config)#ipv6 security-ra enable
```

```
Switch(Config-If-Ethernet1/2)#ipv6 security-ra enable
```

12.9.4 IPv6 安全 RA 排错帮助

IPv6 安全 RA 的功能非常简单，如果在配置 IPv6 安全 RA 后，发现功能与预期不符：

- ☞ 判断是否因为交换机是否正确配置。
- ☞ 判断是否因为交换机配置了与安全 RA 功能冲突的规则使 RA 报文转发。

12.10 SAVI

12.10.1 SAVI 介绍

SAVI(Source Address Validation Improvement)功能是一种提供节点源地址粒度级的安全验证方式，它通过监听相关协议报文(如 ND 协议，DHCPv6 协议)的交互过程，利用 CPS (Control Packet Snooping)机制，提取节点可信任的信息(如端口，MAC 地址等信息)即锚信息，然后将节点源地址与锚信息进行绑定，并下发绑定信息对应的过滤规则，放行匹

配过滤规则的报文, 丢弃不匹配过滤规则的报文, 从而达到对节点源地址合法性检测的目的。

按照协议报文类型, SAVI 功能主要包括: ND Snooping 功能, DHCPv6 Snooping 功能和 RA Snooping 功能; ND Snooping 功能主要探测 ND 协议报文, 主要建立节点以无状态地址配置获取的 IPv6 地址的绑定; DHCPv6 Snooping 功能主要探测 DHCPv6 协议报文, 主要建立节点以有状态地址自动配置过程获取的 IPv6 地址的绑定; RA Snooping 功能主要用于防止非法节点冒充路由器发送伪造 RA 报文, 以欺骗链路合法性节点的目的。

12.10.2 SAVI 配置

SAVI 的配置任务序列

1. 开启或关闭 SAVI 功能
2. 开启或关闭 SAVI 应用场景功能
3. 配置 SAVI 绑定功能
4. 配置 SAVI max-dad-delay 全局参数
5. 配置 SAVI max-dad-prepare-delay 全局参数
6. 配置 SAVI max-slaac-life 全局参数
7. 配置 SAVI 保护绑定的生存周期参数
8. 开启或关闭 SAVI 前缀检查功能
9. 配置链路 IPv6 地址前缀功能
10. 配置 IPv6 地址过滤表项数目参数
11. 配置 SAVI 冲突绑定检测模式
12. 开启或关闭端口用户验证功能
13. 开启或关闭端口 DHCPv6 报文信任功能
14. 开启或关闭端口 ND 报文信任功能
15. 配置绑定数目参数

1. 开启或关闭 SAVI 全局功能

命令	解释
全局配置模式	
savi enable no savi enable	打开全局 SAVI 功能; 本命令的 no 操作为关闭全局 SAVI 功能。

2. 开启或关闭 SAVI 应用场景全局功能

命令	解释
全局配置模式	
savi ipv6 {dhcp-only slaac-only dhcp-slaac} enable no savi ipv6 {dhcp-only slaac-only dhcp-slaac} enable	使能 SAVI 应用场景功能; 本命令的 no 操作为关闭 SAVI 应用场景功能。

3. 配置 SAVI 绑定功能

命令	解释
全局配置模式	
