

CCIE Routing and Switching Study Notes

Christian Kyony

Version 1.0, 2015-10-12

Table of Contents

Dedication	1
Part I : Layer 2 Technologies	2
1. Ethernet	3
1.1. Frame Formats	3
1.2. Ethernet MAC Addresses	4
1.2.1. Types Of MAC Addresses	5
1.3. RJ-45 Pinouts and Cat5 Wiring	5
1.4. Auto-Negotiation, Speed and Duplex	6
1.5. Standards	7
1.6. Ethertype	8
1.7. Troubleshooting	8
1.7.1. Problems and Approaches	9
2. CDP, LLDP	12
2.1. CDP	12
2.1.1. Packet Format	12
2.1.2. CDP Operations	12
2.1.3. CDP Updates	13
2.1.4. Version	13
2.1.5. Monitoring and Maintenance	13
2.1.6. Neighbors	13
2.2. LLDP	14
2.2.1. LLDP Global State	14
2.2.2. LLDP Interfaces	15
2.2.3. Neighbors	15
2.2.4. Timers	16
2.2.5. TLV	16
2.2.6. Network-Policy Profiles	17
2.2.7. LLDP-MED	17
2.2.8. Wired Location Service	18
3. VLANs and Trunking	20
3.1. Normal and Extended VLANs	20
3.1.1. VLAN Numbering	20
3.1.2. VLAN Trunks	20
3.1.3. Basic Configuration	21
3.1.4. VLAN State	21
3.1.5. Troubleshoot	21
3.2. Voice VLANs	21
3.3. Private VLANs	22

3.3.1. Primary VLANs	22
3.3.2. Isolated VLANs	22
3.3.3. Community VLANs	23
3.3.4. Private-Vlan Host Port	23
3.3.5. Private-VLAN Promiscuous Ports	23
3.3.6. Private-VLAN SVI	23
3.3.7. Private-Vlan Accross Multiple Switches	24
3.3.8. Interaction with Other Features	24
3.3.9. PVLAN Edge or Protected Ports	25
3.4. VTP	26
3.4.1. VTP Version	26
3.4.2. VTP Message Format	27
3.4.3. VTP Domain	30
3.4.4. Configuration Revision Number	30
3.4.5. VTP Modes	31
3.4.6. VTP Security	32
3.4.7. VTP Pruning	32
3.4.8. Troubleshooting	34
3.5. DTP	34
3.5.1. Trunking Between a Switch and a Router	35
3.5.2. Verify	35
3.5.3. DTP and VTP	36
3.6. ISL	36
3.6.1. Frame	36
3.7. IEEE 802.1Q	38
3.7.1. Frame Format	38
3.7.2. Ethernet Frame Size with 802.1Q Tagging	39
3.7.3. Native VLAN	39
3.8. 802.1Q-In-Q Tunneling	39
3.8.1. Frame	40
4. Spanning Tree Protocols	43
4.1. 802.1d Common Spanning Tree	43
4.1.1. BPDU	44
4.1.2. Root Bridge	45
4.1.3. Root Port	45
4.1.4. Designated Port	46
4.1.5. Blocking Ports	46
4.1.6. Convergence	46
4.1.7. Topology Change Notification	47
4.1.8. Timers	47
4.2. PVST+ Per-Vlan STP	48

4.3. Optimizing, Improving Spanning Tree	48
4.3.1. PortFast	48
4.3.2. UplinkFast	49
4.3.3. BackboneFast	50
4.3.4. BPDU Guard	50
4.3.5. BPDU Filter	51
4.3.6. Loop Guard	52
4.3.7. Root Guard	54
4.4. 802.1w Rapid STP	55
4.4.1. RSTP Link Types	55
4.4.2. RSTP Port Types	56
4.4.3. RSTP Port States	56
4.4.4. RSTP Port Roles	56
4.4.5. Proposal/Agreement Process	57
4.4.6. Topology Change Handling	57
4.5. 802.1s Multiple Spanning Trees	58
4.5.1. MST Frame format	59
4.5.2. MST Region	59
4.5.3. MST region ports	60
4.5.4. MST Revision Number	60
4.5.5. MST Instance	60
4.6. Protecting Against Unidirectional Link Issues	62
4.6.1. UDLD	62
4.6.2. Bridge Assurance	64
4.6.3. Dispute Mechanism	65
4.6.4. Unicast Flooding	65
4.7. Troubleshooting	65
4.7.1. Flapping Port That Is Generating BPDUs with the TCN Bit Set	65
4.8. Alternatives to STP	65
4.9. Troubleshooting	65
5. Link Aggregation	67
5.1. EtherChannel	67
5.1.1. Link Aggregation Protocol	67
5.1.2. Layer 2 EtherChannels	68
5.1.3. Layer 3 EtherChannels	68
5.1.4. EtherChannel Modes	69
5.1.5. EtherChannel Misconfiguration Guard	70
5.1.6. Vlan internal allocation policy	70
5.2. LACP	71
5.2.1. Restrictions	71
5.2.2. Modes	71

5.2.3. LACP Hot-Standby Ports	71
5.2.4. LACP Port-Channel MaxBundle Feature	72
5.2.5. LACP Port-Channel Min-Links Feature	72
5.3. PAgP	72
5.3.1. Modes	73
5.3.2. Physical Vs Aggregate Learners	73
5.3.3. Priority	74
5.3.4. Restrictions	74
5.3.5. Silent Mode	75
5.4. Virtual Switch System	75
5.4.1. VSS Active and Standby Switch	76
5.4.2. Virtual Switch Link	76
5.4.3. Multi-chassis Ethernet Channel	79
5.4.4. vPC	79
5.4.5. Stackwise	79
6. SPAN , RSPAN and ERSPAN	81
6.1. Local SPAN Sessions	81
6.1.1. Source	82
6.1.2. Destination Port	83
6.1.3. VLAN Filtering	84
6.2. Remote SPAN Sessions	84
6.2.1. RSPAN VLAN	85
6.2.2. RSPAN Source Session	85
6.2.3. RSPAN Destination Session	85
6.3. Encapsulated RSPAN	86
6.3.1. ERSPAN Source Session	86
6.3.2. ERSPAN Destination Session	87
6.3.3. ESPAN Dummy MAC Address Rewrite	87
6.4. Interaction with Other Features	87
7. Port-Based Traffic Control	90
7.1. Storm Control	90
7.2. Protocol Storm Protection	91
7.3. Protected Port	91
7.4. Port Blocking	91
7.5. Port Security	91
8. WAN	92
8.1. HDLC	92
8.1.1. HDLC Frame Format	92
8.1.2. Encapsulation	92
8.1.3. Clock Rate	92
8.2. PPP	93

8.2.1. PPP Frame Format	93
8.2.2. PPP LCP	93
8.2.3. Multilink PPP	94
8.2.4. PPP Compression	96
8.2.5. PPP Authentication	96
8.2.6. CHAP	97
8.3. PPPoE	98
8.3.1. PPPoE packets	98
8.3.2. PPPoE Server	99
8.3.3. PPPOE Client	99
8.3.4. PPPoE Authentication	101
8.4. Ethernet WAN	101
8.4.1. VPLS	101
8.4.2. Metro-Ethernet	102
8.4.3. Ethernet Private Line (EPL)	102
8.4.4. Ethernet Virtual Private Line (EVPL)	102
Part II : Layer 3 Technologies	104
9. IPv4	105
9.1. IP Packet Format	105
9.2. IP Address	106
9.3. CIDR	107
9.4. Private Addressing	107
9.5. VLSM	107
9.6. Subnet Zero	107
9.7. Unnumbered Interfaces	107
9.8. 31-Bit Prefix	108
9.9. Checksum	108
9.9.1. Sender	108
9.9.2. Receiver	109
9.10. Protocol	109
9.11. IP Options	110
9.12. IP fragmentation and Re-assembly	111
10. ICMP	112
10.1. Header	112
10.2. Control Messages	112
10.3. ICMP Unreachable Messages	112
10.4. ICMP Source Quench Messages	114
10.5. ICMP Redirect Messages	114
10.6. ICMP Echo Request	115
10.7. ICMP Router Messages	115
10.8. ICMP Time Exceeded	115

10.9. Parameter Problem	115
10.10. Timestamp Messages	115
10.11. Information Request	115
10.12. Address Mask Messages	115
10.13. Ping	116
10.14. Traceroute	116
11. TCP	117
11.1. Connection establishment	119
11.2. TCP MSS	119
11.3. TCP Window	119
11.4. Sliding window operations	119
11.5. TCP Options	119
11.6. Ident	119
11.7. TCP Small Servers	120
12. UDP	121
12.1. Message Format	121
12.2. UDP checksum	121
12.3. UDP dominance	121
12.4. UDP Small Servers	121
13. ARP	122
13.1. Protocol	122
13.2. Static ARP Entries	122
13.3. Dynamic ARP Entries	123
13.4. Proxy ARP	123
13.5. Gratuitous ARP	124
13.6. RARP and BootP As DHCP Precursors	125
13.7. ARP vulnerabilities	125
14. DHCP	127
14.1. Protocol Operations	127
14.2. DHCP Client	129
14.2.1. Configurable DHCP Client Feature	130
14.2.2. FORCERENEW Message Handling	131
14.3. DHCP Server	131
14.3.1. Database Agent	131
14.3.2. Address Pool	132
14.3.3. Address Bindings	134
14.3.4. Static Mapping	135
14.3.5. Pings	136
14.3.6. BOOTP Interoperability	136
14.3.7. Central DHCP Server	136
14.3.8. Option 82	137

14.3.9. Statistics	138
14.4. DHCP Relay Agent	138
14.4.1. Option 82	138
14.5. Accounting and Security	140
14.5.1. DHCP Accounting	140
14.5.2. DHCP Secured IP Address Assignment	140
14.5.3. DHCP Per Interface Lease Limit and Statistics	141
14.5.4. DHCP Authorized ARP	141
14.5.5. ARP Auto-Logoff	142
14.6. DHCP Snooping	142
15. NAT	143
15.1. Purpose	143
15.2. Inside and Outside Address	143
15.3. Static NAT	143
15.4. Dynamic NAT Without PAT	144
15.5. PAT	144
15.6. NAT for Overlapping Address	145
15.7. TCP Load Distribution for NAT	145
15.8. Overlapping Networks	146
15.9. Server TCP Load Balancing	146
15.10. NAT Order Of Operations	147
15.10.1. Inside-to-Outside	147
15.10.2. Outside-to-Inside	148
16. NHRP	149
17. IPv6	151
17.1. IPv6 Header	151
17.2. Traffic Class	151
17.3. Flow Label	152
17.4. Payload Length	152
17.5. Next Header	152
17.5.1. Hop-by-hop options and destination options	153
17.5.2. Routing Extension Header	153
17.5.3. Fragment Extension Header	154
17.6. Fragmentation And Reassembly	154
17.6.1. Fragmenting	155
17.6.2. Re-Assembly	155
17.6.3. Security	155
17.7. Addressing	156
17.7.1. IPv4 Vs IPv6	156
17.7.2. Address Abbreviation Rules	156
17.7.3. Address Types	156

17.7.4. IPv6 Address Autoconfiguration	160
17.8. Basic IPv6 Functionality Protocols	161
17.8.1. Neighbor Discovery	161
17.8.2. ICMPv6	163
17.8.3. DNS	164
17.8.4. CDP	164
17.8.5. DHCPv6	164
17.8.6. Access Lists	165
17.9. IPv6 tunneling	165
17.9.1. 6in4	165
17.9.2. 6to4	165
17.9.3. ISATAP	166
17.9.4. 6RD	166
17.9.5. 6VPE	166
17.10. IPv6 Routing	167
17.10.1. Static Routes	167
17.10.2. OSPFv3	167
17.10.3. EIGRPv6	167
17.11. Ospfv3	167
17.12. Readings	168
17.12.1. IPv6 General Prefix	168
17.13. misc	168
18. CEF	169
18.1. RIB	169
18.2. FIB	169
18.3. Adjacency Table	170
18.3.1. Adjacency Discovery	170
18.3.2. Unresolved Adjacency	171
18.4. CEF Load Balancing	171
18.5. Misc	172
19. BFD	173
19.1. BFD Control Packet Format	173
19.2. BFD operating modes	176
19.3. BFD Session Parameters on the Interface	177
19.4. BFD Support for Dynamic Routing	178
19.5. BFD Support for Static Routing	179
19.6. BFD Templates for Multi-Hop	180
19.7. BFD Multihop Support for IPv4 Static Routes	180
19.7.1. BFDv4 Associated Mode	181
19.7.2. BFDv4 Unassociated Mode	181
19.8. BFD on Multiple Hops	181

19.9. BFD dampening	182
20. RIP	183
20.1. RIP Messages	183
20.2. Request Message	184
20.3. Response Message	184
20.4. Default RIP Configuration	184
20.5. Basic Configuration	185
20.6. Version	185
20.7. Authentication	185
20.8. Summarization	185
20.9. Route Updates	186
20.10. Route Filtering	186
20.11. Route Metric	187
20.12. Split Horizon	187
20.13. Interpacket Delay for RIP Updates	187
20.14. Rip Optimization Over WAN	187
20.15. Timers	187
21. EIGRP	189
21.1. EIGRP Packet Format	189
21.2. EIGRP Messages	190
21.3. Neighbors	193
21.4. EIGRP Loop Prevention Techniques	196
21.4.1. Split Horizon	196
21.5. Classic Metric	196
21.5.1. Bandwidth Metric Component	196
21.5.2. Delay Metric Component	196
21.5.3. Reliability Metric Component	197
21.5.4. Load Metric Component	197
21.5.5. MTU Metric Component	197
21.5.6. Hop Count Metric Component	197
21.5.7. Routing Metric Offset Lists	198
21.6. Wide Metric	198
21.6.1. Throughput	199
21.6.2. Latency Metric Component	199
21.6.3. Reliability	199
21.6.4. Load	199
21.6.5. MTU	199
21.6.6. Hop Count	199
21.6.7. Extended Metrics	199
21.7. Reliable Transport Protocol	200
21.8. EIGRP Autonomous System Configuration	200

21.9. EIGRP Named Configuration	201
21.9.1. Address Family Section	201
21.9.2. Per-AF-Interface Section	201
21.9.3. Per-AF-Topology Configuration Section	202
21.10. DUAL	203
21.10.1. Topology Table	204
21.10.2. Feasibility Condition	205
21.10.3. Topology Changes	206
21.10.4. Local Computation	206
21.10.5. Diffusing Computation	206
21.10.6. Multiple Topology Changes	207
21.10.7. Stuck-In-Active	209
21.11. Stub Routing	210
21.12. EIGRP Stub Routing Leak Map Support	213
21.13. Protocol-Dependent Modules	213
21.14. Goodbye Message and Graceful Shutdown	213
21.15. Summarization	213
21.15.1. Leak Map	214
21.15.2. Floating Summary Routes	215
21.15.3. Poisoned Floating Summarization	215
21.16. EIGRP Route Authentication	215
21.17. Link Bandwidth Percentage	215
21.18. EIGRP Autonomous System Configuration	215
21.19. Router ID	216
21.20. Unequal Load Balancing	216
21.21. Add-Path Support	217
21.22. Passive Interface	217
21.23. EIGRP Over the Top	217
21.23.1. LISP	218
21.23.2. OTP CE	218
21.23.3. OTP Route Reflectors	219
21.23.4. EIGRP Logging and Reporting	219
21.23.5. SoO	219
22. OSPF	220
22.1. Neighbors	220
22.2. Router Id	220
22.3. DR Election	221
22.4. Ospf Cost	221
22.5. OSPF Packet Format	221
22.5.1. Common OSPF Packet Header	221
22.5.2. Hello Packet	222

22.5.3. Database Description Packet	223
22.5.4. Link State Request	225
22.5.5. Link State Update	225
22.5.6. Link State Acknowledgment	226
22.5.7. Link-State Packets	226
22.6. Backbone	228
22.7. Stubby Areas	228
22.7.1. Stubby Area	228
22.7.2. Totally Stubby	229
22.7.3. NSSA	229
22.7.4. Totally NSSA	229
22.8. OSPF Path Selection	229
22.9. Virtual Links	230
22.10. Network Types	230
22.11. Graceful Restart	232
22.12. SPF Throttling	232
22.13. Capability Vrf-Lite	232
22.14. Summarization	232
22.15. OSPF States	232
22.16. OSPF Process	234
22.17. OSPF Authentication	235
22.17.1. Classic OSPF Authentication	235
22.17.2. Extended Cryptographic OSPF Authentication	236
22.18. TTL Security Check	237
22.19. SPF	238
22.19.1. Spf Timers	238
22.19.2. SPF Throttling	238
22.19.3. LSA Throttling	238
22.19.4. Incremental SPF	239
22.20. OSPF Filtering	239
22.20.1. Routes Filtering Not LSA Filtering	239
22.20.2. ABR Type 3 LSA Filtering	240
22.20.3. Using the Area Range No-Advertise Option	240
22.21. OSPFv2 Prefix Suppression	240
22.22. OSPF Stub Router	240
22.23. OSPF Graceful Restart	241
22.24. OSPF Graceful Shutdown	241
23. IS-IS	242
23.1. NSAP Address	242
23.2. Levels Of Routing	243
23.3. Adjacency	243

23.4. Metrics	244
23.5. Packets	244
23.5.1. Hello	244
23.5.2. Link State PDU	245
23.5.3. Complete Sequence Numbers PDU	245
23.5.4. Partial Sequence Numbers PDU	246
23.6. Network Types	246
23.6.1. Point-to-Point Links	246
23.6.2. Broadcast Networks	247
23.7. Areas	247
23.8. Authentication	248
23.9. IPv6 Support	248
24. BGP	249
24.1. BGP Message Format	249
24.1.1. BGP Header	249
24.1.2. OPEN	250
24.1.3. KEEPALIVE	251
24.1.4. UPDATE	251
24.1.5. NOTIFICATION	253
24.1.6. BGP FSM States	253
24.2. Autonomous Systems	254
24.2.1. ASN Format	254
24.3. BGP Peers	255
24.4. BGP Peer Groups	255
24.5. BGP Session Reset	256
24.5.1. Hard Reset	256
24.5.2. Soft Reset	256
24.5.3. Dynamic Inbound Soft Reset	257
24.5.4. Routing Policy Change Management	257
24.6. BGP Route Aggregation	257
24.6.1. BGP Route Aggregation Generating AS_SET Information	257
24.7. BGP Backdoor Routes	258
24.8. Best Path Selection Algorithm	258
24.9. Community Attributes	259
24.10. BGP Routing Process	259
24.10.1. Aggregating Route Prefixes Using BGP	259
24.11. BGP Routes	260
24.12. Peer Session Template	261
24.13. Peer Policy Template	261
24.14. BGP Routing Table	261
24.15. Troubleshoot	263

24.16. Todos	263
24.17. BGP PIC	263
24.18. BGP TTL Security Check	264
24.19. BGP Auto-Summary	265
25. Redistribution	266
25.1. Administrative Distance	266
25.2. Spot Issues	266
25.3. Heuristics	266
25.4. Connected Routes	267
25.5. Static Routes	267
25.6. RIP	268
25.7. EIGRP	268
25.8. OSPF Redistribution	268
25.9. BGP Redistribution	269
25.9.1. IGP to BGP	269
25.9.2. BGP to IGP	269
Part III : VPN Technologies	270
26. VRF	271
26.1. VRF definition	271
26.2. Route Distinguisher	271
26.3. Route Target	271
26.4. VRF Interface	272
26.5. VRF Static Route	272
26.6. VRF lite	272
26.7. Multi-VRF	273
27. MPLS	274
27.1. MPLS Label Stack	274
27.2. Label Distribution	275
27.3. MPLS ping and traceroute	275
27.4. L3VPNs	276
27.5. IPv6 over MPLS: 6PE and 6VPE	276
28. LDP	278
28.1. LDP process	278
28.2. Discovery Of Adjacent LDP Peers	278
28.3. LDP Sessions	279
28.4. LDP label binding, label spaces and identifiers	280
28.5. LDP Session protection	282
28.6. LDP Authentication	282
28.7. LDP MD5 Global Configuration	283
28.8. LDP Auto-configuration	283
28.9. LDP outbound label filtering	284

28.10. LDP Inbound Label Binding Filtering.....	284
28.11. LDP Graceful restart.....	284
29. GRE	285
29.1. Tunneling	285
29.2. GRE Header	285
29.3. GRE Keepalive	286
29.4. GRE Tunnel	287
29.4.1. Configuration Example.....	287
29.5. GRE backup interface	288
29.6. Troubleshooting.....	289
29.6.1. "%TUN-5-RECURDOWN" Error Message and Flapping EIGRP/OSPF/BGP Neighbors Over a GRE Tunnel	289
29.7. Questions	289
30. DMVPN	291
30.1. Phases	291
30.1.1. Phase 1	291
30.1.2. Phase 2	291
30.1.3. Phase 3	291
30.2. DMVPN with IPsec using pre-shared key.....	292
30.3. QoS profile	296
30.4. QoS Pre-classify	298
31. IPSEC	299
31.1. IPSec with pre-shared key	299
31.1.1. IPv4 site to IPv4 site	299
31.1.2. IPv6 in IPv4 tunnels.....	301
31.1.3. VTI	301
31.2. GET VPN	302
31.2.1. Group Member	302
31.2.2. Key Server	303
32. L2 VPN	304
32.1. Pseudo-Wire	304
32.2. L2TPv3	304
32.3. ATOM	305
32.4. VPLS	306
32.5. OTV	308
Part IV : Infrastructure Security	310
33. AAA	311
33.1. Local AAA Server	311
33.2. PPP Security	311
34. CoPP	312
35. CPP	313

36. Management Plane Protection	315
37. Access Control List	316
37.1. References	316
37.2. Reflexive ACL	316
38. IP Source Guard	317
39. Dynamic ARP Inspection	318
39.1. Router Security	318
40. IEEE 802.1X	319
40.1. Definition	319
40.2. Port Security	319
Part V : Infrastructure Services	320
41. System Management	321
42. Switch Administration	322
42.1. Interface Characteristics	322
42.2. System Clock	322
42.3. System Name and Prompt	322
42.4. MOTD Login Banner	323
42.5. Login Banner	323
42.6. MAC Address Table	323
42.6.1. Aging Time	324
42.6.2. MAC Address Change Notification Traps	324
42.6.3. MAC Address Move Notification Traps	324
42.6.4. MAC Threshold Notification Traps	325
42.6.5. Static Address	325
42.6.6. Unicast MAC Address Filtering	326
42.6.7. MAC Address Learning	326
42.7. Error disable	327
42.7.1. Err-Disable detection	327
42.7.2. Link-flap detection	328
42.7.3. Loopback error	328
42.7.4. L2TP guard	329
42.7.5. Incorrect GBIC / Small Form-Factor Pluggable module or cable	329
42.7.6. Err-Disable recovery	329
42.8. L2 MTU	330
42.9. Switch Internal Processing	331
42.10. Switching and Bridging Logic	331
42.11. Telnet	331
42.12. SSH	333
42.13. SCP	333
42.14. [T]FTP	334
42.14.1. FTP Client	334