savi ipv6 check source binding ip <ip-address> mac <mac-address> interface <if-name> {type [slaac dhcp] lifetime <lifetime> type static} no savi ipv6 check source binding ip <ip-address> interface <if-name>	手工配置动态或静态绑定功能；本命令的 no 操作为删除手工配置的动态或静态绑定功能。该命令可在 savi enable, slaac-only enable, dhcp-only enable 和 dhcp-slaac enable 任何一个全局功能下进行配置。

4. 配置 SAVI max-dad-delay 全局参数

命令	解释
全局配置模式	
savi max-dad-delay <max-dad-delay> no savi max-dad-delay	配置 SAVI 绑定处于 DETECTION 状态最大生存周期，本命令 no 操作为恢复默认值。

5. 配置 SAVI max-dad-prepare-delay 全局参数

命令	解释
全局配置模式	
savi max-dad-prepare-delay <max-dad-prepare-delay> no savi max-dad-prepare-delay	配置 SAVI 绑定重新探测的最大生存周期，本命令 no 操作为恢复默认值。

6. 配置 SAVI max-slaac-life 全局参数

命令	解释
全局配置模式	
savi max-slaac-life <max-slaac-life> no savi max-slaac-life	配置 slaac 动态绑定处于 BOUND 状态的生存周期，no 命令为恢复默认值。

7. 配置 SAVI 保护绑定的生存周期参数

命令	解释
全局配置模式	
savi timeout bind-protect <protect-time> no savi timeout bind-protect	配置端口由 up 状态变成 down 状态后，对该端口绑定进行保护的生存周期；no 命令为恢复默认值。

8. 开启或关闭 SAVI 前缀检查功能

命令	解释
全局配置模式	
ipv6 cps prefix check enable no ipv6 cps prefix check enable	使能 SAVI 地址前缀检查功能；本命令的 no 操作为关闭 SAVI 地址前缀检查功能。

9. 配置链路 IPv6 地址前缀功能

命令	解释
全局配置模式	
ipv6 cps prefix <ip-address> vlan <vid> no ipv6 cps prefix <ip-address>	手工配置一条 IPv6 地址链路前缀，本命令 no 操作为删除一条配置的 IPv6 地址链路前缀。

10. 配置 IPv6 地址过滤表项数目参数

命令	解释
全局配置模式	
savi ipv6 mac-binding-limit <limit-num> no savi ipv6 mac-binding-limit	配置同一 MAC 地址对应的 SAVI 动态绑定数目；no 操作为恢复默认值。注意：该配置绑定表项限制仅适用于动态绑定数目，而不限静态绑定的数目。

11. 配置 SAVI 冲突绑定检测模式

命令	解释
全局配置模式	
savi check binding <simple probe> mode no savi check binding mode	配置冲突绑定检查模式；no 命令为删除配置的冲突绑定的检查模式。

12. 开启或关闭端口用户验证功能

命令	解释
端口配置模式	
savi ipv6 check source [ip-address mac-address ip-address mac-address] no savi ipv6 check source	开启用户控制验证功能，本命令的 no 操作为关闭用户验证功能。

13. 开启或关闭端口 DHCPv6 报文信任功能

命令	解释
端口配置模式	
ipv6 dhcp snooping trust no ipv6 dhcp snooping trust	开启端口 DHCPv6 报文信任功能，本命令的 no 操作为关闭端口 DHCPv6 信任功能，端口由信任端口转变为非信任端口。

14. 开启或关闭端口 ND 报文信任功能

命令	解释
端口配置模式	
ipv6 nd snooping trust no ipv6 nd snooping trust	开启端口为 slaac trust 和 RA trust 功能，本命令的 no 操作为删除 slaac trust 和 RA trust 功能。

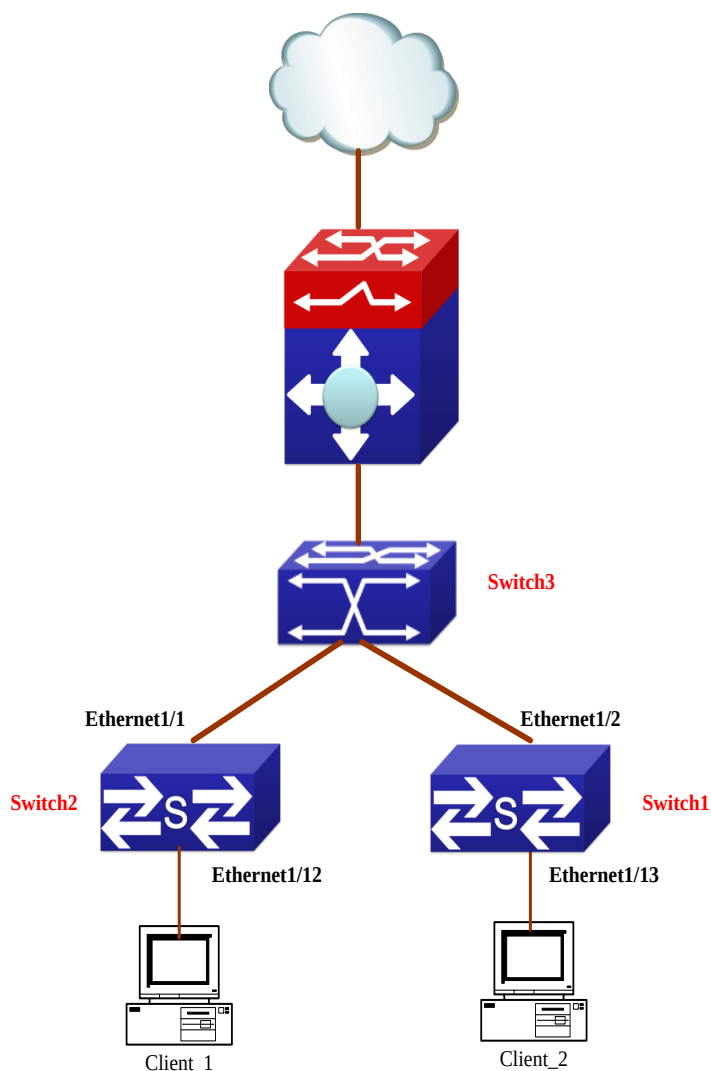
15. 配置绑定数目参数

命令	解释
端口配置模式	
savi ipv6 binding num <limit-num> no savi ipv6 binding num	配置端口的绑定数目，本命令的 no 操作为还原默认值。注意：该端口绑定数据限制仅包括动态绑定数目，不对静态绑定数目进行限制。

12.10.3 SAVI 典型应用

在实际工程应用中，SAVI 功能一般应用在接入层交换机中，对直连链路上的用户节点进行源地址合法性检测；SAVI 功能典型的应用场景主要有四种：DHCP-Only 场景、Slaac-Only 场景、DHCP-Slaac 场景和静态绑定场景。在具体的网络环境中，用户可根据实际需求选择对应的场景。在双栈网络中，SAVI 功能也可以和 IPv4 的 DHCP snooping 配合使用，同时实现 IPv4 和 IPv6 源地址验证。

SAVI 功能典型的应用网络拓扑环境



其中 Client_1 和 Client_2 代表两台不同用户的 PC 机, 每台 PC 机都已安装 IPv6 协议, 直连交换机 Switch1 和 Switch2, 直连端口分别为 Ethernet1/12, Ethernet1/13, 并开启 SAVI 的源地址检查功能。Switch1 和 Switch2 上联端口分别为 Ethernet1/1, Ethernet1/2, 并开启 DHCP 报文信任和 ND 报文信任功能。汇聚交换机 Switch3 开启 DHCPv6 服务器功能和路由公告功能。

SAVI DHCP-SLAAC 组合场景配置步骤:

```
Switch1>enable
```

```
Switch1#config
```

```
Switch1(config)#savi enable
```

```
Switch1(config)#savi ipv6 dhcp-slaac enable
```

```
Switch1(config)#savi check binding probe mode
```

```
Switch1(config)#interface ethernet1/1
```

```
Switch1(config-if-ethernet1/1)#ipv6 dhcp snooping trust
```

```
Switch1(config-if-ethernet1/1)#ipv6 nd snooping trust
```

```
Switch1(config-if-ethernet1/1)#exit
Switch1(config)#interface ethernet1/12-20
Switch1(config-if-port-range)#savi ipv6 check source ip-address mac-address