42.14.2. FTP Server	334
42.15. SNMP	334
42.15.1. Version	334
42.15.2. MIB	334
42.15.3. Packet Format	335
42.15.4. Basic System Information	335
42.15.5. Views	335
42.15.6. Communities	336
42.15.7. Traps	336
42.15.8. SNMP V3	336
42.15.9. SNMP Manager	337
42.15.10. SNMP Shutdown Mechanism	337
42.16. RMON	338
42.17. Syslog	339
42.18. NTP	339
42.18.1. NTP Associations	339
42.18.2. NTP Access Groups	340
42.18.3. NTP Authentication	340
42.18.4. Source IP Address	341
42.18.5. Authoritative Server	341
42.18.6. Panic Threshold	341
42.18.7. Orphan Mode	341
42.18.8. External Reference Clock	341
42.18.9. Software Clock	342
42.18.10. Hardware-Clock	342
42.18.11. Time Ranges	342
42.18.12. Vulnerability	342
42.18.13. Example	342
42.19. HTTP	343
42.20. RTP/RTCP	344
43. QoS	345
43.1. Classification and Marking	345
43.1.1. Fields That Can Be Marked for QoS Purposes	345
43.1.2. NBAR	347
43.1.3. CB Marking	348
43.1.4. QoS Pre-Classification	348
43.1.5. AutoQoS	349
43.2. Congestion Management and Avoidance	352
43.2.1. Queues	352
43.2.2. CBWFQ	352
43.2.3. LLQ	352

43.3. Shaping, Policing and Link Fragmentation	353
43.4. MQC	353
43.4.1. Traffic Class	353
43.4.2. Elements Of a Traffic Policy	354
43.4.3. Service Policy	355
43.5. RSVP	355
43.5.1. Configuration Tasks	355
44. First Host Redundancy Protocols	356
44.1. HSRP	356
44.1.1. HSRP Packet	356
44.1.2. HSRP Version	356
44.1.3. HSRP OpCode	357
44.1.4. HSRP State	357
44.1.5. Priority	358
44.1.6. HSRP Timers	358
44.1.7. HSRP Authentication	358
44.1.8. HSRP and Object Tracking	359
44.1.9. HSRP Support for ICMP Redirects	359
44.1.10. HSRP Virtual IP Address and Group	360
44.1.11. Multiple HSRP	360
44.2. GLBP	361
44.2.1. GLBP Packet Type	361
44.2.2. Active Virtual Gateway	361
44.2.3. Active Virtual Forwarder	363
44.2.4. Authentication	364
44.3. VRRP	364
44.4. IDRPs	365
44.4.1. Message Format	365
44.4.2. Configuration	366
44.5. IPv6 RA/RS	367
45. Multicast	368
45.1. IP Multicast	368
45.1.1. Multicast IP Addressing	368
45.1.2. Mapping IP Multicast Address to MAC Addresses	369
45.2. IGMP	369
45.2.1. IGMP Packet Format	369
45.2.2. Messages	370
45.2.3. Default IGMP Configuration	371
45.2.4. IGMP Version	371
45.2.5. Querier Election	372
45.2.6. IGMPv2 Query Timeout	372

45.2.7. Maximum Response Time Field	372
45.2.8. Join the Club	372
45.2.9. Leave Process	373
45.2.10. IGMP Message Restriction	374
45.2.11. IGMP Proxy	374
45.2.12. CGMP	375
45.2.13. RGMP	376
45.2.14. IGMP Filtering and Throttling	377
45.2.15. IGMP Snooping	378
45.2.16. Multicast Router Port	379
45.2.17. MVR	381
45.2.18. IGMP Filtering and Throttling	384
45.3. PIM	385
45.3.1. Versions	385
45.3.2. Modes	386
45.3.3. PIM Designated Routers	389
45.3.4. Rendez-Vous Points	389
45.3.5. Auto-RP	390
45.3.6. Bootstrap Router	392
45.3.7. Neighbor Discovery	394
45.3.8. Auto-RP and BSR Configuration Guidelines	395
45.3.9. PIM Domain Border	396
45.3.10. Delay the Use Of PIM Shortest-Path Tree	396
45.3.11. Troubleshoot	397
45.3.12. Misc	397
45.3.13. PIM snooping	398
45.3.14. PIM stub routing	398
45.4. MLD	398
45.4.1. MLD Snooping	399
46. Network Optimization	400
46.1. IP SLA	400
46.1.1. IP SLAs Operation Types	400
46.1.2. IP SLAs Responder	400
46.1.3. IP SLAs Operation Scheduling	401
46.1.4. IP SLAs Operation Threshold Monitoring	401
46.1.5. MPLS VPN Awareness	402
46.1.6. History Statistics	402
46.1.7. Troubleshooting Tips	402
46.2. Enhanced Object Tracking	402
46.2.1. Interface Tracking	402
46.2.2. IP Route Tracking	403

46.2.3. IP SLA Operation Tracking	404
46.2.4. Tracked List	404
46.3. NetFlow	405
46.3.1. NetFlow Flows	405
46.3.2. NetFlow Version	406
46.3.3. NetFlow Cache	407
46.3.4. NetFlow Data Export	408
46.4. Embedded Event Manager	408
46.5. Embedded Packet Capture	409
46.5.1. Capture Buffer	409
46.5.2. Capture Point	411
46.5.3. Using Wireshark trace analyzer	412
46.6. Performance Monitor	412
Part VI : Evolving Technologies	413
47. Cloud	414
47.1. Compare and Contrast Cloud Deployment Models	414
47.1.1. Infrastructure, Platform, and Software Services [XaaS]	414
47.1.2. Performance and Reliability	414
47.1.3. Security and Privacy	414
47.1.4. Scalability and Interoperability	414
47.2. Describe Cloud Implementations and Operations	414
47.2.1. Automation and Orchestration	415
47.2.2. Workload Mobility	415
47.2.3. Troubleshooting and Management	415
47.2.4. OpenStack Components	415
47.2.5. Resources and References	417
48. SDN	418
48.1. Models	418
48.2. Describe Functional Elements Of Network Programmability and How They Interact	420
48.2.1. Controllers	420
48.2.2. APIs	420
48.2.3. Scripting	420
48.2.4. Agents	420
48.2.5. Northbound Vs. Southbound Protocols	421
48.3. Describe Aspects Of Virtualization and Automation In Network Environments	422
48.3.1. DevOps Methodologies, Tools and Workflows	423
48.3.2. Network/Application Function Virtualization [NFV, AFV]	423
48.3.3. Service Function Chaining	424
48.3.4. Performance, Availability, and Scaling Considerations	424
49. Internet Of Things	426
49.1. Describe Architectural Framework and Deployment Considerations for IoT	426

49.1.1. Data center (DC) Cloud	427
49.1.2. Core Networking and Services	427
49.1.3. Multi-service Edge (access network).....	427
49.1.4. Embedded Systems (Smart Things Network)	428
49.1.5. Performance, Reliability and Scalability	429
49.1.6. Mobility	429
49.1.7. Security and Privacy	430
49.1.8. Standards and Compliance	431
49.1.9. Migration	433
49.1.10. Environmental Impacts on the Network	433
Appendices	435
50. Lab Equipment and IOS Releases	436
51. Labs	437
51.1. MST	437
51.1.1. Two VLANs mapped to the same instance block the same ports	438
52. IOS	439
53. IOS-XE	440
54. IOS System Management	441
54.1. Configuration files	441
55. General	444
56. Notes To Self	445

Dedication

To Cyril "Matiere" Kalenga

Part I : Layer 2 Technologies

Chapter 1. Ethernet

- IEEE 802.3 standards
- CSMA/CD protocol
- Medium: coaxial, twisted-pair, optical fiber
- Data rates: 10/100/1000/10000 Mbps

1.1. Frame Formats

8	6	6	2	46-1500	4
Preamble	DA	SA	Type/Length	Data and Padding	FCS

Figure 1. Ethernet (DIX) and Revised (1997) IEEE 802.3

8	6	6	2	1	1	1-2	43-1500	4
Preamble	DA	SA	Length	DSAP	SSAP	Control	Data and Padding	FCS

Figure 2. Original IEEE 802.3

8	6	6	2	1	1	1-2	3	2	43-1500	4
Preamble	DA	SA	Length	DSAP	SSAP	Control	OUI	Type	Data and Padding	FCS

Figure 3. IEEE 802.3 with SNAP

Preamble DIX or Preamble and Start of Frame Delimiter(802.3)

- 62 alternating 1s and 0s, and ends with a pair of 1s.
- For clocking synchronization of the transmitted signal.

Type

- Type of protocol

Length

- Length in bytes of the data following the Length field, up to the Ethernet trailer.

DA

- Destination address can be an individual or group address

SA

- Source address is always unicast address

DSAP

- Destination Service Access Protocol
- The size limitations, along with other Point (802.2) uses of the low-order bits, required the later addition of SNAP headers.

SSAP

- Source Service Access Protocol
- Describes the upper-layer protocol Point (802.2) that created the frame.

Control

- Enables both connectionless and connection-oriented operation.
- Generally used only for connectionless operation by modern protocols, with a 1-byte value of 0x03.

SNAP OUI

- Generally unused today,
- Providing a place for the sender of the frame to code the OUI representing the manufacturer of the Ethernet NIC.

SNAP Type

- Using same values as the DIX Type field, overcoming deficiencies with size and use of the DSAP field.

Data

- N bytes where $46 \leq N \leq 1500$
- If $N < 46$, use padding

FCS (Frame check sequence)

- Contains a 32-bit cyclic redundancy check (CRC) value
- Calculated by the sending MAC
- Re-calculated by the receiving MAC to check for damaged frames.
- Generated from the DA, SA, Length/Type, and Data fields

1.2. Ethernet MAC Addresses

- 48 bits in hexadecimal
- Canonical transmission (little endian)= MSO to LSO with LSB to MSB for each octet where
 - I/G bit: (0/1) Individual or Group address, first bit to be transmitted as LSB of MSO
 - U/L bit: (0/1) Universally or Locally administrated, second bit to be transmitted

Example: AC-10-7B-3A-92-3C

Convert to Hexa : 10101100 00010000 01101011 00111010 01010010 00111100
Transmission : 00110101 00001000 11010110 01011100 01001010 00111100

Task: Change the MAC Address

```
(config-if)# mac-address AC-10-BE-EF-DE-AD
```



Even if you change the MAC address of the switch port, STP will continue to use the BIA.

1.2.1. Types Of MAC Addresses

- Unicast : I/G bit = 0
- Multicast: I/G bit = 1
- Broadcast: all devices in the segment

1.3. RJ-45 Pinouts and Cat5 Wiring

- Defined by [EIA / TIA](#)

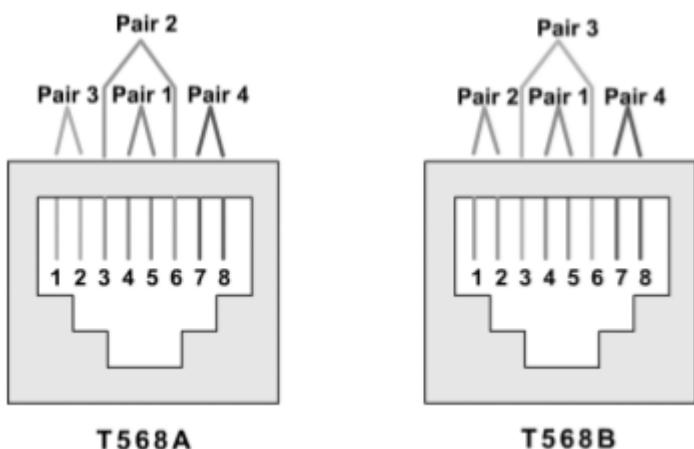


Table 1. Ethernet Cabling Types

Type of cable	Pinouts	Key pins connected
Straight-through	T568A or T568B both ends	1-1; 2-2; 3-3; 6-6
Cross-over	T568A on one end; T568B on the other	1-3; 2-6; 3-1; 6-2

- Auto-MDIX (automatic medium-dependent interface crossover)
 - Detects the wrong cable and causes the switch to swap the pair it uses for transmitting and receiving
 - Not supported on all Cisco switch models

Table 2. UTP Cabling References

UTP	Speed	Description
1	—	Used for telephones and not for data
2	4 Mbps	Originally intended to support Token Ring over UTP

UT P	Speed	Description
3	10 Mbps	popular option for Ethernet in years past, if Cat 3 cabling for phones was already in place
4	16 Mbps	Intended for the fast Token Ring speed option;
5	1 Gbps	Very popular for cabling to the desktop
5e	1 Gbps	Added mainly for the support of copper cabling for Gigabit Ethernet
6	1 Gbps+	Cat5e replacement, with multi-gigabit support

1.4. Auto-Negotiation, Speed and Duplex

- By default, Ethernet auto-negotiation uses FLP (Fast Link Pulses) to determine the speed and duplex setting.
- To disable auto-negotiation, manually configure the speed and the duplex settings.
- if auto-negotiation is disabled on one end by statically setting the speed , the other end
 - detects the speed based on the incoming electrical signal
 - sets duplex to half for 10 and 100 Mbps and full duplex for 1Gps interfaces
- if auto-negotiation is disabled on both end and different speeds statically configured, link down

Task: Set Speed for the Interface

```
(config-if)# speed {10 | 100 | 1000 | auto | nonegotiate}
```

Task: Set Duplex Mode for the Interface

```
(config-if)# duplex {auto | full | half}
```

Task: Show Controllers

```
Router# show controllers fastethernet1
!
Interface FastEthernet1    MARVELL 88E6052
Link is DOWN
Port is undergoing Negotiation or Link down
Speed :Not set, Duplex :Not set
!
Switch PHY Registers:
~~~~~
00 : 3100  01 : 7849  02 : 0141  03 : 0C1F  04 : 01E1
05 : 0000  06 : 0004  07 : 2001  08 : 0000  16 : 0130
17 : 0002  18 : 0000  19 : 0040  20 : 0000  21 : 0000
!
Switch Port Registers:
~~~~~
Port Status Register      [00] : 0800
Switch Identifier Register [03] : 0520
Port Control Register     [04] : 007F
Rx Counter Register       [16] : 000A
Tx Counter Register       [17] : 0008
```

1.5. Standards

802.1Q : 802.1d : STP
dot1q
trunking

802.1s : 802.1w : Rapid STP
MST

802.1ax : 802.2 : Logical Link Control
LACP
(formerl
y
802.3ad)

802.3u : 802.3z : Gigabit ethernet over optical cable
Fast
ethernet
over
copper
and
optical
cable

Table 3. Ethernet Types and Cabling Standards

Standard	Cabling	Maximum Single Cable Length
10BASE5	Thick coaxial	500 m

Standard	Cabling	Maximum Single Cable Length
10BASE2	Thin coaxial	185 m
10BASE-T	UTP Cat 3, 4, 5, 5e, 6	100 m
100BASE-FX	Two strands, multimode	400 m
100BASE-T	UTP Cat 3, 4, 5, 5e, 6, 2 pair	100 m
100BASE-T4	UTP Cat 3, 4, 5, 5e, 6, 4 pair	100 m
100BASE-TX	UTP Cat 3, 4, 5, 5e, 6, or STP, 2 pair	100 m
1000BASE-LX	Long-wavelength laser, MM or SM fiber	10 km (SM) 3 km (MM)
1000BASE-SX	Short-wavelength laser, MM fiber	220 m with 62.5-micron fiber; 550 m with 50-micron fiber
1000BASE-ZX	Extended wavelength, SM fiber	100 km
1000BASE-CS	STP, 2 pair	25 m 100 m
1000BASE-T	UTP Cat 5, 5e, 6, 4 pair	100 m

1.6. EtherType

Protocol	EtherType
ARP	0x806
IP	0x800
IPv6	0x86DD
MPLS (Unicast)	0x8847
MPLS (Multicast)	0x8848
PPPoE (Discovery Stage)	0x8863
PPPoE (PPP Session Stage)	0x8864
RARP	0x8035

1.7. Troubleshooting

Runts

- Runts are frames smaller than 64 bytes.

Overruns

- The number of times the receiver hardware was unable to hand received data to a hardware buffer.
- Common Cause: The input rate of traffic exceeded the ability of the receiver to handle the data.

Ignores

- The number of received packets ignored by the interface because the interface hardware ran low on internal buffers.

- Common Causes: Broadcast storms and bursts of noise can cause the ignored count to be increased.

CRC errors

- The frame's cyclic redundancy checksum value does not match the one calculated by the switch or router.

Frames

- Frame errors have a CRC error and contain a noninteger number of octets.

Alignment

- Alignment errors have a CRC error and an odd number of octets.

Collisions

- Look for collisions on a full-duplex interface (meaning that the interface operated in half-duplex mode at some point in the past), or excessive collisions on a half-duplex interface.
- Excessive collisions occur when a frame is dropped because the switch encounters 16 collisions in a row.

Late collisions on a half-duplex interface

- A late collision occurs after the first 64 bytes of a frame.
- Late collisions occur after every device on the wire should have recognized that the wire was in use.

Possible causes of collisions include:

- A cable that is out of specification (either too long, the wrong type, or defective)
- A bad network interface card (NIC) card (with physical problems or driver problems)
- A port duplex misconfiguration
- A port duplex misconfiguration is a common cause of the errors because of failures to negotiate the speed and duplex properly between two directly connected devices (for example, a NIC that connects to a switch). Only half-duplex connections should ever have collisions in a LAN. Because of the carrier sense multiple access (CSMA) nature of Ethernet, collisions are normal for half duplex, as long as the collisions do not exceed a small percentage of traffic.

1.7.1. Problems and Approaches

Problem	Questions?	Commands
Lack of reachability to devices in same VLAN	<ul style="list-style-type: none"> • Layer 1 issues ? • VLAN exists on the switch? • Interface assigned to the correct VLAN? • VLAN allowed on the trunk? 	<ul style="list-style-type: none"> • show interface • show vlan • show interface switchport • traceroute mac source-mac destination-mac • show interface trunk
Intermittent reachability to devices in same VLAN	<ul style="list-style-type: none"> • Excessive interface traffic? • Unidirectional links? • Spanning-tree problems such as BPDU floods or flapping MAC addresses? 	<ul style="list-style-type: none"> • show interface • show spanning-tree • show spanning-tree root • show mac address-table
No connectivity between switches	<ul style="list-style-type: none"> • Trunk links active? • EtherChannels active? • BPDU Guard is not enabled on a trunk interface? 	<ul style="list-style-type: none"> • show interfaces status err-disabled • show interfaces trunk • show etherchannel summary • show spanning-tree detail
Poor performance across a link	<ul style="list-style-type: none"> • Duplex mismatch? 	<ul style="list-style-type: none"> • show interface

Misc: how does Ethernet know how long a frame is ?

TODO

Answer1 The Physical Coding Sublayer is responsible for delimiting the frames, and sending them up to the MAC layer.

In Gigabit Ethernet, for example, the 8B/10B encoding scheme uses a 10 bit codegroup to encode an 8-bit byte. The extra two bits tell whether a byte is control information or data. Control information can be Configuration, Start_of_packet, End_of_packet, IDLE, Carrier_extend, Error_propagation.

That is how a NIC knows where a frame start and ends. This also means that the length of the frame is not known before it has fully decoded, analogous to a NULL-terminated string in C.

Answer2 The length field inside the frame is not needed for layer1.

Layer1 uses other means to detect the end of a frame which vary depending on the type of physical layer.

with 10Base-T a frame is followed by a TP_IDL waveform. The lack of further Manchester coded data bits can be detected. with 100Base-T a frame is ended with an End of Stream Delimiter bit pattern that may not occur in payload data (because of its 4B/5B encoding). A rough description you can find e.g. here: <http://ww1.microchip.com/downloads/en/AppNotes/01120a.pdf> "Ethernet Theory of Operation"

Chapter 2. CDP, LLDP

2.1. CDP

Catalyst3560-X Configuration Guides › CDP

- Layer 2 discovery protocol running on Cisco devices
- Retrieves device type and SNMP agent address of neighboring devices
- Send CDP announcements to the multicast destination address 01-00-0c-cc-cc-cc,
- All CDP packet includes the VLAN ID of the access port or the lowest VLAN ID in a trunk port

2.1.1. Packet Format

- Header followed by a set of TLV value

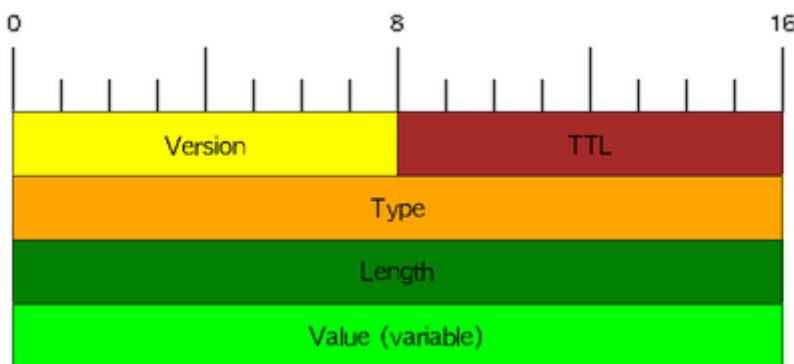


Figure 4. CDP Frame Format

2.1.2. CDP Operations

- Enabled by default

Task: Display Global Information About CDP Characteristics

```
# show cdp
```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
Router3	Ser 1	120	R	2500	Ser 0
Router1	Eth 1	180	R	2500	Eth 0
Switch1	Eth 0	240	S	1900	2

```
show cdp entry <entry-name> [protocol | version]
```

Task: Disable CDP

```
(config)# no cdp run
```

Task: Enable CDP on an Interface

```
(config-if)# cdp enable
```

2.1.3. CDP Updates

- by default every 60 seconds
- can be set between 5 to 254 seconds

Task: Set the Transmission Frequency Of CDP Updates In Seconds

```
(config)# cdp timer <seconds>
```

Task: Specify the Amount Of Time a Receiving Device Should Hold the Information Sent by Your Device

```
(config)# cdp holdtime <seconds>
```



default: 180 seconds, range: 10 to 255 seconds

2.1.4. Version

Task: Send Version-2 Advertisements

```
(config)# cdp advertise-v2
```

2.1.5. Monitoring and Maintenance

Task: Reset the Traffic Counters to Zero

```
# clear cdp counters
```

2.1.6. Neighbors

Task: Delete the CDP Table Of Information About Neighbors

```
# clear cdp table
```

Task: Display Information About Interfaces Where CDP Enabled

```
sh cdp interface [<interface-id>]
```

Task: Display Information About Neighbors

```
# sh cdp neighbors [<interface-id>] [detail]
```

Task: Display CDP Counters, Including the Number Of Packets Sent and Received and Checksum Errors

```
# show cdp traffic
```

```
Total packets output: 543, Input: 333
Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
No memory: 0, Invalid: 0, Fragmented: 0
CDP version 1 advertisements output: 191, Input: 187
CDP version 2 advertisements output: 352, Input: 146
```

2.2. LLDP

[Catalyst Configuration Guides](#) › [LLDP](#)

- IEEE 802.1AB link layer discovery protocol
- Neighbor discovery protocol
- Advertises TLV(type, length, value) for each attribute
 - Basic mandatory
 - port description
 - system name
 - system description
 - system capabilities
 - management address
 - Optional
 - port vlan ID for ieee 802.1
 - MAC/PHY configuration/status for ieee 802.3

2.2.1. LLDP Global State

- Disabled by default

Task: Enable LLDP Globally on the Switch

```
(config)# lldp run
```

Task: Display Global Information, Such As Frequency Of Transmissions, the Holdtime for Packets Being Sent, and the Delay Time Before LLDP Initializes on an Interface.

```
# show lldp
```

Task: Display LLDP Counters, Including the Number Of Packets Sent and Received, Number Of Packets Discarded, and Number Of Unrecognized TLVs.

```
# show lldp traffic
```

Task: Reset the Traffic Counters to Zero.

```
# clear lldp counters
```

Task: Delete the LLDP Neighbor Information Table.

```
# clear lldp table
```

Task: Clear the NMSP Statistic Counters.

```
# clear nmsp statistics
```

2.2.2. LLDP Interfaces

- Disabled by default

Task: Enable an Interface to Send LLDP Packets

```
(config-if)# lldp transmit
```

Task: Enable an Interface to Receive LLDP Packets

```
(config-if)# lldp receive
```

Task: Display Information About Interfaces with LLDP Enabled.

```
# show lldp interface [<interface-id>]
```

2.2.3. Neighbors

Task: Display Information About a Specific Neighbor.

```
# show lldp entry <entry-name>
```

Task: Display Information About All Neighbors.

```
# show lldp entry *
```

Task: Display Information About Neighbors, Including Device Type, Interface Type and Number, Holdtime Settings, Capabilities, and Port ID.

```
# show lldp neighbors [<interface-id>] [detail]
```

2.2.4. Timers

Task: Specify the Amount Of Time a Receiving Device Should Hold the Information from Your Device

- default: 120 s, range: 0 - 65535

```
(config)# lldp holdtime <seconds>
```

Task: Specify the Delay Time In Seconds for LLDP to Initialize on an Interface.

The range is 2 to 5 seconds; the default is 2 seconds.

```
(config)# lldp reinit delay
```

Task: Set the Sending Frequency Of LLDP Updates In Seconds.

The range is 5 to 65534 seconds; the default is 30 seconds.

```
(config)# lldp timer rate
```

2.2.5. TLV

Task: Specify the LLDP TLVs to Send or Receive.

```
(config)# lldp tlv-select
```

Task: Specify the LLDP-MED TLVs to Send or Receive.

```
(config)# lldp med-tlv-select
```

Task: Specify the LLDP-MED TLV to Send

```
(config-if)# lldp med-tlv-select {inventory-management | location | network-policy | power-management }
```

Task: Configure Network Policy TLV

```
(config)# network-policy profile <profile-number>
(config)# {voice | voice-signaling} vlan [<id> {cos <cvalue> | dscp <dvalue>}]
    | [[dot1p {cos <cvalue> | dscp <dvalue>}]] | none | untagged]
(config-if)# network-policy <profile-number>
(config-if)# lldp med-tlv select network-policy
```



- if the interface is configured as a tunnel port, LLDP is automatically disabled.
- If you first configure a network-policy profile on an interface, you cannot apply the switchport voice vlan command on the interface. If the switchport voice vlan vlan-id is already configured on an interface, you can apply a network-policy profile on the interface. This way the interface has the voice or voice-signaling VLAN network-policy profile applied on the interface.
- You cannot configure static secure MAC addresses on an interface that has a network-policy profile.
- You cannot configure a network-policy profile on a private-VLAN port.
- For wired location to function, you must first enter the ip device tracking global configuration command.

Task: Display the Location Information for an Endpoint.

```
# show location
```

2.2.6. Network-Policy Profiles

Task: Display the Configured Network-Policy Profiles.

```
# show network-policy profile
```

Task: Display the NMSP Information.

```
# show nmsp
```

2.2.7. LLDP-MED

- LLDP for Media Endpoint Devices
- operates between endpoint devices (ip phones) and network devices (switches)
- supports VoIP applications
- TLVs enabled by default:
 - LLDP-MED capabilities TLV
 - network policy TLV

- Power management TLV
- Inventory management TLV
- Location TLV

2.2.8. Wired Location Service

- The switch uses the wired location service feature to send location and attachment tracking information for its connected devices to a Cisco Mobility Services Engine (MSE). The tracked device can be a wireless endpoint, a wired endpoint, or a wired switch or controller. The switch notifies the MSE of device link up and link down events through the Network Mobility Services Protocol (NMSP) location and attachment notifications.

The MSE starts the NMSP connection to the switch, which opens a server port. When the MSE connects to the switch there are a set of message exchanges to establish version compatibility and service exchange information followed by location information synchronization. After connection, the switch periodically sends location and attachment notifications to the MSE. Any link up or link down events detected during an interval are aggregated and sent at the end of the interval.

When the switch determines the presence or absence of a device on a link-up or link-down event, it obtains the client-specific information such as the MAC address, IP address, and username. If the client is LLDP-MED- or CDP-capable, the switch obtains the serial number and UDI through the LLDP-MED location TLV or CDP.

Depending on the device capabilities, the switch obtains this client information at link up:

- Slot and port specified in port connection
- MAC address specified in the client MAC address
- IP address specified in port connection
- 802.1X username if applicable
- Device category is specified as a wired station
- State is specified as new
- Serial number, UDI
- Model number
- Time in seconds since the switch detected the association

Depending on the device capabilities, the switch obtains this client information at link down:

- Slot and port that was disconnected
- MAC address
- IP address
- 802.1X username if applicable
- Device category is specified as a wired station
- State is specified as delete

- Serial number, UDI
- Time in seconds since the switch detected the disassociation

When the switch shuts down, it sends an attachment notification with the state delete and the IP address before closing the NMSP connection to the MSE. The MSE interprets this notification as disassociation for all the wired clients associated with the switch.

If you change a location address on the switch, the switch sends an NMSP location notification message that identifies the affected ports and the changed address information.

Chapter 3. VLANs and Trunking

3.1. Normal and Extended VLANs

- Administratively defined subset of switch ports that are in the same broadcast domain
- Best practice: one VLAN, one IP subnet
- Traffic inside same VLAN is layer 2 switched
- Traffic between VLANs is layer 3 routed
- Can span multiple physical switches over "trunks"

3.1.1. VLAN Numbering

- VLAN ID = 12 bits

Reserved [0, 4095]

- Not available for use

Normal-range [1-1005]

- Advertised and pruned by VTP v1 and v2 except vlan 1, 1002-1005
- Configured in both vlan database mode and configuration mode
- Stored in VLAN.DAT file in Flash
- Special VLANs:
 - Vlan 1 is the default Ethernet VLAN for all access ports; cannot be deleted or changed.
 - Vlan 1002,1004 : default for FDDI (default, net)
 - Vlan 1003,1005 : default for Token Ring (default,translational bridge).

Extended-range [1006-4094]

- Cannot be advertised or pruned by VTP v1 and v2
- Configured only in VTP transparent mode
- Stored only in the running configuration

3.1.2. VLAN Trunks

- Trunk: point-to-point links for multiple VLANs between devices
- Trunking add ISL or 802.1q headers to include VLAN id.
 - ISL : Cisco proprietary, 30-bytes (26-byte header + 4-byte trailer), does not modify original frame
 - 802.1q: IEEE standard, 4-byte tag except for native VLAN, modifies original frame

3.1.3. Basic Configuration

Configuring VLANs requires few steps:

1. Create the VLAN Itself
2. Associate the Correct Ports with VLAN

VLAN creation can be done either in VLAN database mode configuration (after **vlan database**) or normal configuration mode

Table 4. Catalyst 3550 VLAN Database Vs Configuration Mode

VLAN Database	Configuration
vtp {domain domain-name password password pruning v2-mode {server client transparent}}	vtp {domain domain-name file filename interface name mode {client server transparent} password password pruning version number}
vlan vlan-id [backupcrf {enable disable}] [mtu mtu-size] [name vlan-name] [parent parent-vlan-id] [state {suspend active}]	vlan vlan-id 1
show {current proposed difference}	No equivalent
apply abort reset	No equivalent

3.1.4. VLAN State

- Can be active or suspended

Task: Modify the Operational State Of a VLAN

```
(config)# vlan <id>
(config-vlan)#state [active | suspend]
```

3.1.5. Troubleshoot

Check "Creating ethernet VLANs on catalyst switches: troubleshoot tips"

- SVI will be in "up/down" state after being deleted
- SVI will be in "up/up" if
 - The VLAN associated with the SVI exists in the VLAN database
 - At least one trunk or access port in the "up/up" state has been assigned to the VLAN
 - Those ports in the "up/up" state are not blocked by STP

3.2. Voice VLANs

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swvoip.html

3.3. Private VLANs

Cat3560-X Configuration Guides > [Private VLANs](#)

- Separate ports as if they are on different VLAN while consuming only one subset.
- Typically used by service provider in a multi-tenant offerings: one router, one switch, multiple customers
- PVLAN
 - one primary VLAN (promiscuous ports) + multiple secondary VLANs
 - At most one isolated secondary VLAN
 - Zero or more community secondary VLANs
- Cannot use vlan 1, 1002-1005 as private vlans

Task: Display Private VLAN Information

```
# sh vlan private-vlan [type]
```

Primary	Secondary	Type	Ports
10	501	isolated	Gi2/0/1, Gi3/0/1, Gi3/0/2
10	502	community	Gi2/0/11, Gi3/0/1, Gi3/0/4
10	503	non-operational	

3.3.1. Primary VLANs

- carries unidirectional traffic downstream from promiscuous ports to all ports

Task: Configure Primary VLAN and Associated Secondary VLANs

```
(config)# vtp mode transparent
(config)# vlan 100
(config-vlan)#private-vlan primary
(config-vlan)#private-vlan association 123-321,999
```

3.3.2. Isolated VLANs

- carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.

Task: Configure Isolated VLANs

```
(config)# vtp mode transparent
(config)# vlan 102
(config-vlan)# private-vlan isolated
```

3.3.3. Community VLANs

- carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.

Task: Configure Isolated VLANs

```
(config)# vtp mode transparent  
(config)# vlan 102  
(config-vlan)# private-vlan isolated
```

3.3.4. Private-Vlan Host Port

Task: Configure a Layer 2 Port Interface As a Private-Vlan Host Port

```
(config-if)# switchport mode private-vlan host  
(config-if)# switchport private-vlan host-association <primary-vlan-id> <secondary-vlan-ids>
```



Although private VLANs provide host isolation at Layer 2, hosts can communicate with each other at Layer 3.

3.3.5. Private-VLAN Promiscuous Ports

Task: Configure a Layer 2 Port Interface As a Private-Vlan Promiscuous Port

```
(config-if)# switchport mode private-vlan promiscuous  
(config-if)# switchport private-vlan mapping <primary-vlan-id> [add | remove]  
<secondary-vlan-ids>
```

3.3.6. Private-VLAN SVI

- SVI can only be associated to primary VLANs
- SVIs for secondary VLANs are inactive
- If you assign an IP subnet to the primary VLAN SVI, this subnet is the IP subnet address of the entire private VLAN

Task: Configure a Layer 3 Vlan Interface As a Private-Vlan Promiscuous Port

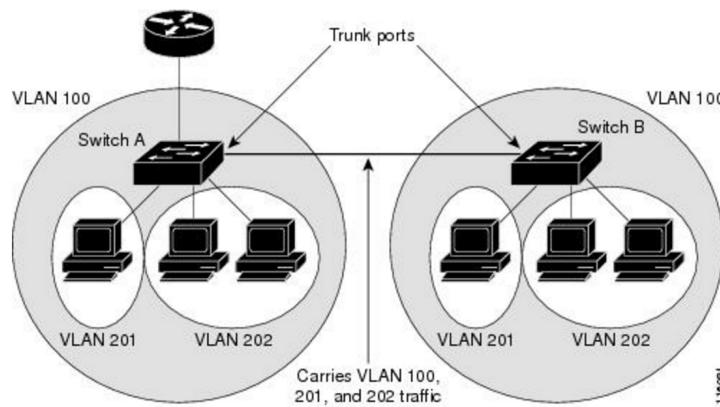
```
(config)# interface vlan <primary-vlan-id>  
(config-if)# private-vlan mapping <primary-vlan-id> [add | remove] <secondary-vlan-ids>
```

Task: Display Information About the Private-VLAN Mapping for VLAN SVIs

```
# sh interface private-vlan mapping
```

TODO ip sticky arp

3.3.7. Private-Vlan Accross Multiple Switches



VLAN 100 = Primary VLAN
VLAN 201 = Secondary isolated VLAN
VLAN 202 = Secondary community VLAN

As with regular VLANs, private VLANs can span multiple switches. A trunk port carries the primary VLAN and secondary VLANs to a neighboring switch. The trunk port treats the private VLAN as any other VLAN. A feature of private VLANs across multiple switches is that traffic from an isolated port in switch A does not reach an isolated port on Switch B.



- Manually configure private VLANs on all switches because VTP (v1 and v2) does not support private VLANs, -

TODO interaction with switch that do not support private-vlan

TODO PVLAN Trunk

TODO PVLAN Isolated

TODO see page 67 Narbick

3.3.8. Interaction with Other Features

VTP

- VTP v1 and v2 don't propagate private-vlans
 - Set transparent mode
 - Save the VTP transparent mode and private-vlan to startup configuration
- VTP v3 supports private-vlans

STP

- only one STP instance for the entire private-vlan
- the STP parameters of the primary VLAN are propagated to the secondary VLANs
- Enable Port Fast and BPDU guard on isolated and community host ports to prevent STP loops due to misconfigurations and to speed up STP convergence

- Do not enable Port Fast and BPDU guard on promiscuous ports.

DHCP snooping

- Can be enabled on the private VLAN
- propagates to all secondary vlans when enabled on the primary VLAN
- If you configure DHCP on a secondary VLAN, the configuration does not take effect if the primary VLAN is already configured (?!)

IP source guard

- enabled only if DHCP snooping is enabled on the primary vlan

SPAN

- You can configure a private-VLAN port as a SPAN source port.
- You can use VLAN-based SPAN (VSPAN) on primary, isolated, and community VLANs or use SPAN on only one VLAN to separately monitor egress or ingress traffic.
- A private-VLAN host or promiscuous port cannot be a SPAN destination port. If you configure a SPAN destination port as a private-VLAN port, the port becomes inactive.
- A RSPAN vlan can not be a private-vlan primary or secondary vlan.

PAgP or LACP

- If a port is part of a private vlan, any Etherchannel configuration is inactive

IGMP snooping

- When enabled (the default), the switch supports no more than 20 private-vlan domain

802.1x

- You can configure IEEE 802.1x port-based authentication on a private-VLAN port,
- You can not configure IEEE 802.1x with port security, voice VLAN, or per-user ACL on private-VLAN ports.

Static MAC address

- If you configure a static MAC address on a promiscuous port in the primary VLAN, you must add the same static address to all associated secondary VLANs.
- If you configure a static MAC address on a host port in a secondary VLAN, you must add the same static MAC address to the associated primary VLAN.
- When you delete a static MAC address from a private-VLAN port, you must remove all instances of the configured MAC address from the private VLAN.

3.3.9. PVLAN Edge or Protected Ports

- only local significance to the switch (unlike Private Vlans),
- no isolation provided between two protected ports located on different switches.
- A protected port does not forward any traffic (unicast, multicast, or broadcast) to any other port that is also a protected port in the same switch.

- Traffic cannot be forwarded between protected ports at L2, all traffic passing between protected ports must be forwarded through a Layer 3 device.

Task: Configure a protected port

```
(config-if)# switchport protected
```

3.4. VTP

[Catalyst Configuration guides](#) › [Configuring VTP](#)

- Vlan Trunk Protocol
- Cisco-proprietary that distributes VLAN information among Catalyst switches
- Advertises the VLAN ID, Name and Type but not which ports should be in each VLAN
- Eases administrative burden for addition, deletion and renaming of VLANs
- Supports 1005 VLANs (IP base or IP services feature set) or 255 VLANs (LAN base feature set)

Task: Show VTP Status

```
# show vtp status

VTP Version: 3 (capable)
Configuration Revision: 1
Maximum VLANs supported locally: 1005
Number of existing VLANs: 37
VTP Operating Mode: Server
VTP Domain Name: [smartports]
VTP Pruning Mode: Disabled
VTP V2 Mode: Enabled
VTP Traps Generation: Disabled
MD5 digest : 0x26 0xEE 0x0D 0x84 0x73 0x0E 0x1B 0x69
Configuration last modified by 172.20.52.19 at 7-25-08 14:33:43
Local updater ID is 172.20.52.19 on interface Gi5/2 (first layer3 interface fou)
VTP version running: 2
```

3.4.1. VTP Version

V1

- Default: version 1
- supports normal range only

V2

- supports for Token Ring Concentrator Relay Function and Bridge Relay Function
- propagates unknown TLV records

- Optimized VLAN database consistency checking:
 - In VTPv1, VLAN database consistency checks are performed whenever the VLAN database is modified, either through CLI, SNMP, or VTP.
 - In VTPv2, these consistency checks are skipped if the change was caused by a received VTP message, as the message itself was originated as a result of a CLI or SNMP action that must already have been sanitized.
 - This is really just an implementation optimization.

V3

- Supports the whole IEEE 802.1q vlan range up to 4095 (v1 and v2 support only normal range VLANs 1-1005)
- Can send PVLAN information in addition to normal VLAN information.
- Backward compatible with VTP 2
- Add support for databases other than VLAN databases such as MST databases.
- Clear text or hidden password protection
 - The encrypted VTP password cannot be displayed back as plaintext.
 - While this encrypted string can be carried over to a different switch to make it a valid member of the domain, the promotion of a secondary server into the primary server role will require entering the password in its plaintext form.
- Supports the **off** mode in which the switch does not participate in VTPv3 operations and drops all received VTP messages:
- Can deactivate VTP on a per-trunk basis
- VTPv3 is a generalized mechanism for distributing contents of an arbitrary database, and is not limited to synchronizing VLAN information over a set of switches:
 - For example, VTPv3 is capable of distributing and synchronizing the MST region configuration among all switches in a VTP domain

For more information, read [VTP version 3](#)

3.4.2. VTP Message Format

- Encapsulated in ISL or 802.1q frames
- Multicasted to MAC address: 0100-0CCC-CCCC, LLC code: SNAP (AAAA), Type 2003 in the SNAP Header
- Carried through trunk ports and VLAN 1

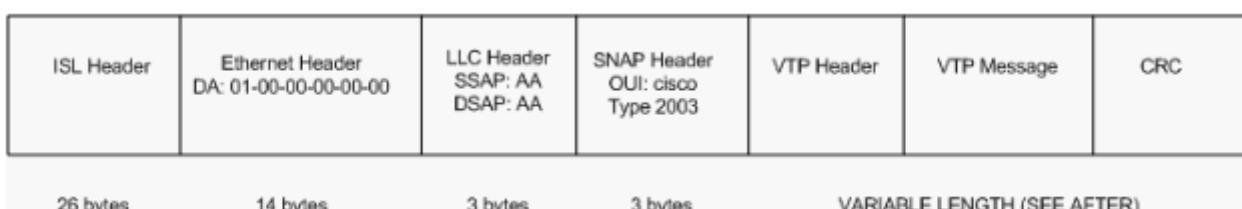


Figure 5. Example: VTP Packet Encapsulated In ISL Frame

- The VTP header contains these fields:

- VTP protocol version: 1,2,3
- VTP message types: summary advertisements, subset advertisements, advertisement requests, VTP join messages
- management domain length
- management domain name

Summary Advertisements

- Sent by Server and Client every 5 minutes intervals, and in addition, after each modification of the VLAN database
- carries information about VTP domain name, revision number, identity of the last updater, time stamp of the last update, MD5 sum computed over the contents of the VLAN database and the VTP password (if configured), and the number of Subset Advertisement messages that optionally follow this Summary Advertisement.
- summary messages do not carry VLAN database contents.
- When the switch receives a summary advertisement packet,
 - The switch compares the VTP domain name to its own VTP domain name.
 - If the name is different, the switch simply ignores the packet.
 - If the name is the same, the switch then compares the configuration revision to its own revision.
 - If its own configuration revision is higher or equal, the packet is ignored.
 - If it is lower, an advertisement request is sent.

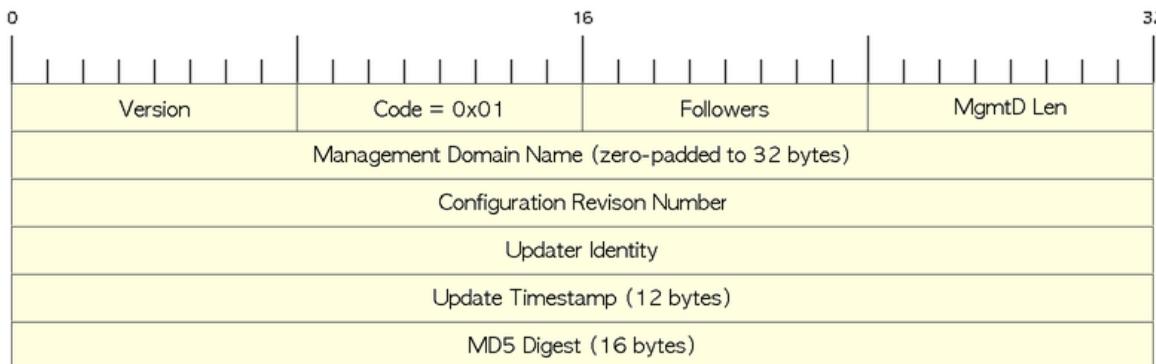


Figure 6. VTP Summary Advertisement

Followers

Indicates that this packet is followed by a Subset Advertisement packet.