Switch1(config-if-port-range)#savi ipv6 binding num 4
Switch1(config-if-port-range)#exit
Switch1(config)#exit
Switch1#write
```

12.10.4 SAVI 排错帮助

在确保 SAVI 客户机硬件、线缆等没有问题的前提下，检查可能存在的情况和建议的解决方法：

- ☞ 若交换机开启 SAVI 功能后，没有正确的过滤 IPv6 报文，首先检查交换机上 SAVI 全局功能是否已开启，若没有开启，首先开启 SAVI 全局功能；然后根据实际的应用场景，启用对应的 SAVI 场景全局功能，并同时开启端口验证功能；
- ☞ 若交换机开启 SAVI 功能后，客户端无法正确的获取上联 DHCPv6 服务器的分配的 IPv6 地址，首先检查开启交换机上联 DHCPv6 服务器的接口是否配置 DHCP 端口信任功能，若没有启用，启用 DHCP 端口信任功能；
- ☞ 若交换机开启 SAVI 功能后，无法建立新增用户节点的绑定，首先检查用户直连交换机端口是否配置端口绑定数目限制功能，是否绑定数目已达到最大值；若已超过端口最大绑定数目限制，建议配置较大的端口绑定数目限制；
- ☞ 若配置较大端口绑定数目限制后，仍无法建立新增用户节点的绑定，则应检查用户直连交换机端口是否配置同一 MAC 地址对应绑定数目限制功能，同一 MAC 地址对应的绑定数目是否已达到最大值；若已超过最大绑定数目限制，建议配置较大的绑定数目限制。

12.11 IPv6 VRRPv3

12.11.1 VRRPv3 简介

VRRPv3 是针对 IPv6 的虚拟路由冗余协议，它是基于 IPv4 环境下的 VRRP(VRRPv2)来设计的，下面对其进行简要介绍。

在基于 TCP/IP 协议的网络中，为了保证不直接物理连接的设备之间的通信，必须指定路由。目前常用的指定路由的方法有两种：一种是通过路由协议（比如：内部路由协议 RIP 和 OSPF）动态学习；另一种是静态配置。在每一个终端都运行动态路由协议是不现实的，大多客户端操作系统平台都不支持动态路由协议，即使支持也受到管理开销、收敛度、安全性等许多问题的限制。因此普遍采用对终端 IP 设备静态路由配置，一般是给终端设备指定一个或者多个默认网关(Default Gateway)。静态路由的方法简化了网络管理的复杂度和减

轻了终端设备的通信开销，但是它仍然有一个缺点：如果作为默认网关的路由器损坏，所有使用该网关为下一跳主机的通信必然要中断。即便配置了多个默认网关，如不重新启动终端设备，也不能切换到新的网关。采用虚拟路由冗余协议 (VRRP) 可以很好的避免静态指定网关的缺陷。

在 VRRP 协议中，有两组重要的概念：VRRP 路由器和虚拟路由器，主控路由器(Master)和备份路由器(Backup)。VRRP 路由器是指运行 VRRP 的路由器，是物理实体；虚拟路由器是指 VRRP 协议创建的，是逻辑概念。一组 VRRP 路由器协同工作，共同构成一台虚拟路由器。该虚拟路由器对外表现为一个具有唯一固定 IP 地址和 MAC 地址的逻辑路由器。处于同一个 VRRP 组中的路由器具有两种互斥的角色：主控路由器和备份路由器，一个 VRRP 组中有且只有一台处于主控角色的路由器，可以有一个或者多个处于备份角色的路由器。VRRPv3 协议使用选择策略从路由器组中选出一台作为主控，负责 ND(Neighbor Discovery, 邻居发现)邻居请求消息(IPv4 为 ARP)回应和转发 IP 数据包，组中的其它路由器作为备份的角色处于待命状态。当由于某种原因主控路由器发生故障时，备份路由器能在几秒钟的时延后升级为主路由器。由于此切换非常迅速而且不用改变 IP 地址和 MAC 地址，故对终端使用者系统是透明的。