Updater Identity

IP address of the switch that is the last to have incremented the configuration revision.

Update Timestamp

Date and time of the last increment of the configuration revision.

MD5 Digest

If MD5 is configured and used to authenticate the validation of a VTP update.

Subset Advertisements

- Follows the summary advertisement after addition, deletion or modification of a VLAN.
- Contains a list of VLAN information.

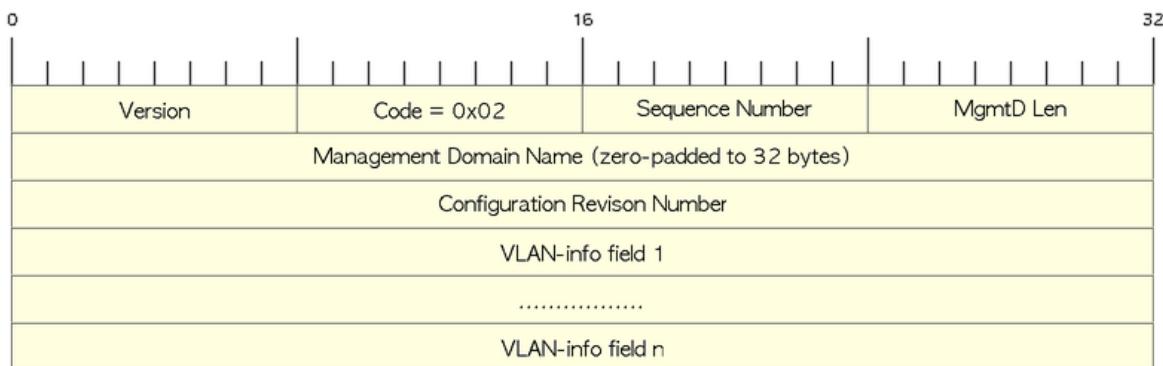


Figure 7. VTP Subset Advertisements

Sequence number

- Identify the packet in the stream of packets that follow a summary advertisement
- Starts with value 1

Advertisement Request

A switch needs a VTP advertisement request in these situations:

- The switch has been reset.
- The VTP domain name has been changed.
- The switch has received a VTP summary advertisement with a higher configuration revision than its own.

Upon receipt of an advertisement request, a VTP device sends a summary advertisement. One or more subset advertisements follow the summary advertisement. This is an example:

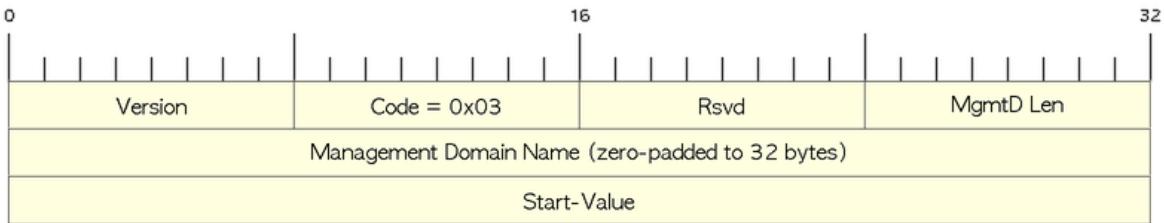


Figure 8. VTP Advertisement Request

Start-Value

This is used in cases in which there are several subset advertisements. If the first (n) subset advertisement has been received and the subsequent one (n+1) has not been received, the Catalyst only requests advertisements from the (n+1)th one.

Join Message

- originated by each VTP Server and Client switch periodically every 6 seconds if VTP Pruning is active.
- Join messages contain a bit field that, for each VLAN in the normal range, indicates whether it is active or unused (that is, pruned)



In any VTP version, VTP messages are transmitted and accepted only on trunk ports. Access ports neither send nor accept VTP messages. For two switches to communicate in VTP, they must first be interconnected through a working trunk link.

3.4.3. VTP Domain

- Controls which devices can exchange VTP advertisements
- Defaults to NULL value
- Switch inherits VTP domain name of first received advertisement over trunk links
- A switch can only be part of one domain at a time
- case-insensitive ??? CHECK in lab
- vtp needs at least 10Mb Ethernet CHECK

Task: Set the VTP Domain Name

```
(config)# vtp domain <name>
```

3.4.4. Configuration Revision Number

- 32-bit
- Incremented by one for each configuration change
- Higher revision indicates newer database

For a newly connected VTP server or client to change another switch's VTP database, the following must be true:



- The new link connecting the new switch is trunking.
- The new switch has the same VTP domain name as the other switches.
- The new switch's revision number is higher than that of the existing switches.
- The new switch must have the same password, if configured on the existing switches.

3.4.5. VTP Modes

You can configure a switch to operate in any one of these VTP modes:

Server

- Default mode
- Allows addition, deletion and modification of VLAN information
- Changes on server overwrite the rest of the domain
- Configuration saved in NVRAM

Task: Configure the Switch As a VTP Server

```
(config)# vtp mode server
```

Client

- Cannot add, remove or modify VLAN information
- Listens for advertisements originated by server, install them and passes them on
- Configuration saved in NVRAM only for VTPv3

Task: Configure the Switch As a VTP Client

```
(config)# vtp mode client
```

Transparent

- Keeps a separate VTP database from the rest of the domain
- Does not originate advertisements
- "transparently" passes received advertisements through without installing them
- Can still create, remove or renamed VLANs which are not advertised to neighboring switches.
- Need for some applications like Private VLANs

Task: Setup VTP Transparent Mode

```
(config)# vtp mode transparent
```

Off (configurable only in CatOS switches)

- Like VTP transparent mode with the exception that VTP advertisements are not forwarded

Table 5. VTP Modes and Features

Function	Server Mode	Client Mode	Transparent Mode
Originates VTP advertisements	Yes	Yes	No
Processes received advertisements to update its VLAN configuration	Yes	Yes	No
Forwards received VTP advertisements	Yes	Yes	Yes
Saves VLAN configuration in NVRAM or vlan.dat	Yes	Yes	Yes
Can create, modify, or delete VLANs using configuration commands	Yes	No	Yes

3.4.6. VTP Security

- MD5 authentication prevents against certain attacks
 - Does not prevent against misconfiguration
 - Password must be setup manually because switches only exchanges MD5 digest of the password.

Task: Configure VTP Authentication

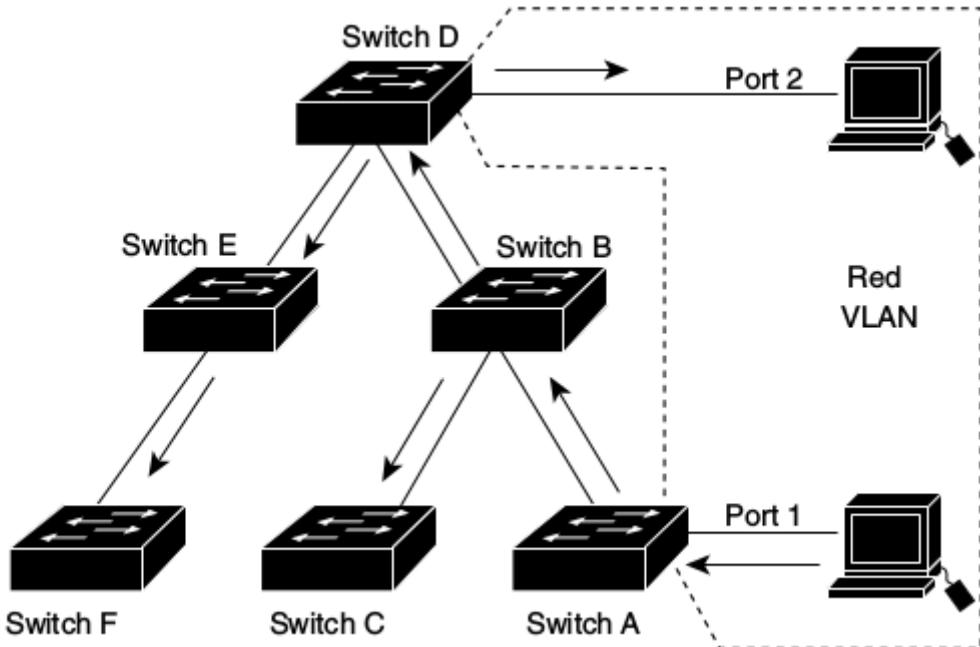
```
(config)# vtp password <string>
```

Task: Show the VTP Password

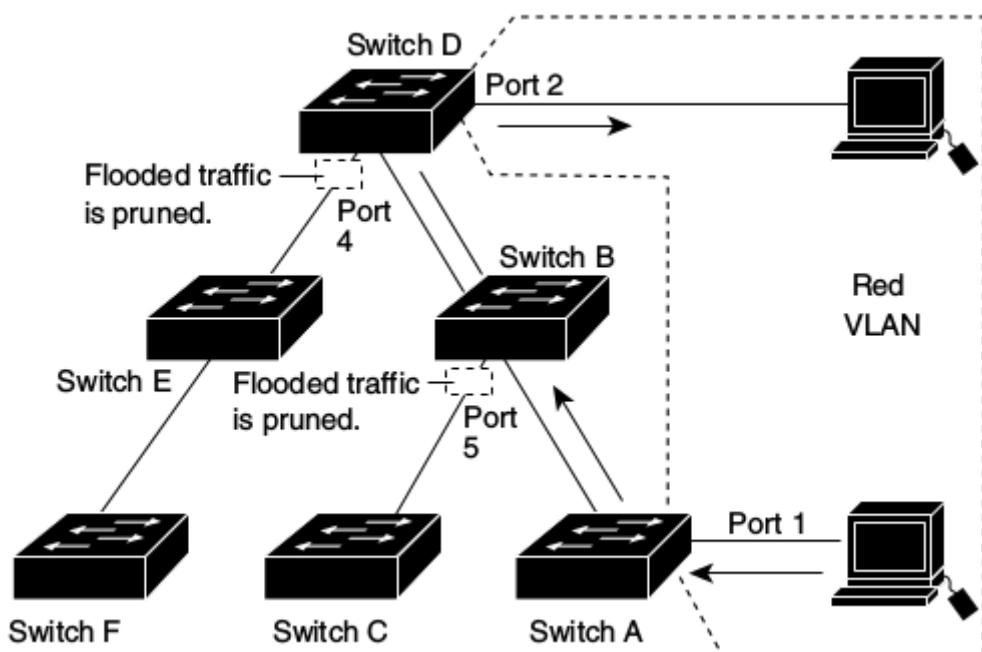
```
(config)# sh vtp password
```

3.4.7. VTP Pruning

- Problem:
 - Broadcasts and unknown unicast/multicast frame are flooded everywhere in the broadcast domain included through trunks links
 - Manual editing allowed list is a huge administrative overhead



- Solution: VTP pruning
 - Switches advertise what they need
- All other VLANs are pruned off the trunk link



- Restriction:
 - Pruning does not work in transparent mode. Why?

Pruning Eligibility

- When VTP pruning is enabled on a VTP server, pruning is enabled for the entire management domain except for pruning-ineligible VLANs (Vlan 1, 1002-1005, 1006-4094)
- Making VLANs pruning-eligible or pruning-ineligible affects pruning eligibility for those VLANs on that trunk only (not on all switches in the VTP domain).

- VTP pruning takes effect several seconds after you enable it.

3.4.8. Troubleshooting

<http://www.cisco.com/c/en/us/support/docs/lan-switching/vtp/98155-tshoot-vlan.html#topic9>

3.5. DTP

Dynamic Trunk Protocol

- negotiate trunk status
- default to **dynamic auto**
- both sides must be on the same VTP domain or one must be in the NULL domain

Table 6. Trunking Configuration Options That Lead to a Working Trunk

Configuration Command	Short name	Meaning	To trunk other side must be
switchport mode trunk	Trunk	Always trunks on this end; sends DTP to help other side choose to trunk	On, desirable, auto
switchport mode trunk ; switchport nonegotiate	Nonegotiate	Always trunks on this end; does not send DTP messages (good when other switch is a non-Cisco switch)	On
switchport mode dynamic desirable	Desirable	Sends DTP messages, and trunks if negotiation succeeds	On, desirable, auto
switchport mode dynamic auto	Auto	Replies to DTP messages, and trunks if negotiation succeeds	On, desirable
switchport mode access	Access	Never trunks; sends DTP to help other side reach same conclusion	Never trunks
switchport mode access; switchport nonegotiate	Access (with nonegotiate)	Never trunks; does not send DTP messages	(Never trunks)

Task: Configure an Inter-Switch Link to Be In Dynamic Desirable State

```
(config-if)# switchport mode dynamic desirable
```

Task: Disable DTP for a Port Administratively Configured As a Trunk

```
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
```

Task: Put the Interface Into Permanent Nontrunking Mode

```
(config-if)# switchport mode access
```

Task: Display a Summary Of Trunk-Related Information

```
show interface trunk: Summary of trunk-related information
```

Task: List Trunking Details for a Specified Interface

```
show interface <type number> trunk
```

Task: List Nontrunking Details for a Particular Interface

```
show interface <type number> switchport
```

Task: Display DTP Information for the Switch

```
# show dtp
```

Task: Display DTP Information for a Specific Interface

```
# show dtp interface <type slot/number>
```

3.5.1. Trunking Between a Switch and a Router

Because DTP is not supported on Router

- on the router, create a sub-interface for each desired vlan
- on the switch, disable DTP and manually configure the trunk

Task: Enable Trunking but Disable DTP for Routers

```
! SW1
conf t
int e0/0
  switchport trunk enc dot1q
  switchport mode trunk
  switchport nonegotiate
```

```
! R1
conf t
int e0/0.1
  enc dot1q <vlan-id> [native]
```

3.5.2. Verify

What is TOS/TAT in

! The TOS/TAS/TNS stand for Trunk Operating/Administrative/Negotiation Status ! The TOT/TAT/TNT stand for Trunk Operating/Administrative/Negotiation Type

```
sh dtp interface fa 0/19
```

3.5.3. DTP and VTP

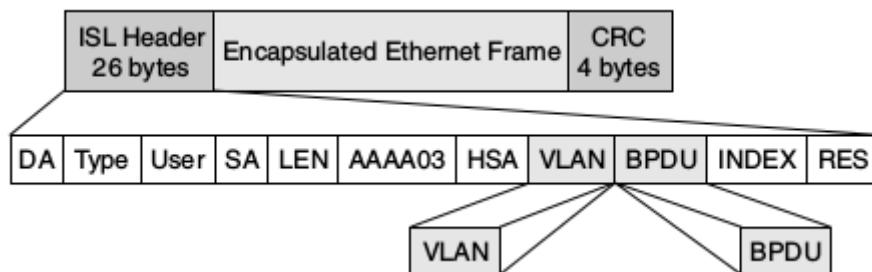
While DTP and VTP are independent protocols, DTP carries the VTP domain name in its messages. Switches will successfully negotiate the link operating mode only if the VTP domain name on both switches is the same, or one switch has no VTP domain name configured yet (that is, it uses a NULL domain name). The reason behind tying the DTP negotiation to the VTP domain name is that in different VTP domains, there might be different sets of VLANs, and identically numbered VLANs might be used for different purposes (that is why the network was split into several VTP domains in the first place—to keep the VLAN databases separate and independent). As a result, switches should not try to bring up the link as a trunk, as extending VLANs from one VTP domain to another can have undesired consequences.

3.6. ISL

- Inter-Switch Link
- Cisco proprietary
- Provides VLAN trunking
- Supports normal and extended VLANs
- Encapsulates the original header with 26-byte header
- Removes the header at the receiving end

3.6.1. Frame

The ISL frame consists of three fields: the ISL header(26 bytes), the original frame and the FCS (4 bytes)



DA—Destination Address

- 40-bit
- Multicast address: "0100-0C00-00" or "0300-0C00-00".
- The first 40 bits of the DA field signal the receiver that the packet is in ISL format. ???

TYPE—Frame Type

- 4 bits

- Indicates the type of the original frame
 - 0000: Ethernet
 - 0001: Token Ring
 - 0010: FDDI
 - 0011: ATM

USER—User Defined Bits (TYPE Extension)

- 4 bits
- Extends the meaning of the TYPE field
- Default value: "0000"
- For Ethernet frames, the USER field bits "0" and "1" indicate the priority of the packet as it passes through the switch. Whenever traffic can be handled in a manner that allows it to be forwarded more quickly, the packets with this bit set should take advantage of the quick path. It is not required that such paths be provided.
 - XX00 Normal Priority
 - XX01 Priority 1
 - XX10 Priority 2
 - XX11 Highest Priority

SA—Source Address

- 48 bits set to set MAC address of the switch port that transmits the frame.
- May be ignored by the receiving device

LEN—Length

- 16 bits set to the length of the packet in bytes with the exclusion of the DA, TYPE, USER, SA, LEN, and FCS fields.

AAAA03 (SNAP)—Subnetwork Access Protocol (SNAP) and Logical Link Control (LLC)

- 24 bits set to "0xAAAA03".

HSA—High Bits of Source Address

- 24 bits set to 0x00-00-0C (Cisco OUI) of the SA field.

VLAN—Destination Virtual LAN ID

- 15 bits set to the VLAN ID of the frame

BPDU—BPDU and CDP Indicator

- 1 bit set when STP or CDP encapsulates an ISL packet

INDX—Index

- 16 bits set to the port index of the source of the packet as it exits the switch
- Used for diagnostic purposes only
- May be ignored by the receiving bridge

RES—Reserved for Token Ring and FDDI

- 16 bits used when Token Ring or FDDI packets are encapsulated with an ISL frame
 - In the case of Token Ring frames, the Access Control (AC) and Frame Control (FC) fields are placed here.
 - In the case of FDDI, the FC field is placed in the Least Significant Byte (LSB) of this field.
- For Ethernet packets, the RES field should be set to all zeros.

ENCAP FRAME—Encapsulated Frame

- Encapsulated data packet with its own CRC value completely unmodified
- Length from 1 to 24575 bytes

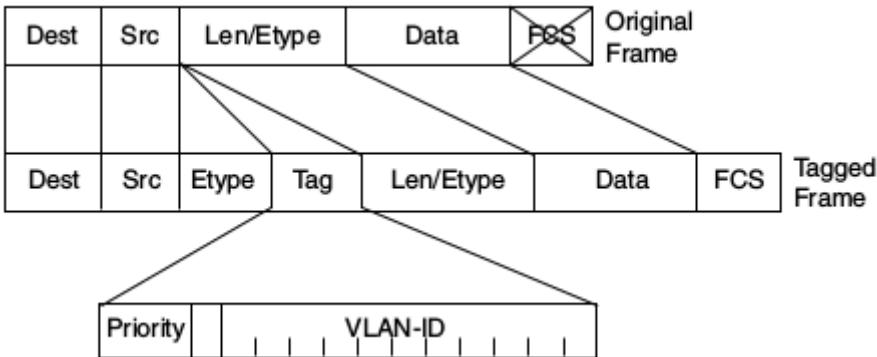
FCS—Frame Check Sequence

- 4 bytes set by the sending MAC and recalculated by the receiving bridge
- New FCS calculated over the entire ISL packet

3.7. IEEE 802.1Q

- Tags frames on a trunk
 - Inserts a 4-byte tag into the original frame between the Source Address and the Type/Length field
 - Recomputes the frame check sequence (FCS) before the device sends the frame over the trunk link.
 - Removes the tag at the receiving end
- Does not tag frames on the native VLAN.
 - Must use the same native VLAN on both sides of the trunk
 - Default to VLAN 1
 - enables frames to transit switches not yet capable for 802.1g
- Supports up to 4096 VLANs
 - Defines a single instance of spanning tree that runs on the native VLAN for all the VLANs in the network.
 - lacks the flexibility and load balancing capability of PVST that is available with ISL.
 - PVST+ offers the capability to retain multiple spanning tree topologies with 802.1Q trunking.

3.7.1. Frame Format



TPID-Tag Protocol Identifier

- 16 bits
- Value: 08100

Priority

- 3 bits
- Called also user priority or IEEE 802.p
- Indicates the frame priority level
- Can be used to prioritize the traffic

CFI-Canonical Format Indicator

- 1 bit
- Value: 0 if MAC address is in canonical format otherwise 1

VID-VLAN Identifier

- 12 bits
- Identifies the VLAN to which the frame belongs

3.7.2. Ethernet Frame Size with 802.1Q Tagging

- Maximum size: 1522 bytes
- Minimum size: 68 bytes

3.7.3. Native VLAN

Task: Configure a Native VLAN Over a Trunk Link

```
(config-if)# switchport trunk native vlan <id>
```

TODO How can you force the tagging of the native vlan?