在 IPv6 环境下，局域网中的主机通常通过邻居发现协议(NDP)而学习到默认网关，这是基于定期接收路由器通告消息来实现的。IPv6 的邻居发现协议中存在一种叫做邻居不可达性检测(Neighbor Unreachability Detection)的机制，该机制通过向邻居节点发送单播的邻居请求消息来检测邻居节点是否失败。为了降低发送邻居请求消息的开销，邻居请求消息只被发送到那些正在发送流量的邻居节点，并且是在一段时间内没有路由器 UP 状态的指示以后才发送的。在邻居不可达性检测机制中，如果采用默认参数，需要花费大约 38 秒才能检测到某个路由器不可达，这种延迟对用户来说非常可观，并且可能引起某些传输协议超时。与 NDP 相比，VRRP 提供了快速的默认网关切换。在 VRRP 中，备份路由器能够在大约 3 秒(默认参数)的时间内接管失效的主控路由器，并且该过程不需要与主机进行任何交互，即对主机是透明的。

12.11.1.1 VRRPv3 报文结构

VRRPv3 具有自己的报文格式，VRRP 报文用来在备份组内交流路由器的优先级和 Master 的状态，VRRPv3 报文是封装在 IPv6 报文中来发送的，它们被发送到指定的 IPv6 组播地址。VRRPv3 的报文格式如图 1 所示。其中封装 VRRPv3 报文的 IPv6 报文中的源地址为报文出接口的 IPv6 链路本地地址，目的地址为 IPv6 组播地址(分配给 VRRPv3 的组播地址为 FF02:0:0:0:0:0:0:12)，跳数限制必须为 255，下一报头为 112(表示 VRRP 报文)。

VRRPv3 报文的各个域所代表的含义如下：

Version: VRRPv3 版本，值为 3；

Type: VRRP 报文类型，只有一种类型 ADVERTISEMENT，值为 1；

Virtual Rtr ID: 虚拟路由器 ID；

Priority: 优先级，取值范围为 0~255；

Count IPv6 Addr: VRRPv3 报文中包含的 IPv6 地址个数，最小值为 1；

Rsvd: 保留域，值为 0；

Adver Int: VRRPv3 报文通告时间间隔，单位为厘秒；

Checksum: 校验和，计算范围为整个 VRRPv3 报文和一个 IPv6 伪头(具体请参考 RFC2460)；

IPv6 Address(es): 一个或多个关联于虚拟路由器的 IPv6 地址,地址个数和“Count IPv6 Addr”相同，其中第一个地址必须为虚拟路由器的虚拟 IPv6 地址。

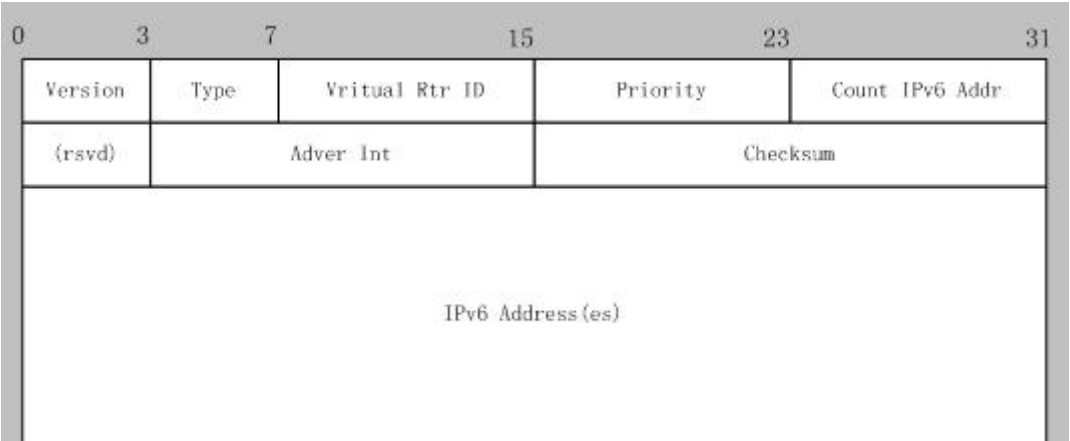


图 12-19 VRRPv3 报文

12.11.1.2 VRRPv3 工作原理

VRRPv3 的工作原理和 VRRPv2 的工作原理基本相同，主要是通过 VRRP 通告报文的交互来进行的，现简要描述如下。

一个 VRRP 路由器有唯一的标识：VRID，范围为 1-255。该路由器对外表现为唯一的虚拟 MAC 地址，VRRPv3 的虚拟 MAC 地址的格式为 00-00-5E-00-02-{VRID}(VRRPv2 的虚拟 MAC 地址格式为 00-00-5E-00-01-{VRID})。主控路由器负责对 ND 邻居请求(VRRPv2 为 ARP 请求)用该 MAC 地址做应答。这样，无论如何切换，保证给终端设备的是唯一一致的 IP 和 MAC 地址，减少了切换对终端设备的影响。

VRRP 控制报文只有一种：VRRP 通告(advertisement)。它使用 IP 组播数据包进行封装,组地址格式为 FF02:0:0:0:0:0:XXXX:XXXX。为了保持与 VRRPv2 中组地址(224.0.0.18)的一致性，VRRPv3 通告报文所使用的组播地址可以为 FF02:0:0:0:0:0:0:12，发布范围只限于同一局域网内。这保证了 VRID 在不同网络中可以重复使用。为了减少网络带宽消耗，只有主控路由器才可以周期性的发送 VRRP 通告报文。备份路由器在连续 3 个通告间隔内收不到 VRRP 通告或收到优先级为 0 的通告后启动新一轮的 VRRP 选举。

在 VRRP 路由器组中，按优先级选举主控路由器，VRRP 协议中优先级范围是 0-255。若 VRRP 路由器的 IP 地址和虚拟路由器的接口 IP 地址相同，则称该虚拟路由器作 VRRP 组中的 IP 地址所有者；IP 地址所有者自动具有最高优先级：255。优先级 0 一般用在 IP 地址所有者主动放弃主控者角色时使用。可配置的优先级范围为 1-254。优先级的配置原则可以依据链路的速度和成本、路由器性能和可靠性以及其它管理策略设定。主控路由器的选举

中，高优先级的虚拟路由器获胜，因此，如果在 VRRP 组中有 IP 地址所有者，则它总是作为主控路由的角色出现。对于相同优先级的候选路由器，按照 IP 地址大小顺序选举(地址大者优先)。VRRP 还提供了优先级抢占策略，如果配置了该策略，高优先级的备份路由器便会剥夺当前低优先级的主控路由器而成为新的主控路由器。

为了避免 Ping 虚拟 IP 时返回物理 MAC 地址的错误，这里规定虚拟 IP 不能为接口的真实 IP。这样，所有参与备份组选举的接口默认都是 Backup。

12.11.2 VRRPv3 配置

12.11.2.1 配置任务序列

10. 创建/删除虚拟路由器（必须）
11. 配置 VRRPv3 的虚拟 IPv6 地址和接口（必须）
12. 启动/关闭虚拟路由器（必须）
13. 配置 VRRPv3 辅助参数（可选）
 - (1) 配置 VRRPv3 抢占模式
 - (2) 配置 VRRPv3 优先级
 - (3) 配置 VRRPv3 通告时间间隔
 - (4) 配置 VRRPv3 的监控接口

1. 创建/删除虚拟路由器

命令	解释
全局配置模式	
router ipv6 vrrp <vrid> no router ipv6 vrrp <vrid>	创建/删除 VRRPv3 虚拟路由器。

2. 配置 VRRPv3 虚拟 IPv6 地址和接口

命令	解释
VRRPv3 协议配置模式	
virtual-ipv6 <ipv6-address> Interface {Vlan <ID> IFNAME } no virtual-ipv6 interface	配置 VRRPv3 虚拟 IPv6 地址和接口，no 命令删除虚拟 IPv6 地址和接口。

3. 启动/关闭虚拟路由器

命令	解释
VRRPv3 协议配置模式	
enable	启动该虚拟路由器。
disable	关闭该虚拟路由器。

4. 配置 VRRPv3 辅助参数

- (1) 配置 VRRPv3 抢占模式

命令	解释
VRRPv3 协议配置模式	
preempt-mode {true false}	配置 VRRPv3 的抢占模式。

(2) 配置 VRRPv3 优先级

命令	解释
VRRPv3 协议配置模式	
priority < priority >	配置 VRRPv3 优先级。

(3) 配置 VRRPv3 通告时间间隔

命令	解释
VRRPv3 协议配置模式	
advertisement-interval <time>	配置 VRRPv3 通告时间间隔(单位为厘秒)。

(4) 配置 VRRPv3 的监控接口