3.8. 802.1Q-In-Q Tunneling

- Adds a metro tag or PE-VLAN to the 802.1q tagged packets

- Expands the VLAN space by double-tagging frames
- Allows Service Providers
 - to preserve 802.1Q VLAN tags across WAN links.
 - to provide services such as Internet access on specific VLANs for specific customers, yet providing other services on other VLANs

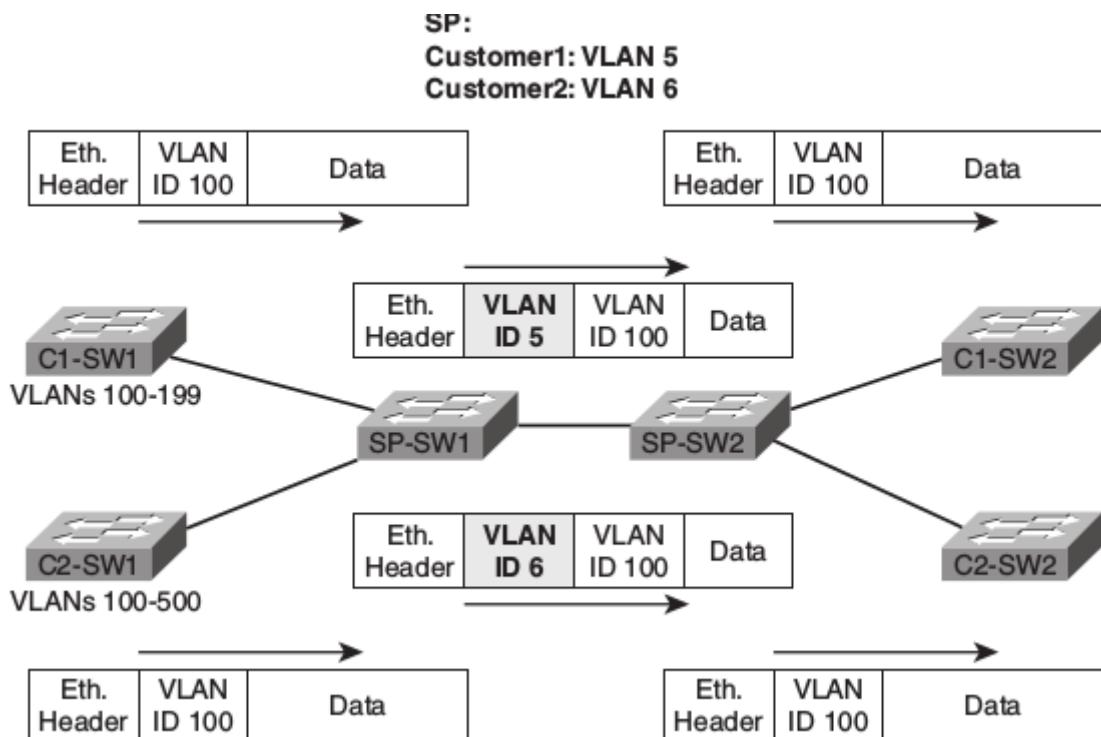
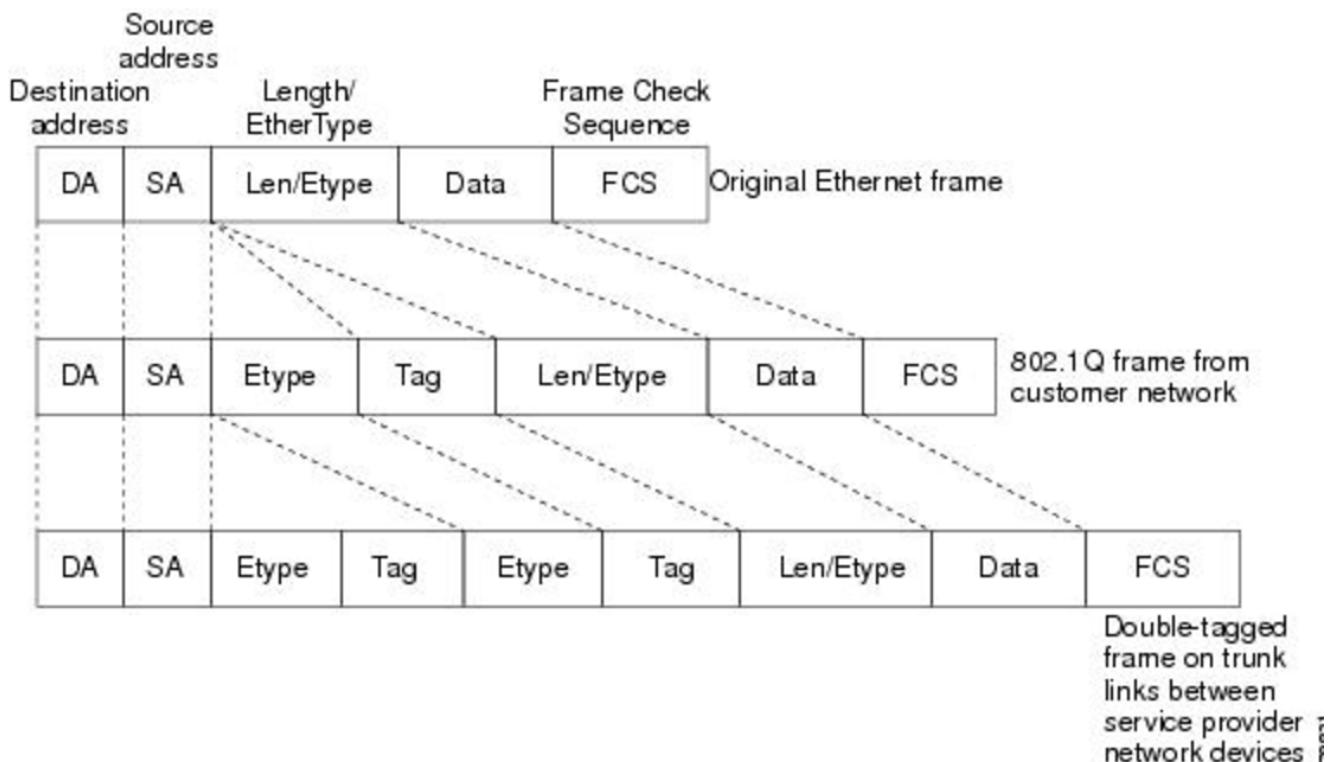


Figure 9. Q-In-Q: Basic Operation

3.8.1. Frame



Frame Size

- Recommended minimum MTU: 1504 bytes
 - default MTU: 1500 bytes
 - outer VLAN tag: 4 bytes

TPID

- Contains the modified tag protocol identifier
- Set to 0x8100 for IEEE 802.1q

The QinQ frame contains the modified tag protocol identifier (TPID) value of VLAN Tags. By default, the VLAN tag uses the TPID field to identify the protocol type of the tag. The value of this field, as defined in IEEE 802.1Q, is 0x8100.

The device determines whether a received frame carries a service provider VLAN tag or a customer VLAN tag by checking the corresponding TPID value. After receiving a frame, the device compares the configured TPID value with the value of the TPID field in the frame. If the two match, the frame carries the corresponding VLAN tag. For example, if a frame carries VLAN tags with the TPID values of 0x9100 and 0x8100, respectively, while the configured TPID value of the service provider VLAN tag is 0x9100 and that of the VLAN tag for a customer network is 0x8200, the device considers that the frame carries only the service provider VLAN tag but not the customer VLAN tag.

In addition, the systems of different vendors might set the TPID of the outer VLAN tag of QinQ frames to different values. For compatibility with these systems, you can modify the TPID value so that the QinQ frames, when sent to the public network, carry the TPID value identical to the value of a particular vendor to allow interoperability with the devices of that vendor. The TPID in an Ethernet frame has the same position with the protocol type field in a frame without a VLAN tag. In order to avoid problems in packet forwarding and handling in the network, you cannot set the TPID value to any of the values in this table:

Protocol type	Value
ARP	0x0806
PUP	0x0200
RARP	0x8035
IP	0x0800
IPv6	0x86DD
PPPoE	0x8863/0x8864
MPLS	0x8847/0x8848
IS-IS	0x8000
LACP	0x8809
802.1x	0x888E

The QinQ Support feature is generally supported on whatever Cisco IOS features or protocols are

supported. For example, if you can run PPPoE on the subinterface, you can configure a double-tagged frame for PPPoE. IPoQinQ supports IP packets that are double-tagged for QinQ VLAN tag termination by forwarding IP traffic with the double-tagged (also known as stacked) 802.1Q headers.

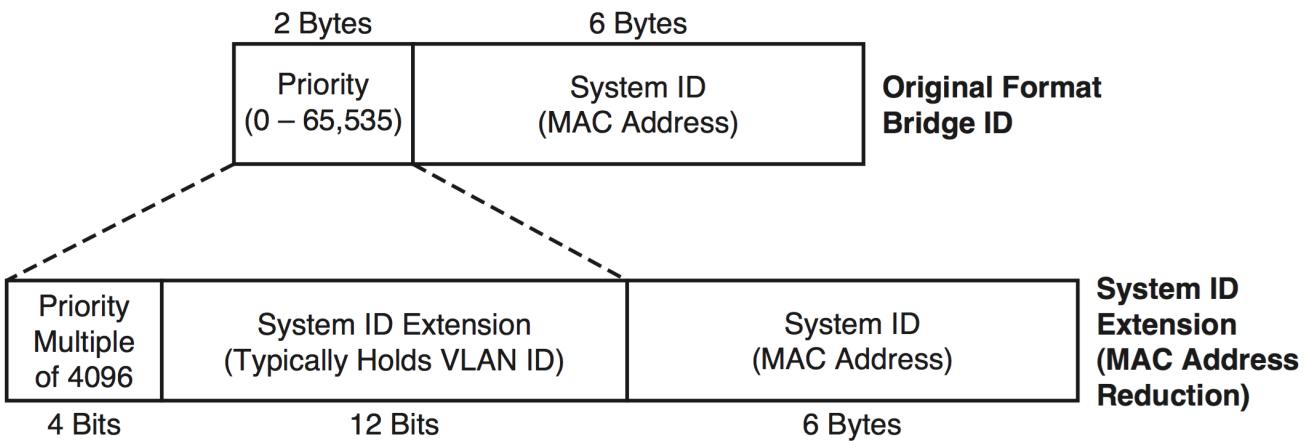
Chapter 4. Spanning Tree Protocols

3750x Configuration Guides › STP

- Creates loop-free layer 2 topology
- Prevents broadcast storms
- STP variations:
 - 802.1d : Common Spanning Tree
 - PVST/PVST+ : Cisco per-VLAN Spanning Tree
 - 802.1w : Rapid Spanning Tree Protocol
 - 802.1s : Multiple STP

4.1. 802.1d Common Spanning Tree

- 1 instance
- Uses BPDU
- Elect one root switch and one designated switch for each segment
- One root port per non-root switch, one designated port for each segment
- Other ports on blocking state
- Steps
 - Elect the root switch with the lowest bridge id (2-byte priority + 6-byte MAC)
 - Determine each switch's root port: with the least cost path to the root
 - Determine the designated port for each segment: the switch that forwards the least cost Hello on the segment
 - If there is a tie, select the lowest port ID
- Original IEEE 802.1d bridge Id
 - 2-byte priority
 - 6-byte MAC address
- Revised IEEE 802.1d bridge id Priority for MAC address reduction
 - 4 bits : priority multiple of 4096
 - 12 bits : system id extension (vlan id) to support pvst+ and IEEE 802.1s



4.1.1. BPDU

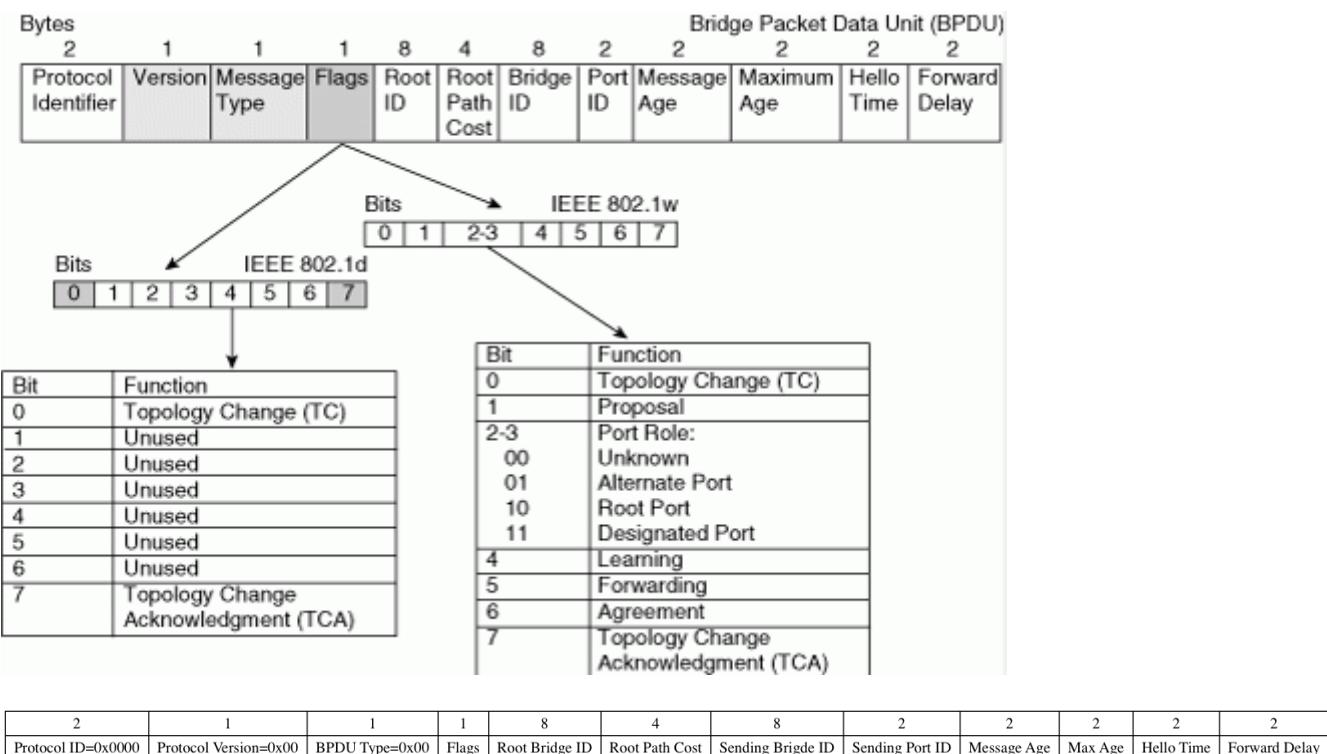


Figure 10. Configuration BPDU

- The Flags field uses 2 bits out of 8 to handle topology change events: the Topology Change Acknowledgment flag and the Topology Change flag.
- The MessageAge field is an estimation of the BPDU's age since it was originated by the root bridge. At the root bridge, it is set to 0. Any other switch will increment this value, usually by 1, before forwarding the BPDU further. The remaining lifetime of a BPDU after being received by a switch is MaxAge-MessageAge. Finally, the remaining fields carry the values of STP timers: MaxAge, HelloTime, ForwardDelay. These timer values always reflect the timer settings on the root switch. Timers configured on a nonroot switch are not used and would become effective only if the switch itself became the root switch.

2	1	1
Protocol ID=0x0000	Protocol Version=0x00	BPDU Type= 0x80

Figure 11. Topology Change Notification BPDU

TODO: Find a better position

- VLAN1 STP BPDU are sent untagged
 - to IEEE STP MAC 0180.c200.0000
 - to PVST+
- Non-VLAN1 STP BPDU are sent to PVST+ MAC 0100.0CCC.CCCD
- To determine which BPDU out of a pair of configuration BPDUs is superior, compare the following sequence of values, looking for the first occurrence of a lower value: Root Bridge ID, Root Path Cost, Sender Bridge ID, Sender Port ID, Receiver Port ID (not included in BPDU; evaluated locally)
- Each port in STP stores/remembers the superior BPDU it has either sent (DP port) or received (RP and Blocking Ports). Essentially, each port stores the DP's BPDU—whether it is the port itself that is Designated or it is a neighbor's port. Should a port stores a received BPDU, it must be received again within a time interval of MaxAge-MessageAge seconds; otherwise it will expire after this period. This expiry is always driven by the root switch's timers in the BPDU.

4.1.2. Root Bridge

- Election with hello BPDU
 - Each switch begins its STP logic by creating and sending an Hello BPDU message, claiming itself to be the root switch.
 - If a switch hears a superior Hello to its own Hello bridge ID, it stops claiming to be root by ceasing to originate and send Hellos. Instead, the switch starts forwarding the superior Hellos received from the superior candidate.
 - Eventually, all switches except the switch with the lowest bridge ID cease to originate Hellos; that one switch wins the election and becomes the root switch.

Task: Force Election Of a Root Bridge

```
# spanning-tree vlan <id> root [primary|secondary]
```



Going towards root uses priority, Going away from root uses cost.

4.1.3. Root Port

- RP is upstream facing towards Root bridge
- Lowest root path cost (cumulative cost of all links to get to the root)
 - cost = advertised cost in the BPDU hello + cost on the receiving post

- Cost based on inverse bandwidth

TODO right justify the numbers in table below

Table 7. Default Port Costs

Speed	Original	Revised	802.1D-2004
10 Mbps	100	100	2000000
100 Mbps	10	19	200000
1 Gbps	1	4	20000
10 Gbps	1	2	2000

Task: Choose Default STP Path Cost (Original or Revised)

```
(config)# spanning-tree pathcost method {short | long}
```

Tie breaker when a switch receives multiple Hellos with equal cost

1. Lowest Bridge Id
2. Lowest Port Priority
3. Lowest Port Number

4.1.4. Designated Port

- Designated switch: send the Hello with the lowest advertised cost for the segment
- DP: port that forward frames onto that segment
- DP are downstream facing away from root bridge
- Elected based on lowest root path cost, BID, port ID

4.1.5. Blocking Ports

- Receive BPDU's
- Discard all other traffic
- Cannot send traffic
- Do not send Hellos

4.1.6. Convergence

- Steady operations: one Root bridge, one RP on each non-root bridge, one DP on each segment, blocking state
 1. Root Switch Generates a Hello Every 2 Seconds
 2. Each RP on Non-Root Switch Receives a Copy Of the Root's Hello
 3. Each DP Updates and Forwards the Hello Out

4. Each Blocking Port Receives a Copy Of the Hello from the DP Without Forwarding It

4.1.7. Topology Change Notification

- Topology change: event that occurs when
 - A TCN BPDU is received by a DP of a switch
 - A port moves to the Forwarding state and the switch has at least one DP (meaning that it is not a standalone switch with just a Root Port connected to an upstream switch and no other connected ports)
 - A port moves from Learning or Forwarding to Blocking
 - A switch becomes the root switch

more at [understand new topology changes](#)

TODO Split this section for 802.1d and 802.1w

1. A switch experiencing the STP port state change sends a TCN BPDU out its root port; it repeats this message every Hello time until it is acknowledged.
2. The next switch receiving that TCN BPDU sends back an acknowledgment via its next forwarded Hello BPDU by marking the Topology Change Acknowledgment (TCA) bit in the Hello.
3. The switch that was the DP on the segment in the first two steps repeats the first two steps, sending a TCN BPDU out its Root Port, and awaiting acknowledgment from the DP on that segment.

By each successive switch repeating Steps 1 and 2, eventually the root receives a TCN BPDU. Once received, the root sets the TC flag on the next MaxAge + ForwardDelay seconds, which are forwarded to all switches in the network, notifying them that a change has occurred. A switch receiving a Hello BPDU with the TC flag set uses the short (Forward Delay time) timer to time out entries in the CAM.

Table 8. Transitioning from Blocking to Forwarding

State	Forward data frames	Learn source MAC	Stable?
Blocking	No	No	Yes
Listening	No	No	No
Learning	No	Yes	No
Forwarding	Yes	Yes	Yes
Disabled	No	No	Yes

4.1.8. Timers

Hello timer

- 2 seconds
- Interval at which the root sends Hellos

Forward delay

- 15 seconds
- Time that switch leaves a port in listening state and learning state
- also used for the short CAM timeout timer

Maxage

- 20 seconds
- Time without hearing a Hello before believing that the root has failed

TODO Add tasks to modify default timers .Task:

```
(config)#
```

4.2. PVST+ Per-Vlan STP

- Per-VLAN STP : for better load balancing
 - One instance of legacy STP per VLAN
 - CISCO ISL support
- PVST+
 - One instance of legacy STP per VLAN
 - CISCO ISL and 802.1q support
 - Interoperability between CST and PVST
- default mode on most Catalyst platforms
- allows root bridge/port placement per VLAN
- Non-CISCO + 802.1q ⇒ one Common Spanning Tree over vlan 1
- When mixing CISCO and non CISCO switches with 802.1q trunking,
 - Send BPDU to multicast destination MAC of 0100.0CCC.CCCD

TODO add picture here pp. 78

Task: Display Spanning-Tree Information

```
# sh spanning-tree root
# sh spanning-tree vlan 1 root detail
```

4.3. Optimizing, Improving Spanning Tree

4.3.1. PortFast

- Used on access ports connected to end users devices not other switches

- Puts the port into forwarding state immediately
- Prevent them to generate TCNs
- Can generate loops if another switch is connected. so must be used with BPDU guard and root guard features

Task: Enable Portfast on a given interface

```
(config-if)# spanning-tree portfast
```

Task: Enable Portfast globally

```
(config)# spanning-tree portfast default
```

- The port must be an access port
- If the port is configured as trunk,
 - the global portfast command will not convert the port to an edge port.
 - use **spanning-tree portfast trunk**
- If BPDUs are received on the port, the port may transition to blocking



Task: Disable Portfast on a given interface

```
(config-if)# spanning-tree portfast disable
```

4.3.2. UplinkFast

- Used on access layer switches that have multiple uplinks to distribution/core switches
- Immediately replaces a lost RP with an alternate RP
- Increases the root and all port priority so the switch does not become root or transit switch
- Time-out the correct entries in their CAMs but doesn't use the TCN process. Instead, finds all the MAC addresses of local devices and sends one multicast frame with each local addresses as the source MAC causing all the other switches to update their CAMs. The access switch also clears out the rest of the entries in its own CAM.
- Used only if the switch runs legacy STP because it is built in to RSTP802.1w
- can not be enabled on a switch that has its default STP priority modified
- can not be enabled on root bridge

Task: Enable Uplink Fast

```
(config)# spanning-tree uplinkfast [max-update-rate rate]
```

Task: Use default STP priority for all VLANs

```
(config)# default spanning-tree vlan 1-4094 priority
```

4.3.3. BackboneFast

- Used in core switches to detect indirect link failures to the Root
 - All switches must have backbone fast configured
 - Cisco proprietary for legacy 802.1d STP (now included in RSTP and MSTP)
- Do not wait for MaxAge to expire when another switch's direct link fails
- Send a Root Link Query out the port in which the missing Hello should arrive.
 - The RLQ asks the neighboring switch if that neighboring switch is still receiving Hellos from the root.
 - If that neighbor had a direct link failure, it can tell the original switch via another RLQ that this path to the root is lost.
 - Once known, the switch experiencing the indirect link failure can go ahead and converge without waiting for MaxAge to expire

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

Backbone Fast works in two stages. In the first stage, when a switch receives inferior BPDUs through a nondesignated port (NDP), it knows the switch that sent the inferior BPDUs has lost its connection to the root bridge. When the local switch receives the inferior BPDUs, it verifies if the source that generated these messages is from a local segment. If the source is from a local segment, it knows that an indirectly connected link failure has occurred. If the source of the inferior BPDUs is from a switch that is not on a local segment, the local switch will ignore them.

In the second stage, the local switch goes through a verification process. The switch uses a request and response protocol. This process queries other switches to determine if the connection to the root bridge is lost.

The verification process is done by the local switch that received the inferior BPDUs. The switch generates Root Link Query (RLQ) requests. These messages are sent out of the root port(s).

These messages are sent to query the upstream switch(es) if their connection to the root bridge is up. The receiving switch sends RLQ responses to reply to the[...]” Therefore, it expires the max-age timer in order to speed up the convergence.”

4.3.4. BPDU Guard

- Ensures that unauthorized switches cannot be plugged in to the network
- Puts a portfast enabled port into the errdisable state when a BPDU is received and shuts down the port
- The port must be manually re-enabled or it can be recovered automatically through the

errable timeout function.

- A port configured with bpdu guard will not be put into the root-inconsistent state.

Task: disable an interface if a BPDU is detected

```
(config-if)# spanning-tree bpduguard enable
```

Task: Re-enable the interface with BPDU Guard after n seconds

```
(config)# errdisable recovery cause bpduguard  
(config)# errdisable recovery interval <seconds>
```

Task: Verify the status and reason for a err-disabled interfaces

```
# sh interfaces <interface type id> status err-disabled
```

4.3.5. BPDU Filter

- Filter BPDUs out for all portfast interfaces
- when configured at interface level
 - silently drops all received inbound BPDUs
 - doesn't send any outbound BPDUs
 - the port never goes into err-disabled state
 - may cause permanent loops if a switch is connected
- when configured at switch level
 - only affected PortFast-enabled ports
 - transmits 10 BPDUs at startup
 - disables portfast and portfast bpdu filtering if BPDU receives during that time
 - after startup, if BPDU are received, disables portfast and bpdufilter and acts as other STP port



When PortFast is enabled on a port, the port will still send out BPDUs and it will accept and process received BPDUs. The BPDU Guard feature would prevent the port from receiving any BPDUs, but it will not prevent it from sending them. The BPDU Filter feature effectively disables STP on the selected ports by preventing them from sending or receiving any BPDUs.

Task: Enable BPDU filter at the interface level

```
(config-if)# spanning-tree bpdufilter enable
```

Task: Enable BPDU filter on all portfast-enabled ports

```
(config)# spanning-tree portfast bpdufilter default
```

4.3.6. Loop Guard

- Protects against unidirectional links
- Prevents non-designated ports from inadvertently forming layer 2 switching loops if the flow of BPDUs is interrupted.
- Puts the port into the **loop-inconsistent** state when the steady flow of BPDUs is interrupted
- Only used on point-to-point links
- Can be used with **UDLD aggressive mode** to get extra protection.
- Cannot be enabled at the time with root guard on the same port
- When configured at the switch level, only monitors Non-Designated ports
- Takes actions on a per-VLAN level (although configured on a port)
 - If a trunk port is in blocking state and stops receiving BPDUs for VLAN 8 from the DP on the segment, it transitions the port into *loop inconsistent only for that VLAN 8
- Recovers automatically when the port starts receiving BPDUs

STP Loop Guard is an added logic related to receiving BPDU s on Root and Alternate Ports on point-to-point links. In the case of a unidirectional link, these ports could move

from Root or Alternate to Designated, thereby creating a switching loop. STP Loop Guard assumes that after BPDU s were being received on Root and Alternate Ports, it is

not possible in a correctly working network for these ports to suddenly stop receiving

BPDU s without them actually going down. A sudden loss of incoming BPDU s on Root and Alternate Ports therefore suggests that a unidirectional link condition might have occurred.

Following this logic, STP Loop Guard prevents Root and Alternate Ports from becoming

Designated as a result of total loss of incoming BPDU s. If BPDU s cease being received on

these ports and their stored BPDU s expire, Loop Guard will put them into a loopinconsistent

blocking state. They will be brought out of this state automatically after they start receiving BPDU s again.

Loop Guard can be activated either globally or on a per-port basis, and is a local protection

mechanism (that is, it does not require other switches to be also configured with Loop Guard to work properly). If activated globally using the spanning-tree loopguard

default command, it automatically protects all Root and Alternate Ports on STP point-topoint

link types on the switch. Global Loop Guard does not protect ports on shared type links. It can also be configured on a per-port basis using the spanning-tree guard loop

command, in which case it applies even to ports on shared links

"With the Loop Guard feature enabled, switches do an additional check before transitioning to the STP forwarding state. If switches stop receiving BPDUs on a nondesignated port with the Loop Guard feature enabled, the switch places the port into loop-inconsistent blocking state instead of moving through the listening, learning, and forwarding states. If a switch receives a BPDU on a port in the loop-inconsistent STP state, the port will transition through STP states in accordance with the received BPDU. As a result, recovery is automatic, and no manual intervention is necessary.

When implementing Loop Guard, you should be aware of the following implementation guidelines:

- Loop Guard cannot be enabled simultaneously with Root Guard on the same device.
- Loop Guard does not affect UplinkFast or Backbone Fast operation.
- Loop Guard must be enabled on point-to-point links only.
- Loop Guard operation is not affected by the spanning tree timers.
- Loop Guard cannot actually detect a unidirectional link.
- Loop Guard cannot be enabled on PortFast or dynamic VLAN ports.
- "state). However, in a case where we are dealing with an aggregated link between two devices, all of the links in the aggregate will transition into the inconsistent state for the particular VLAN that is no longer receiving BPDUs."

You configure the Loop Guard feature on a per-port basis, even though the feature is designed to block inconsistent ports on a per-VLAN basis.

4.3.7. Root Guard

- Prevent a port from becoming a root port when receiving a superior BPDU (e.g. inferior priority + mac)
- It is enabled on ports other than the root port and on switches other than the root
- Puts the port in **root-inconsistent** state (no data flow) until it stops receiving superior BPDUs. No traffic is forwarded.
- Enforce the root bridge placement by ensuring the port on which root guard is enabled is the designated port.
- Ensures that the port on which root guard is enabled is the designated port.
- differences with BPDU guard
 - root-inconsistent state vs err-disabled
 - automatic recovery vs manual recovery
 - cannot be configured globally

This message appears after root guard blocks a port:

```
%SPANTREE-2-ROOTGUARDBLOCK: Port 1/1 tried to become non-designated in VLAN 77.  
Moved to root-inconsistent state
```

- Read more at [Root Guard PortFast BPDU Guard](#)

4.4. 802.1w Rapid STP

- Improves convergence by
 - Waiting for only 3 missed Hellos on an RP before flushing the CAM instead of 10 (10 x 2 seconds = MaxAge) with 802.1d
 - Bypass listening state
 - Includes natively Cisco PortFast, UplinkFast, BackboneFast
 - Add backup DP when multiple ports connected to the same segment
 - Doesn't use MessageAge
 - immediate acceptance of inferior BPDUs from DP
 - an inferior BPDU originated by a designated switch on a segment is accepted right away, immediately replacing previously stored BPDUs on receiving ports of attached switches. In other words, if a designated switch on a segment suddenly sends an inferior BPDU, other switches on the segment will immediately accept it as if the superior stored BPDU expired just when the inferior BPDU arrived, and reevaluate their own port roles and states on the segment according to usual rules. This behavior allows a switch to rapidly react to a situation where the neighboring switch experiences a disruptive change in its own connectivity toward the root switch
- Backward compatible with 802.1d
- All bridges generate BPDUs every Hello interval
- Use a single BPDU, No TCN BPDU
- Protocol Version = 0x02
- The Flags field has been updated.
 - In 802.1D STP BPDUs, only 2 bits out of 8 are used: TC (Topology Change) and TCA (Topology Change Acknowledgment)
 - RSTP uses the 6 remaining bits as well to encode additional information: Proposal bit, Port Role bits, Learning bit, Forwarding bit, and Agreement bit.
 - The TCA bit is not used by RSTP. This change allows implementing the **Proposal/Agreement** mechanism and also allows a BPDU to carry information about the originating port's role and state, forming the basis of RSTP's **Dispute** mechanism, protecting against issues caused by unidirectional links.

4.4.1. RSTP Link Types

- **Point-to-point:** Switch to Switch (default if full-duplex port)
- **Shared :** Switch to hub (default if half-duplex port)

Task: Set the RSTP Link-Type

```
spanning-tree link-type { point-to-point | shared }
```

4.4.2. RSTP Port Types

- Edge :
- Non-Edge: default

4.4.3. RSTP Port States

- Default to discarding at start

TODO Improve the table below with spanning (enabled, discarding) over the row

Administrative state	802.1d	802.1w
Disabled	Disabled	Discarding
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding

4.4.4. RSTP Port Roles

Root Port

- Same role as 802.1d RP

Designated Port

- Same role as 802.1d DP
- Default role at boot

Alternate Port

- An alternate root port
- Same concept as Cisco UplinkFast feature
- Protects against the loss of a switch's RP by keeping track of the AP with a path to the root

Backup Port

- No equivalent Cisco feature
- Protects against losing the DP attached to a shared link when the switch has another physical port attached to the same shared segment



root bridge ports are all designated ports unless 2 or more ports of the root bridge are connected together.



a port needs to receive BPDUs to stay blocked.

Task: Configure Rapid PVST

```
(config)# spanning-tree mode rapid-pvst
```



- Rapid PVST+ immediately deletes dynamically learned MAC address entries when it receives a topology change instead of a timer used by PVST+ or MST

4.4.5. Proposal/Agreement Process

The Proposal signifies the willingness of a port to become Designated Forwarding, while the Agreement stands for permission to do so immediately. After a new link point-to-point link is added between two switches, ports on both ends will come up as Designated Discarding, the default role

and state for a Non-Edge port. Any Designated Port in a Discarding or Learning state sends BPDUs with the Proposal bit set. Both switches will therefore attempt to exchange

BPDUs with the Proposal bit set (or simply a Proposal), assuming that they have the right to be Designated. However, if one of the ports receiving a Proposal discovers that

the Proposal constitutes the best received resulting BPDU, its role will change from Designated to Root (the state will remain Discarding yet). Other port roles on that switch

will also be updated accordingly. Furthermore, a switch receiving a Proposal on its Root

Port will immediately put all its Non-Edge Designated ports into a Discarding state. This

operation is called Sync . A switch in Sync state is now isolated from the network, preventing

any switching loop from passing through it: Its Root Port is still in the Discarding state (and even if it was Forwarding, the neighboring Designated Port is still Discarding

or Learning), and its own Designated Ports are intentionally moved to the Discarding state. Now it is safe to move the new Root Port to the Forwarding state and inform the upstream switch that it is now allowed to move its Designated Discarding or Learning port to the Forwarding state. This is accomplished by a switch sending a BPDU with the Agreement bit set (or simply an Agreement) through its Root Port after performing the Sync. Upon receiving an Agreement on its Designated Discarding or Learning port, the upstream switch will immediately move that port into the Forwarding state, completing the Proposal/Agreement exchange between two switches.

4.4.6. Topology Change Handling

- Only a transition of a Non-Edge port from a non-Forwarding state to the Forwarding state is considered a topology change event
- a switch that detects a topology change on a port (that is, one of its own Non-Edge ports transitions into the Forwarding state) or learns about a topology change on a port (a BPDU with

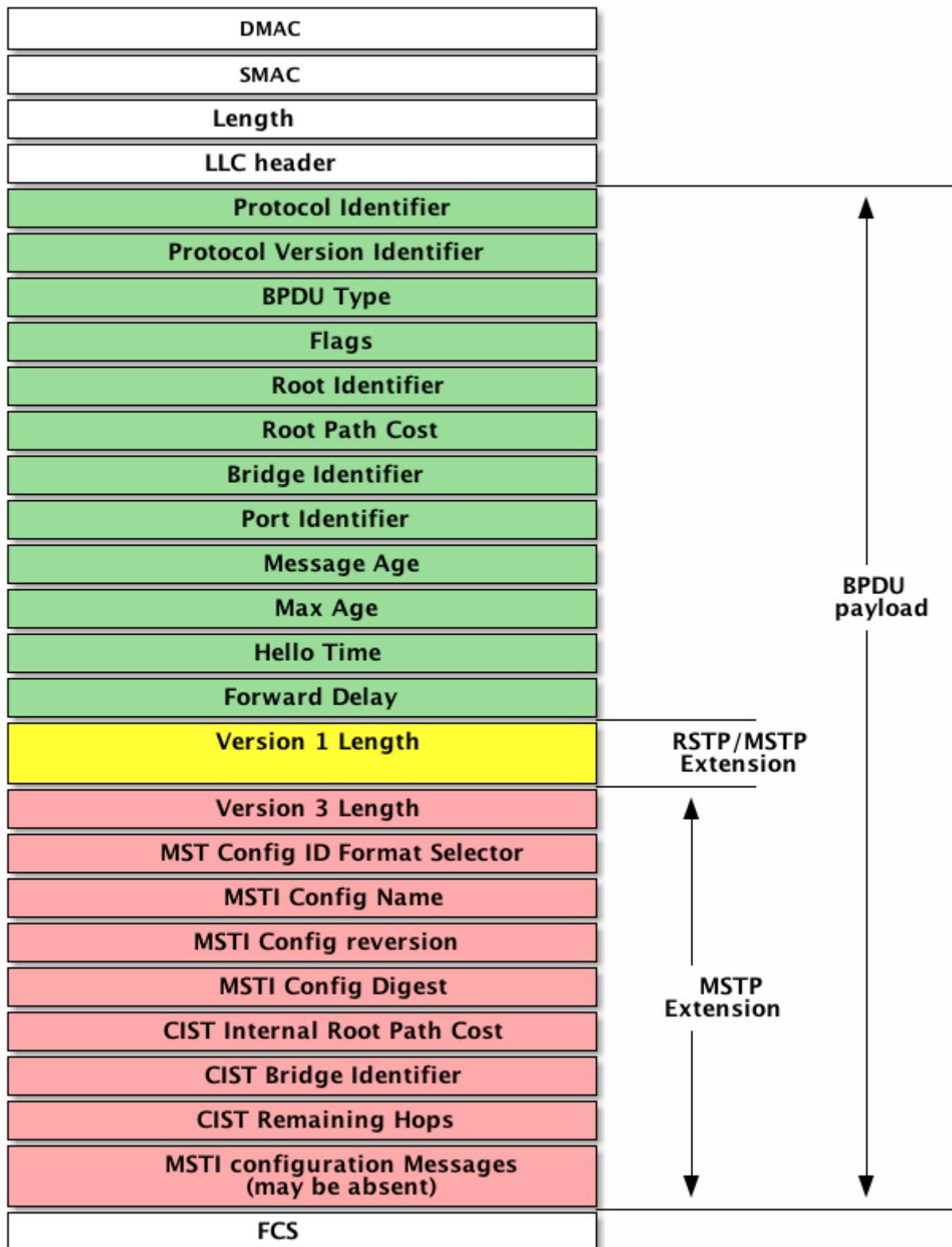
the TC flag set is received on its Root or Designated Port) will do the following:

- Set a so-called tcWhile timer to the value of the Hello time plus one second (older revisions of RSTP set this value to twice the Hello time) on all remaining Non-Edge Designated ports and Root Port if any, except the port on which the topology change was detected or learned.
- Immediately flush all MAC addresses learned on these ports.
- Send BPDUs with the TC flag set on these ports every Hello seconds until the tcWhile timer expires.
- This way, information about a topology change is rapidly flooded along the spanning tree in the form of BPDUs with the TC flag set, and causes switches to immediately flush their CAM tables for all ports except those ports on which the topology change was detected or learned, as they point in the direction of the topology change where a set of MAC addresses might have become reachable through a new or improved path.
- Edge ports never cause a topology change event, and MAC addresses learned on them are not flushed during topology change event handling.

4.5. 802.1s Multiple Spanning Trees

- Multiple VLANs mapped to the same STP instance.
- Enable load balancing
- Improves fault tolerance of the network because a failure in one instance or forwarding path does not affect other instances.
- Uses 802.1w for rapid convergence
- Highly scalable
 - Switches with same instance, configuration revision number and name form a **region**
 - Different regions see each other as virtual bridges
- generates one single BPDU for all configured instances
- runs instance 0 which forms the IST, or CIST in the case of multiple MST regions
- doesn't use the message age and maximum age information to compute the STP topology.
 - use the path cost to the root and a hop-count (similar to IP TTL)
 - the root bridge sends a BPDU (or M-record) with a cost of 0 and the hop count set the maximum value (20 by default)
 - when a switch receives this BPDU, it removes one to the hop counts before sending downstream
 - when the count reaches zero, the switch discards the BPDU and ages the information it holds for the port
 - the message age and maximum age information in the BPDU remain the same throughout the region and propagated by the region's DP at the boundary.

4.5.1. MST Frame format



4.5.2. MST Region

- Each switch have three attributes:
 - Alphanumeric configuration name (32 bytes)

- Configuration number (2 bytes)
- 4096-element table that associates each of the potential 4096 VLANs to a map

Task: configure a MST region

```
conf t
spanning-tree mode mst
spanning-tree mst configuration
  name <region-name>
  revision <number>
  instance <x> vlan <a-b,c, ...>
  instance <y> vlan <a-b,c, ...>
```

4.5.3. MST region ports

- edge port: connects to a non-bridging device or a hub
- boundary port: connects to a single region running RSTP, 802.1d or to another MST region

4.5.4. MST Revision Number

4.5.5. MST Instance

IST

instance 0

- instance that interacts with STP run outside the MST region
- comprises all VLANs not associated with an instance ???
- only one BPDU being shared over the native VLAN of the trunk
- only instance that sends and receives BPDUs
 - other STP instances are contained in the M-record encapsulated within the BPDU
 - reduces the CPU usages
- increments the Root Path Cost and Message Age values at the boundary of the MST region as though the BPDU had traversed only a single switch.
- elects **IST master**

Common and Instance Spanning Tree

- union of CST between regions and ISTs inside individual regions, and is a single spanning tree that spans the entire switched topology. As each MST region has its own IST root, CIST—consisting of ISTs inside regions and CST between regions—can have multiple root switches as a result. These switches are recognized as the CIST Root Switch (exactly one for the entire CIST) and CIST Regional Root Switches (exactly one for the IST inside each region). CIST Regional Root Switch is simply a different name for an IST root switch inside a particular region.
- The CIST Root Switch is elected by the lowest Bridge ID from all switches that participate. The

CIST Root Switch is elected by the lowest Bridge ID from all switches that participate in CIST, that is, from all MST switches across all regions according to their IST Bridge IDs (composed of IST priority, instance number 0, and their base MAC address), and from all STP/RSTP switches, if present, according to the only Bridge IDs they have. If running a pure MST-based network, the CIST Root Switch will be the switch whose IST priority is the lowest (numerically), and in the case of a tie, the switch with the lowest base MAC address. This switch will also become the root of IST inside its own MST region; that is, it will also be the CIST Regional Root Switch. As the CIST Root Switch has the lowest known Bridge ID in the CST, it is automatically the CST Root as well, although this observation would be important only in cases of mixed MST and non-MST environments.

In other MST regions that do not contain the CIST Root Switch, only MST switches at the region boundary (that is, having links to other regions) are allowed to assert themselves as IST root switches. This is done by allowing the CIST Regional Root ID to be set either to the Bridge ID of the switch itself if and only if the switch is also the CIST Root, or in all other cases, to the Bridge ID of an MST boundary switch that receives BPDUs from a different region. Remaining internal switches have therefore no way of participating in IST root elections. From boundary switches, IST root switches are elected first by their lowest external root path cost to the CIST Root Switch. The external root path cost is the sum of costs of inter-region links to reach the region with the CIST Root Switch, or in other words, the CST cost of reaching the region with the CIST Root Switch; costs of links inside regions are not taken into account. In case of a tie, the lowest IST Bridge ID of boundary switches is used. Note that these rules significantly depart from the usual concept of the root switch having the lowest Bridge ID. In MST regions that do not contain the CIST Root Switch, the regional IST root switches might not necessarily be the ones with the lowest Bridge IDs.

A CIST Regional Root Switch has a particular importance for a region: Its own CIST Root Port, that is, the Root Port to reach the CIST Root Switch outside the region, is called the Master port (this is an added port role in MST), and provides connectivity from the region toward the CIST Root for all MST instances inside the region.

CST

Common Spanning Tree

- determines loop-free between regions
- only spanning tree that can be understood and participated in by non- MST (that is, STP and RSTP) switches, facilitating the interoperation between MST and its predecessors. In mixed environments with MST and STP/RSTP, STP/RSTP switches unknowingly participate in CST. Costs in CST reflect only the costs of links between regions and in non-MST parts of the network. These costs are called external costs by MST.

Task: Configure MST path selection with port cost

```
(config-if)# spanning-tree mst <instance-number> cost <number>
```

Task: Modify MST path selection with port priority

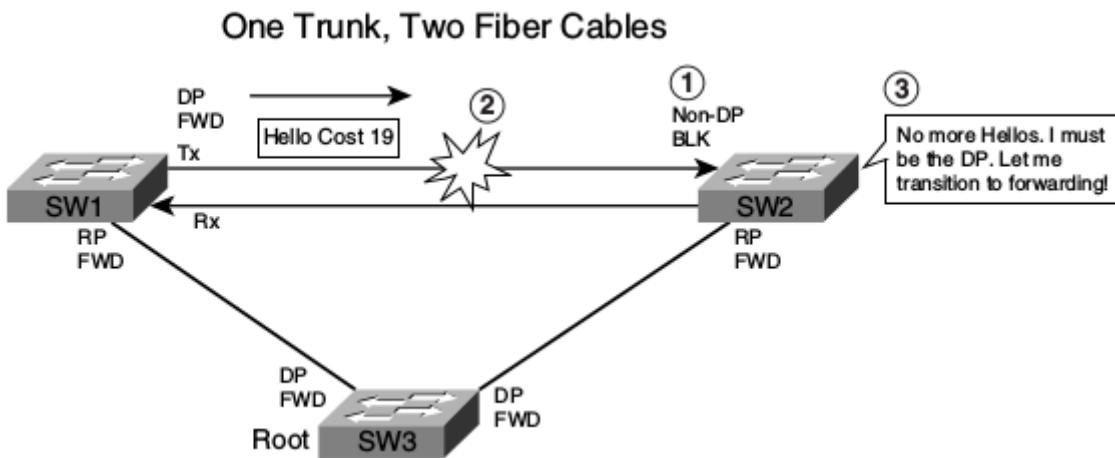
```
(config-if)# spanning-tree mst <instance-number> port-priority <number>
```

4.6. Protecting Against Unidirectional Link Issues

4.6.1. UDLD

Catalyst 3560 Configuration Guides › [UDLD](#)

- Problem: unidirectional links
 - one of the 2 transmission paths has failed but not both
 - due to miscabling, cutting on fiber cable, unplugging one fiber, GBIC problems, ...
 - can cause a loop as the previously blocking port will move to a forwarding state



- solutions:

UDLD UniDirectional Link Detection

- Uses Layer 2 messaging to decide when a switch can no longer receive frames from a neighbor. The switch whose transmit interface did not fail is placed into an err-disabled state.
- Cisco Proprietary
- Multicast Address 01:00:0C:CC:CC:CC

Operations

- Each UDLD port sends protocol messages that contain the own device/port ID and the neighbor's IDs seen by UDLD.
- If the port doesn't see its own device/port ID in the incoming UDLD packet for a specific amount of time, the link is considered unidirectional.
- It is recommended to keep $T_{(\text{det}\text{ection})} < T_{(\text{reconver}\text{gence})}$ by choosing an appropriate message interval which ensures that UDLD is detected before STP forward delay expires

Normal Mode

- default
- marks as **Undertermined** if port at Layer 1 is still up
- does NOT shutdown or disable the port

- does NOT prevent physical loops (informational and less disruptive)

Aggressive Mode

- Attempts to reconnect with the other switch (eight times) after realizing no messages have been received.
- If the other switch does not reply to the repeated additional messages, both sides become err-disabled.
- no automatic recovery unless UDLD err-disable recovery is configured

Default Configuration

Feature	Default Setting
UDLD global enable state	Globally disabled
UDLD per-port enable state for fiber-optic media	Disabled on all Ethernet fiber-optic ports
UDLD per-port enable state for UTP copper media	Disabled on all Ethernet 10/100/1000BASE-TX ports
UDLD aggressive mode	Disabled

Task: Enable UDLD Globally

```
(config)# udld {aggressive | enable | message time <seconds>}
```

Task: Configure the period of time between UDLD probe messages

```
(config)# udld message time <seconds>}
```

- configure the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional.
- range: 1 to 90 seconds
- default: 15 seconds
- This command affects fiber-optic ports only. Use **(config-if)# udld** to enable UDLD on other port types.