命令	解释
VRRPv3 协议配置模式	
circuit-failover {vlan <ID> IFNAME} <value_reduced>	配置 VRRPv3 的监控接口，no 命令删除监控接口。
no circuit-failover	

12.11.3 VRRPv3 典型案例

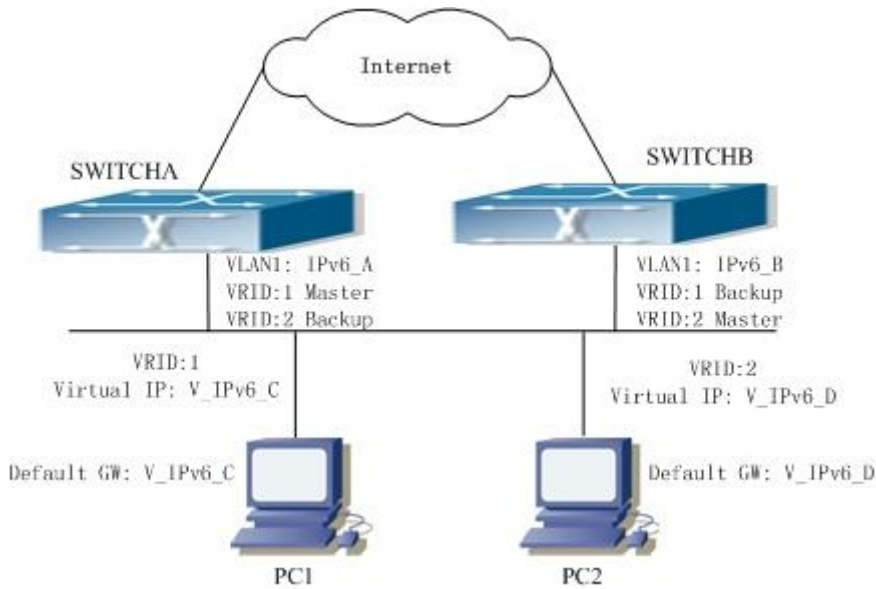


图 12-20 VRRPv3 典型网络拓扑

如图所示，其中交换机 A 和交换机 B 互为备份，交换机 A 为备份组 1 的 Master，同时是备份组 2 的 Backup，交换机 B 为备份组 2 的 Master，同时是备份组 1 的 Backup。交换

机 A 和交换机 B 的 VLAN1 接口 IPv6 地址分别为“IPv6_A”和“IPv6_B”(IPv6_A 和 IPv6_B 建议为同一个网段), 备份组 1 和备份组 2 的虚拟 IPv6 地址分别为“V_IPv6_C”和“V_IPv6_D”, 主机 1 和主机 2 的缺省 IPv6 网关地址分别配置为“V_IPv6_C”和“V_IPv6_D”(在实际应用当中, 主机的 IPv6 网关地址通常是通过路由公告而自动学习到的, 这样主机的 IPv6 下一跳地址就具有一定的随机性)。这样不仅实现了路由备份, 还具有局域网内的流量分担功能。

SwitchA 的配置:

```
SwitchA(config)#interface vlan 1
SwitchA(config)#router ipv6 vrrp 1
SwitchA(config-router)#virtual-ipv6 fe80::2 interface vlan 1
SwitchA(config-router)#priority 150
SwitchA(config-router)#enable
SwitchA(config)#router ipv6 vrrp 2
SwitchA(config-router)#virtual-ipv6 fe80::3 interface vlan 1
SwitchA(config-router)#enable
```

SwitchB 的配置:

```
SwitchB(config)#interface vlan 1
SwitchB(config)#router ipv6 vrrp 2
SwitchB(config-router)#virtual-ipv6 fe80::3 interface vlan 1
SwitchB(config-router)#priority 150
SwitchB(config-router)#enable
SwitchB(config)#router ipv6 vrrp 1
SwitchB(config-router)#virtual-ipv6 fe80::2 interface vlan 1
SwitchB(config-router)#enable
```

12.11.4 VRRPv3 排错帮助

在配置、使用 VRRPv3 协议时, 可能会由于物理连接、配置错误等原因导致 VRRPv3 协议未能正常运行。因此, 用户应注意以下要点:

- ☞ 首先应该保证物理连接的正确无误;
- ☞ 其次, 保证接口和链路协议是 UP (使用 show ipv6 interface 命令);
- ☞ 然后, 确保开启了 IPv6 转发功能(使用 ipv6 enable 命令);
- ☞ 另外, 确保在接口上已启动了 VRRPv3 协议;
- ☞ 检查同一备份组内的不同路由器 (或三层以太网交换机) 配置的 timer 时间是否相同;
- ☞ 检查同一备份组内的虚拟 IPv6 地址是否相同。