Task: Enable UDLD on an Interface

```
(config-if)# udld
```

Task: Reset an Interface Disabled by UDLD

```
# udld reset
```

You can also restart the disabled port

- **shutdown** followed by **no shutdown**
- **no udld {aggressive | enable}** followed by **udld {aggressive | enable}**
- **no udld port** followed by **udld port [aggressive]**

UDLD Error-Disabled State

Task: Recover from the UDLD Error-Disabled State

```
! Enable UDLD to automatically recover
(config)# errdisable recovery cause udld

! Specify the time to recover from the UDLD error-disabled state
(config-if)# errdisable recovery interval <seconds>
```

Task: Display UDLD Status

```
# show udld [interface-id]
```

UDLD vs Loop Guard

- complementary and can be both configured at the same time

In certain designs there are unidirectional links that Loop Guard can prevent and UDLD cannot, and likewise ones that UDLD can prevent but Loop Guard cannot. For example, if a loop occurs because of a physical wiring problem (for example, someone mistakenly mixes up the send and receive pairs of a fiber link), UDLD can detect this, but Loop Guard cannot. Likewise, if there is a unidirectional link caused by a failure in the STP software itself, although much more rare, Loop Guard can detect this but UDLD cannot.

4.6.2. Bridge Assurance

The Bridge Assurance, applicable only with RPVST+ and MST and only on point-to-point links, is a further extension of the idea used by Loop Guard. Bridge Assurance modifies the rules for sending BPDUs. With Bridge Assurance activated on a port, this port always sends BPDUs each Hello interval, whether it is Root, Designated, Alternate, or Backup. BPDUs thus essentially become a Hello mechanism between pairs of interconnected switches. A Bridge Assurance-protected port is absolutely required to receive BPDUs. If no BPDUs are received, the port will be put into a *BA-inconsistent blocking* state until it starts receiving BPDUs again. Apart from unidirectional links, Bridge Assurance also protects against loops caused by malfunctioning switches that completely stop participating in RPVST+/MST (entirely ceasing to process and send BPDUs) while opening all their ports. At the time of this writing, Bridge Assurance was supported on selected Catalyst 6500 and Nexus 7000 platforms. Configuring it on Catalyst 6500 Series requires activating it both globally using spanning-tree bridge assurance and on ports on STP point-to-point link types toward other switches using the spanning-tree portfast network interface command. The neighboring device must also be configured for Bridge Assurance.

4.6.3. Dispute Mechanism

The Dispute mechanism is yet another and standardized means to detect a unidirectional link. Its functionality is based on the information encoded in the Flags field of RST and MST BPDUs, namely, the role and state of the port forwarding the BPDU. The principle of operation is very simple: If a port receives an inferior BPDU from a port that claims to be Designated Learning or Forwarding, it will itself move to the Discarding state. Cisco has also implemented the Dispute mechanism into its RPVST+. The Dispute mechanism is not available with legacy STP/PVST+, as these STP versions do not encode the port role and state into BPDUs. The Dispute mechanism is an integral part of RSTP/MST and requires no configuration.

4.6.4. Unicast Flooding

4.7. Troubleshooting

4.7.1. Flapping Port That Is Generating BPDUs with the TCN Bit Set

4.8. Alternatives to STP

- THRILL
- FabricPath

4.9. Troubleshooting

Inconsistency Type	Description and Probable Cause of Inconsistency Type
(*TYPE_Inc)	PVST+ BPDUs are received on a non-802.1Q port. Usually caused by interconnecting access and trunk ports.
Port VLAN ID (*PVID_Inc)	PVST+ BPDUs are received in a different VLAN than they were originated in. Usually caused by native VLAN mismatch on a trunk.
PVST Simulation (*PVST_Inc)	PVST+ BPDUs received on an MST boundary port do not meet the PVST Simulation consistency criteria.
Loop (*LOOP_Inc)	A Root or Alternate Port tried to become Designated after BPDUs stopped arriving. Seen only on Loop Guard-protected ports.
Root (*ROOT_Inc)	A port tried to become a Root Port after receiving superior BPDUs. Seen only on Root Guard-protected ports. Also on older switches this state was displayed in place of the PVST_Inc state if PVST Simulation Inconsistency was encountered on a port.
Bridge Assurance (*BA_Inc)	A port stopped receiving BPDUs. Seen only on Bridge Assurance-protected ports

Chapter 5. Link Aggregation

5.1. EtherChannel

- EtherChannel aggregates bandwidth of up to 8 physical links
- Consists of two parts:
 - Port-channel interface: logical interface representing the bundle
 - Member interfaces: physical links part of the bundle
- can be any type of interface: Layer 2 access, trunk, tunnel or layer 3 routed
- Configured as either Layer 2 or Layer 3 interfaces.
- To be part of a PortChannel, both sides must agree on:
 - Same speed and duplex settings
 - If not trunking, same access VLAN
 - If trunking, same trunk encapsulation, allowed VLANs, native VLAN, DTP mode
 - On a single switch, each port in a PortChannel must have the same STP cost per VLAN on all links in the PortChannel
 - No ports with SPAN configured
 - If Layer 3, IP address on port-channel and not the physical interface
- When several EtherChannel bundles exist between two switches, STP blocks one of the bundles to prevent redundant links. When spanning tree blocks one of the redundant links, it blocks one EtherChannel, thus blocking all the ports belonging to this EtherChannel link.
- Where there is only one EtherChannel link, all physical links in the EtherChannel are active because STP sees only one (logical) link.
- If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.
- Each EtherChannel has a logical port-channel interface numbered from 1 to 64. The channel groups are also numbered from 1 to 64.
- When a port joins an EtherChannel, the physical interface for that port is shut down.
- When the port leaves the port-channel, its physical interface is brought up, and it has the same configuration as it had before joining the EtherChannel.

5.1.1. Link Aggregation Protocol

- PAgP
 - Maximum 8 ports
- LACP
 - Maximum 16 ports

- Maximum 8 active ports and 8 standby ports

Task: Verify Which Negotiation Protocol Has Been Used for the EtherChannel

```
# show etherchannel protocol
```

Task: Configure the Link Aggregation Protocol Globally

```
(config-if)# channel-protocol {pagg | lacp}
```



- The **channel-group** interface configuration command can also set the mode for the EtherChannel
- If you set the protocol by using **channel-protocol**, the setting is not overridden by the **channel-group** interface configuration command.

5.1.2. Layer 2 EtherChannels

- Logical interfaces are dynamically created when using **channel-group** command.

Task: Configure Layer 2 EtherChannels

```
conf t
interface <type slot/number>
  switchport mode {access | trunk}
  channel-group <n> mode {active | passive | on | {auto [non-silent] | desirable [non-silent]} }
```

5.1.3. Layer 3 EtherChannels

Task: Create the Port Channel Logical Interface

```
conf t
interface port-channel <number>
  no switchport
  ip address <a.b.c.d> <mask>
```

Task: Assign the Physical Interfaces to the Layer 3 Port Channel

```
conf t
interface <type id>
  no switchport
  no ip address
  channel-group <n> mode {active | passive | on | {auto [non-silent] | desirable [non-silent]} }
```



- Always issue the **no switchport** command before the **channel-group** command
- If L3 port-channel configured properly, the **show etherchannel summary** command should show **RU** for routed and in use

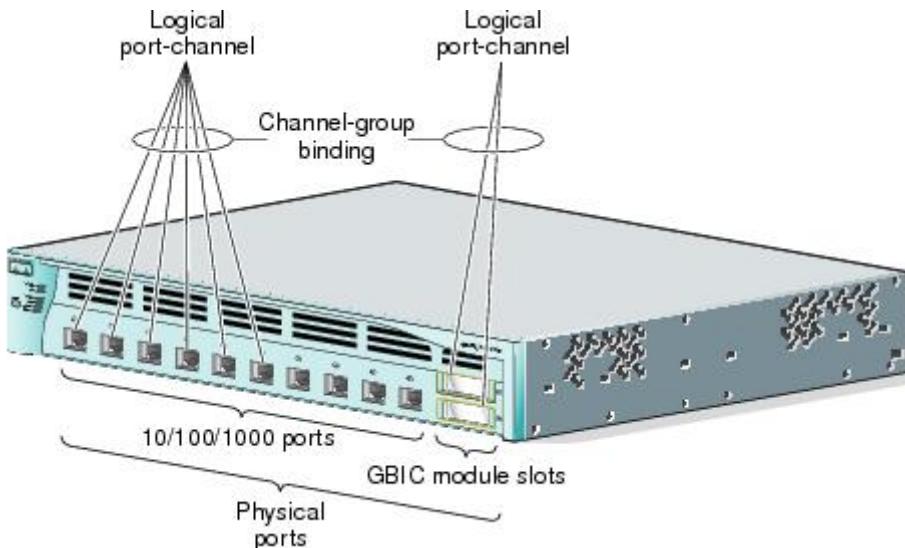


Figure 12. Relationship Of Physical Ports, Logical Port Channels, and Channel Groups

5.1.4. EtherChannel Modes

Table 9. EtherChannel Modes

Cisco PAgP	802.1AD LACP	Description
on	on	disable negotiation and forces the port into the portChannel
off	off	disable negotiation and prevents the ports to be part of the portChannel
desirable	active	initiates the negotiation
auto	passive	waits on other side to start negotiation

Task: Display EtherChannel Status

```
# show etherchannel [group-number]
```

PAgP and LACP Interaction with Other Features

- DTP and CDP send and receive packets over the physical interfaces in the EtherChannel.
- PAgP and LACP transmit PDUs on the lowest numbered VLAN on the interfaces enable for (desirable,auto or active,passive)
- STP sends packets over the first interface in the Etherchannel
- The MAC address of a Layer 3 EtherChannel is the MAC address of the first interface in the port-channel.

Load Balancing and Forwarding Modes

- Load balancing between member interface based on a combination of
 - Source MAC address
 - Destination MAC address
 - Source IP address
 - Destination IP address
- Uses only source MAC address by default

Task: Configure the EtherChannel Load-Balancing Method

```
(config)# port-channel load-balance { dst-ip | dst-mac | src-dst-ip | src-dst-mac |  
src-ip | src-mac}
```

Task: Display the EtherChannel Load-Balancing Method

```
# show etherchannel load-balance  
  
EtherChannel Load-Balancing Configuration:  
src-mac  
  
EtherChannel Load-Balancing Addresses Used Per-Protocol:  
Non-IP: Source MAC address  
IPv4: Source MAC address  
IPv6: Source MAC address
```

5.1.5. EtherChannel Misconfiguration Guard

- This mechanism makes an assumption that if multiple ports are correctly bundled into a Port-channel at the neighbor side, all BPDUs received over links in this Port-channel must have the same source MAC address in their Ethernet header, as the Port-channel interface inherits the MAC address of one of its physical member ports. If BPDUs sourced from different MAC addresses are received on a Port-channel interface, it is an indication that the neighbor is still treating the links as individual, and the entire Port-channel will be err-disabled
- Enabled by default

Task: Deactivate EtherChannel Misconfig Guard

```
(config)# no spanning-tree etherchannel guard misconfig
```

5.1.6. Vlan internal allocation policy

Task: manage the vlan internal allocation policy

```
(config)# vlan internal allocation policy ascending
```

5.2. LACP

- IEEE 802.1ax, formerly IEEE 802.3ad
- Automatic creation of port channels
- Multicast address IEEE 802.3 Slow Protocols: 0180-C200-0002
- EtherType value: 0x8809
- Timers: hellos every second during hand shake
- Maximum: 16 ports with max 8 active

5.2.1. Restrictions

TODO

5.2.2. Modes

Passive

- Does not initiate LACP negotiation but responds to LACP packets
- Default mode

Active

- Initiate LACP negotiation by sending LACP packets

On

- Forces the interface to the channel without PAgP or LACP

Working Etherchannel for On-On, Passive-Active, Active-Active

5.2.3. LACP Hot-Standby Ports

- Only 8 LACP links can be active at one time
- Any additional links are in hot-standby mode
- If one of the active links becomes inactive, a hot-standby link becomes active in its place
- Each link is assigned a unique priority in this order
 - LACP system priority (1..65535, default: 32768)
 - System ID (the switch MAC address)
 - LACP port priority
 - Port number
- In priority comparisons, lower values have higher priority.
- To determine which ports are active and which ports are hot standby,
 - Select the master switch with a low system priority and system-id
 - Select the master ports with the low port priority and number. The port-priority and port-

number of the slave switch are not used.

Task: Check Which Ports Are In the Hot-Standby Mode

```
# show etherchannel summary
```

Task: Configure the LACP System Priority

```
(config)# lacp system-priority <priority>
```

Task: Show the LACP System Priority

```
# show lacp sys-id
```

Task: LACP Port Priority

```
(config-if)# lacp port-priority
```

5.2.4. LACP Port-Channel MaxBundle Feature

- Control the number of ports allowed to be bundled into the etherchannel
- Allows hot-standby ports with fewer bundled ports

Task: Configure the Maximum Number Of Bundled Ports Allowed In a LACP Port Channel

```
(config-if)# lacp max-bundle
```

5.2.5. LACP Port-Channel Min-Links Feature

- Only for LACP Etherchannel
- Prevents low-bandwidth interface from becoming active
- Causes LACP etherChannels to become inactive if they have too few active members ports to supply the required minimum bandwidth

Task: Configure the Minimum Number Of Member Ports That Must Be In the Link-Up State and Bundled In the Etherchannel for the Port Channel Interface to Transition to the Link-Up State

```
(config-if)# port-channel min-link n
```

5.3. PAgP

- Port Aggregation Protocol
- Cisco proprietary
- Automatic creation of a EtherChannel.

- Sends PAgP packets every 30 seconds to multicast 0100-0CCC-CCCC
- Same destination address than CDP, UDLD, VTP, and DTP.
- Checks for configuration consistency and manages link additions and failures between two switches.
- Protocol value: 0x104
- Cannot be enabled on cross-stack EtherChannel

Task: Display PAgP Status

```
# show pagp [channel-group-number]
```

5.3.1. Modes

Auto

- Never initiates PAgP communications but instead listen passively for any received PAgP packets before creating an EtherChannel with the neighboring switch.
- Default mode

Desirable

- Initiates negotiations with other interfaces by sending PAgP packets.

On

- Forces the interface to channel without PAgP.
- Do not exchange PAgP packets.

Etherchannel formed for on-on, desirable-auto, desirable-desirable combinations.

5.3.2. Physical Vs Aggregate Learners

Switches running PAgP are classified as:

PAgP physical learners

- learn MAC addresses using the physical ports within the EtherChannel instead of via the logical EtherChannel link.
- forward traffic to addresses based on the physical port via which the address was learned. The switch will send packets to the neighboring switch using the same port in the EtherChannel from which it learned the source address.

Aggregate learners

- learns addresses based on the aggregate or logical EtherChannel port.
- transmit packets to the source by using any of the interfaces in the EtherChannel.
- Aggregate learning is the default.

By default, PAgP is not able to detect whether a neighboring switch is a physical learner. Therefore, when configuring PAgP EtherChannels on switches that support only physical learning, the learning

method must be manually set to physical learning. It is important when running in this mode, to set the load-distribution method to source-based distribution so that any given source MAC address is always sent on the same physical port.

Task: Configure the PAgP Learning Method

```
(config-if)# pagp learn-method {physical-port | aggregation-port}
```

Task: Verify the PAgP Learning Method

```
# show pagp [channel-group-number] internal
```

5.3.3. Priority

- Range: 0..255
- Default: 128
- The higher the priority, the more likely that the port will be used for PAgP transmission

Task: Assign a Priority So That the Selected Port Is Chosen for Packet Transmission.

```
(config-if)# pagp port-priority <priority>
```

5.3.4. Restrictions

While PAgP allows for all links within the EtherChannel to be used to forward and receive user traffic, there are some restrictions:

- DTP and CDP send and receive packets over all the physical interfaces in the EtherChannel, while PAgP sends and receives PAgP PDU only from interfaces that are up and have PAgP enabled for auto or desirable modes.
- When an EtherChannel bundle is configured as a trunk port, the trunk sends and receives PAgP frames on the lowest numbered VLAN. STP always chooses the first operational port in an EtherChannel bundle.
- When configuring additional STP features such as Loop Guard on an EtherChannel, remember that if Loop Guard blocks the first port, no BPDUs will be sent over the channel, even if other ports in the channel bundle are operational. This is because PAgP will enforce uniform Loop Guard configuration on all of the ports that are part of the EtherChannel group.

Task: Validate the Port That Will Be Used by STP to Send Packets and Receive Packets

```
Switch#show pagp neighbor
```

Flags: S – Device is sending Slow hello. C – Device is in Consistent state.

A – Device is in Auto mode. P – Device learns on physical port.

Channel group 4 neighbors

Partner	Partner	Partner	Partner	Group		
Port	Name	Device ID	Port	Age	Flags	Cap.
Gi1/1/3	Switch.1	00c5.a003.0080	Gi0/1	4s	SC	10001
Gi1/1/4	Switch.1	00c5.a003.0080	Gi0/2	3s	SC	10001

STP will send packets only out of port Gi1/1/3 because it is the first operational interface. If that port fails, STP will send packets out of Gi1/1/4.

5.3.5. Silent Mode

Task: Configure a Switch Port for Nonsilent Operation

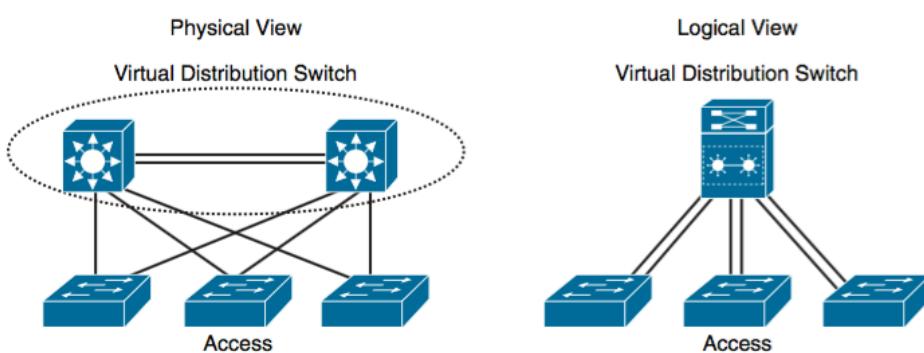
TODO

Task: Configure a Switch Port for Nonsilent Operation

TODO

TODO You can also configure a single interface within the group for all transmissions and use other interfaces for hot standby. The unused interfaces in the group can be swapped into operation in just a few seconds if the selected single interface loses hardware-signal detection.

5.4. Virtual Switch System



- VSS makes two physical switches to act and appear as one single logical network element.
- VSS manages the redundant links from access switches as single Multi-chassis Etherchannel
 - No need for spanning-tree to block one of the links
 - two active links instead of one 1/10/40b interfaces
- on Cat6500, Cat4500 running IOS-XE

5.4.1. VSS Active and Standby Switch

- Uses VLSP to negotiate the active and standby roles at start
- The VSS active switch
 - controls the VSS, running the Layer 2 and Layer 3 control protocols for the switching modules on both switches.
 - provides management functions for the VSS, such as module online insertion and removal (OIR) and the console interface.
- The VSS active and standby switches perform packet forwarding for ingress data traffic on their locally hosted interfaces. However, the VSS standby switch sends all control traffic to the VSS active switch for processing

Task: Configure VSS Domain Number and Switch Number

```
(config)# switch virtual domain <1..255>
(config-vs-domain)# switch [1 | 2]
```

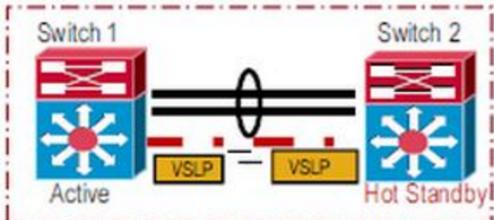
Task: Configure VSS Switch Priority

```
(config-vs-domain)# switch [1 | 2] priority [<number>]
```



1 lowest priority 255 highest priority 100 default

5.4.2. Virtual Switch Link



- Normally built as etherchannel with up to 8 links
- carries system control information (hot-standby supervisor programming, line card status, Distributed Forwarding Card (DFC) card programming, system management, diagnostics, ...)
- carries user data traffic when necessary

Task: Create VSL

```
(config)#interface port channel 5
(config-if)# switchport
(config-if)# switch virtual link 1
(config-if)# no shut
(config-if)# exit
#! add physical interface to port channel
(config)# interface range gi 7/4 - 5
(config-if)# channel group 5 mode one
(config-if)# exit
```

Task: Convert the Switch to Virtual

```
(config)# switch convert mode virtual
```

Task: Displays the VSS Information

```
# sh switch virtual [role | link]
```

Example

Executing the command on VSS member switch role = VSS Active, id = 1
RRP information for Instance 1

Valid Flags Peer Preferred Reserved
Count Peer Peer

TRUE V 1 1 1

Switch Number	Switch Status	Preempt Oper(Conf)	Priority Oper(Conf)	Role	Local SID	Remote SID
LOCAL 1	UP	FALSE(N)	100(100)	ACTIVE	0	0
REMOTE 2	UP	FALSE(N)	100(100)	STANDBY	6834	6152

Peer 0 represents the local switch

Flags : V - Valid

In dual-active recovery mode: No



Executing the command on VSS member switch role = VSS Standby, id = 2
RRP information for Instance 2

Valid Flags Peer Preferred Reserved
Count Peer Peer

TRUE V 1 1 1

Switch Number	Switch Status	Preempt Oper(Conf)	Priority Oper(Conf)	Role	Local SID	Remote SID
LOCAL 2	UP	FALSE(N)	100(100)	STANDBY	0	0
REMOTE 1	UP	FALSE(N)	100(100)	ACTIVE	6152	6834

Peer 0 represents the local switch

Flags : V - Valid

In dual-active recovery mode: No

Task: Displays the VSL Information

```
# sh switch virtual link
```

Example



```
Executing the command on VSS member switch role = VSS Active, id = 1
VSL Status : UP
VSL Uptime : 3 minutes
VSL Control Link : Gi1/7/4
Executing the command on VSS member switch role = VSS Standby, id = 2
VSL Status : UP
VSL Uptime : 3 minutes
VSL Control Link : Gi2/4/45
```

5.4.3. Multi-chassis Ethernet Channel

- VSS enables the creation of Multichassis EtherChannel (MEC), which is an EtherChannel whose member ports can be distributed across the member switches in a VSS.
- Because non-VSS switches connected to a VSS view the MEC as a standard EtherChannel, non-VSS switches can connect in a dual-homed manner.
- Traffic traversing the MEC can be load balanced locally within a VSS member switch much like that of standard EtherChannels.
- Cisco MEC supports dynamic configuration (LACP and PAgP) as well as static EtherChannel configuration.
- In total, a VSS can support a maximum of 256 EtherChannels. This limit applies to the total number of regular EtherChannels and MECs.

5.4.4. vPC

TODO

5.4.5. Stackwise

TODO

Cisco StackWise technology provides a new method for collectively utilizing the capabilities of a stack of switches. Individual switches intelligently join to create a single switching unit with a 32-Gbps switching stack interconnect. Configuration and routing information is shared by every switch in the stack, creating a single switching unit. Switches can be added to and deleted from a working switch stack without affecting performance.

Table 10. Rules and their respective priority order

Priority	Rule
1	The switch that is currently the stack master
2	The switch that is currently the stack master

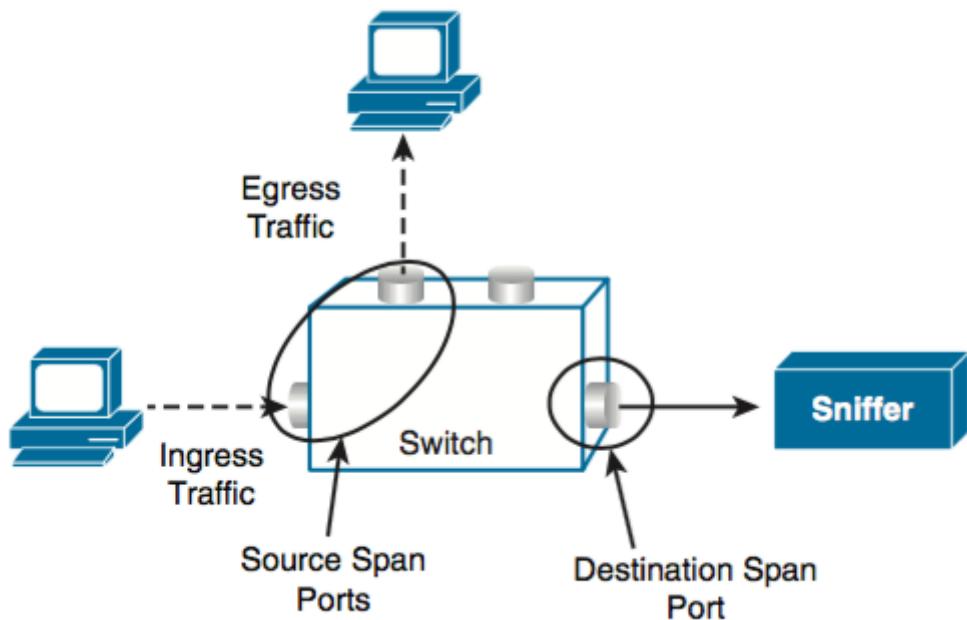
Priority	Rule
3	The switch that uses the non-default interface-level configuration
4	The switch with the higher Hardware/Software priority. These switch software versions are listed from highest to lowest priority : 1. Cryptographic IP services image software 2. Noncryptographic IP services image software 3. Cryptographic IP base image software 4. Noncryptographic IP base image software
5	The switch with the longest system up-time
6	The switch with the lowest MAC address

Chapter 6. SPAN , RSPAN and ERSPAN

- SPAN (Switch Port Analyzer) mirrors monitored (TX,RX or Both) traffic on source ports or VLANs to a destination port for analysis.

6.1. Local SPAN Sessions

- Source and destination on the same switch



Task: Display SPAN Status

```
Switch# show monitor session
```

Session 1

=====

```
Type : Local Session
Source Ports :
    RX Only : None
    TX Only : None
    Both    : Fa0/4
Source VLANs :
    RX Only : None
    TX Only : None
    Both    : None
Source RSPAN VLAN : None
Destination Ports : Fa0/5
    Encapsulation: DOT1Q
        Ingress: Enabled, default VLAN = 5
Reflector Port : None
Filter VLANs   : None
Dest RSPAN VLAN : None
```

6.1.1. Source

- Can not mix source ports and source VLANs in a single session
- Monitored traffic directions can be
 - Rx : before any modification or processing by ACL or QoS or VACL
 - Tx : after all modification and processing performed by the switch.
 - Both: by default

Task: Configure Span Source Ports/VLANs

```
monitor session <1-66> source {interface <id> | vlan <id>} [, | -] [both | rx | tx]
```

- (Optional) [, | -] Specify a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
- A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session.



Source Ports

- Can be physical interfaces
- Can be port-channel logical interfaces with port-channel numbers in (1..48)

- Can be an access port, trunk port, routed port, or voice VLAN port.
- Cannot be a destination port

Source VLANs

- All active ports in the source VLAN are included as source ports and can be monitored in either or both directions.
- On a given port, only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.
- Ignores CDP, BPDU, VTP, DTP and PAgP frames unless **encapsulation replicate** is configured

6.1.2. Destination Port

- Must be a physical port
- Cannot be a source port
- By default, send packets untagged
 - can replicate the source interface encapsulation
- By default, disable the ingress traffic forwarding
 - can accept incoming packets with dot1q, ISL or untagged
- Only one SPAN/RSPAN session can send traffic to a single destination port, cannot be used by two SPAN sessions
- Only monitored traffic passes through the SPAN destination port
- Entering SPAN configuration commands does not remove previously configured SPAN parameters. Enter the **no monitor session {session_number | all | local | remote}** global configuration command to delete configured SPAN parameters.
- For local SPAN, outgoing packets through the SPAN destination port carry the original encapsulation headers—untagged, ISL, or IEEE 802.1Q If the encapsulation replicate keywords are specified. If the keywords are not specified, the packets are sent in native form. For RSPAN destination ports, outgoing packets are not tagged.
- You can configure a disabled port to be a source or destination port, but the SPAN function does not start until the destination port and at least one source port or source VLAN are enabled.
- You cannot mix source VLANs and filter VLANs within a single SPAN session.
- Up to 64 SPAN destination ports can be configured on a switch

Task: Configure the Destination Port for a SPAN Session

```
(config)# monitor <session-number>
    destination interface <interface-id>
    [encapsulation replicate]
    [ingress {dot1q vlan <vlan-id> | isl | untagged vlan <vlan-id>} ]
```

6.1.3. VLAN Filtering

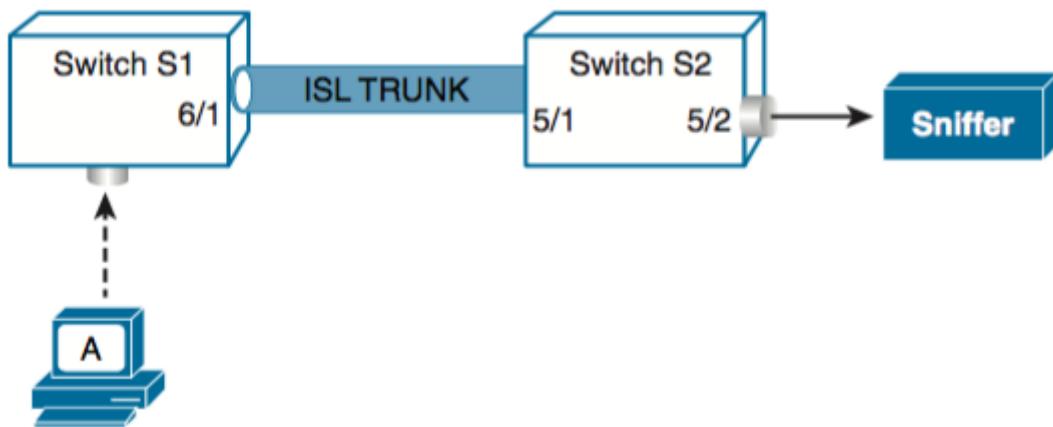
- To limit SPAN traffic monitoring on trunk source ports to specific VLANs
- Applies only to trunk ports or to voice VLAN ports.
- Applies only to port-based sessions
- Not allowed in sessions with VLAN sources.
- When a VLAN filter list is specified, only those VLANs in the list are monitored on trunk ports or on voice VLAN access ports.
- SPAN traffic coming from other port types is not affected by VLAN filtering; that is, all VLANs are allowed on other ports.
- VLAN filtering affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic.

Task: Limit SPAN Source to Specific VLANs

```
(config)# monitor <session-number> filter vlan <vlan-ids>
```

6.2. Remote SPAN Sessions

RSPAN consists of at least one RSPAN source session, an RSPAN VLAN, and at least one RSPAN destination session.



Restrictions and Considerations

When RSPAN is enabled, each packet being monitored is transmitted twice, once as normal traffic and once as a monitored packet. Therefore monitoring a large number of ports or VLANs could

potentially generate large amounts of network traffic.

6.2.1. RSPAN VLAN

- Can be propagated to all switches by VTP if $1 < \text{RSPAN VLAN} < 1002$
- Must be created manually on extended-range VLAN
- Can not be vlan 1, 1002-1005
- Can serve multiple RSPAN source/destination sessions

Restrictions

- You can apply an output ACL to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source switches.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple switches in your network.
- RSPAN does not support BPDU packet monitoring or other Layer 2 switch protocols.
- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that the VLAN remote-span feature is supported in all the participating switches.
- Access ports (including voice VLAN ports) on the RSPAN VLAN are put in the inactive state.
- RSPAN VLANs are included as sources for port-based RSPAN sessions when source trunk ports have active RSPAN VLANs. RSPAN VLANs can also be sources in SPAN sessions. However, since the switch does not monitor spanned traffic, it does not support egress spanning of packets on any RSPAN VLAN identified as the destination of an RSPAN source session on the switch.

Task: Configure RSPAN VLAN on All Participating Switches

```
(config)# vlan <rspan-vlan-id>
(config-vlan)# remote-span
```

6.2.2. RSPAN Source Session

- Must be configured on the monitored port's switch

Task: Configure the RSPAN Source Session

```
monitor session <session-number> source {interface <interface-id> | vlan <vlan-id>} [, | -] [both | rx | tx]
monitor session <session-number> destination remote vlan <rspan-vlan-id>
```

6.2.3. RSPAN Destination Session

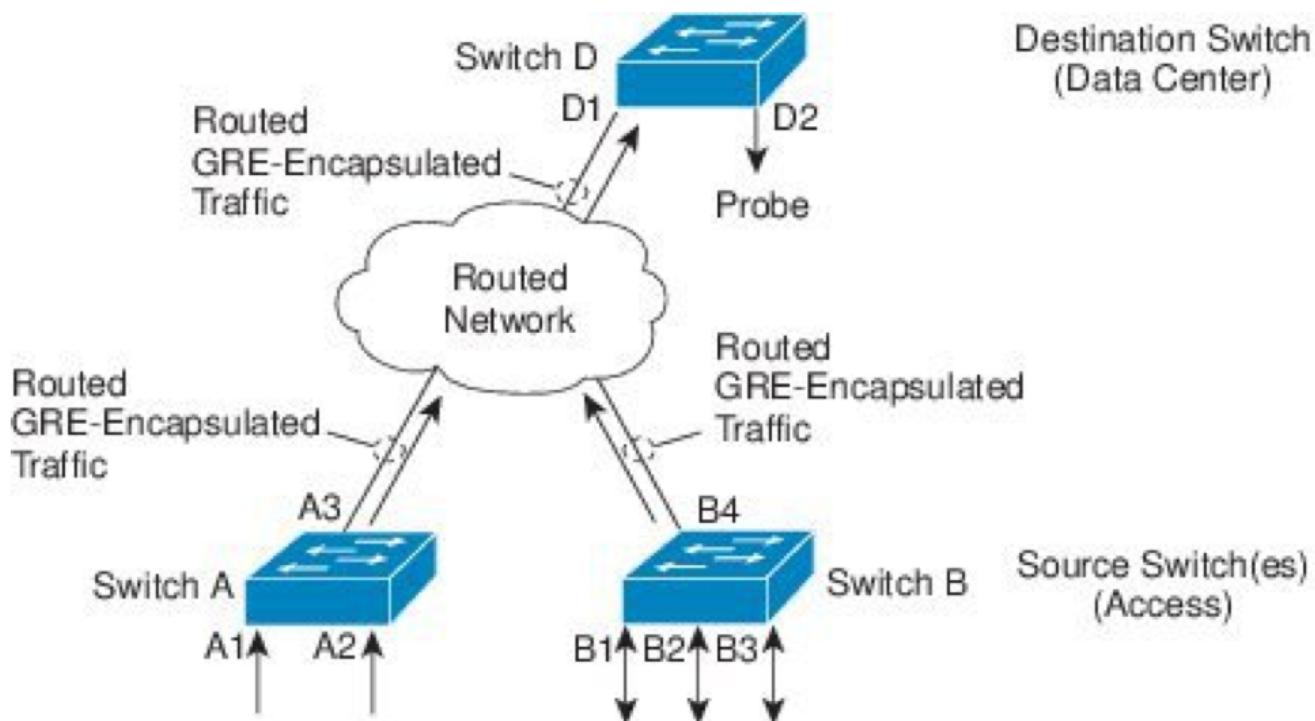
- Takes all packets received on the RSPAN VLAN, strips off the VLAN tagging, and presents them on the destination port.
- Excludes Layer 2 control

Task: Configure the RSPAN Destination Session on a Different Switch (Not the Switch on Which the Source Session Was Configured)

```
(config)# monitor session <session-number> source remote vlan <rspan-vlan-id>
(config)# monitor session <session-number> destination interface <interface-id>
```

6.3. Encapsulated RSPAN

- ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE encapsulated traffic, and an ERSPAN destination session.
- Supported only on high-end switches (ASR1000, Catalyst 6500/7600, Nexus platforms) or IOS-XE



6.3.1. ERSPAN Source Session

Task: Configure ERSPAN Source Session

```
(config)# monitor session <id> type erspan-source
(config-mon-erspan-src)# source { interface <interface-id> | vlan <vlan-ids>
[rx|tx|both]}
(config-mon-erspan-src)# destination
(config-mon-erspan-src-dst)# erspan-id <erspan-flow-id>
(config-mon-erspan-src-dst)# mtu <size>
(config-mon-erspan-src-dst)# origin ip address <a.b.c.d> [force]
(config-mon-erspan-src-dst)# no shutdown
```

6.3.2. ERSPAN Destination Session

Task: Configure ERSPAN Destination Session

```
(config)# monitor session <id> type erspan-destination
(config-mon-erspan-dst)# destination interface <interface-id>
(config-mon-erspan-dst)# source
(config-mon-erspan-dst-src)# erspan-id <erspan-flow-id>
(config-mon-erspan-dst-src)# mtu <size>
(config-mon-erspan-dst-src)# ip address <a.b.c.d> [force]
(config-mon-erspan-dst-src)# no shutdown
```

6.3.3. ESPAN Dummy MAC Address Rewrite

- Supports customized MAC value for WAN interface and tunnel interface
- Monitor the traffic going through WAN interface

Task: Configure ESPAN Dummy MAC Address

```
(config)# monitor session <session-id> type erspan-source
(config-mon-erspan-src-dst)# s-mac <mac-address>
(config-mon-erspan-src-dst)# d-mac <mac-address>
```

6.4. Interaction with Other Features

Routing

- SPAN does not monitor routed traffic.
- RSPAN only monitors traffic that enters or exits the switch, not traffic that is routed between VLANs.

STP

- A destination port does not participate in STP while its SPAN or RSPAN session is active.
- The destination port can participate in STP after the SPAN or RSPAN session is disabled.
- On a source port, SPAN does not affect the STP status. STP can be active on trunk ports carrying an RSPAN VLAN.

CDP

- A SPAN destination port does not participate in CDP while the SPAN session is active.
- After the SPAN session is disabled, the port again participates in CDP.

VTP

- You can use VTP to prune an RSPAN VLAN between switches.

VLAN and trunking

- You can modify VLAN membership or trunk settings for source or destination ports at any

time.

- However, changes in VLAN membership or trunk settings for a destination port do not take effect until you remove the SPAN destination configuration.
- Changes in VLAN membership or trunk settings for a source port immediately take effect, and the respective SPAN sessions automatically adjust accordingly.

EtherChannel

- You can configure an EtherChannel group as a source port but not as a SPAN destination port.
- When a group is configured as a SPAN source, the entire group is monitored.
- If a physical port is added to a monitored EtherChannel group, the new port is added to the SPAN source port list.
- If a port is removed from a monitored EtherChannel group, it is automatically removed from the source port list.
- A physical port that belongs to an EtherChannel group can be configured as a SPAN source port and still be a part of the EtherChannel.
- In this case, data from the physical port is monitored as it participates in the EtherChannel. However, if a physical port that belongs to an EtherChannel group is configured as a SPAN destination, it is removed from the group. After the port is removed from the SPAN session, it rejoins the EtherChannel group. Ports removed from an EtherChannel group remain members of the group, but they are in the inactive or suspended state.
- If a physical port that belongs to an EtherChannel group is a destination port and the EtherChannel group is a source, the port is removed from the EtherChannel group and from the list of monitored ports.

Multicasting

- Multicast traffic can be monitored.
- For egress and ingress port monitoring, only a single unedited packet is sent to the SPAN destination port.
- It does not reflect the number of times the multicast packet is sent.

Private VLAN

- A private-VLAN port cannot be a SPAN destination port.

Secure port

- A secure port cannot be a SPAN destination port.
- For SPAN sessions, do not enable port security on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable port security on any ports with monitored egress.
- An IEEE 802.1x port can be a SPAN source port. You can enable IEEE 802.1x on a port that is a SPAN destination port; however, IEEE 802.1x is disabled until the port is removed as a SPAN destination.
- For SPAN sessions, do not enable IEEE 802.1x on ports with monitored egress when ingress

forwarding is enabled on the destination port. For RSPAN source sessions, do not enable IEEE 802.1x on any ports that are egress monitored.

Chapter 7. Port-Based Traffic Control

7.1. Storm Control

- limits the amount of broadcast, multicast or unicast traffic received inbound on a physical interface.
- blocks the port when the rising threshold is reached
- resumes normal forwards when the traffic rate drops below falling threshold
- the threshold can be expressed in
 - pps (packet per second)
 - bps (bits per seconds)
 - pps for small frames (< 67 bytes)
 - percentage of the available bandwidth of the port
- by default, filters outs the traffic and err-disable the port
- can be configured to send a SNMP trap

Task: Configure storm control

```
(config-if)# storm-control {broadcast | multicast | unicast}
          level {<rising-threshold> [<falling-threshold>] | bps <bps>
[<bps-low>] | pps <pps> [<pps-low>] }
```

Task: Limit traffic in packet per second

```
(config-if)# storm-control {broadcast | multicast | unicast} level pps <rising-
threshold> [<falling-threshold> ] }
```

Task: Limit traffic in bits per second

```
(config-if)# storm-control {broadcast | multicast | unicast} level bps <rising-
threshold> [<falling-threshold> ] }
```

Task: Limit traffic in as a percentage of the bandwidth

```
(config-if)# storm-control {broadcast | multicast | unicast} level <rising-threshold>
[<falling-threshold> ] }
```

Task: Err-disable a port during a storm

```
(config-if)# storm-control action shutdown
```

Task: Generate an SNMP trap when a storm is detected

```
(config-if)# storm-control action trap
```

Task: Verify the storm control suppression levels set on the interface

```
sh storm-control [<interface-id>] [broadcast|multicast|unicast]
```

7.2. Protocol Storm Protection

TODO https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_1_se/configuration/guide/3750xcg/swtrafc.html#71262

7.3. Protected Port

TODO https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_1_se/configuration/guide/3750xcg/swtrafc.html#71262

7.4. Port Blocking

- blocks a port from flooding unknown unicast or multicast to other ports

Task: Block unknown multicast forwarding out of the port

```
(config-if)# switchport block multicast
```

-  • Only pure L2 multicast traffic is blocked
• multicast packets that contain IPv4 or IPv6 information in the header are not blocked

Task: Block unknown unicast forwarding out of the port

```
(config-if)# switchport block unicast
```

7.5. Port Security

TODO https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_1_se/configuration/guide/3750xcg/swtrafc.html#71262

Chapter 8. WAN

8.1. HDLC

- High-Level Data Link Control
- Layer 2 on point-to-point links
- bit oriented synchronous, without retransmission
- uses SLARP to send keepalives
- developed by the ISO (ISO 3309)
- modified by Cisco by adding a proprietary 2-byte Type field to the frame.
 - enabled by default on IOS serial links
 - On a Cisco router, HDLC encapsulation can only connect with another Cisco router

8.1.1. HDLC Frame Format

TODO image::ppp-vs-hdlc.png[PPP vs HDLC]

8.1.2. Encapsulation

- default

Task: Set Encapsulation to HDLC

```
(config-if)# encapsulation hdlc
```

Task: Display Statistics

```
sh controllers serial
```

8.1.3. Clock Rate

- Automatically set

Task: Modify the Clock Rate

```
(config-if)# clock-rate <bps>
```

Task: Specify the Clock Rate Is In the Network

```
(config-if)# clock-rate line
```

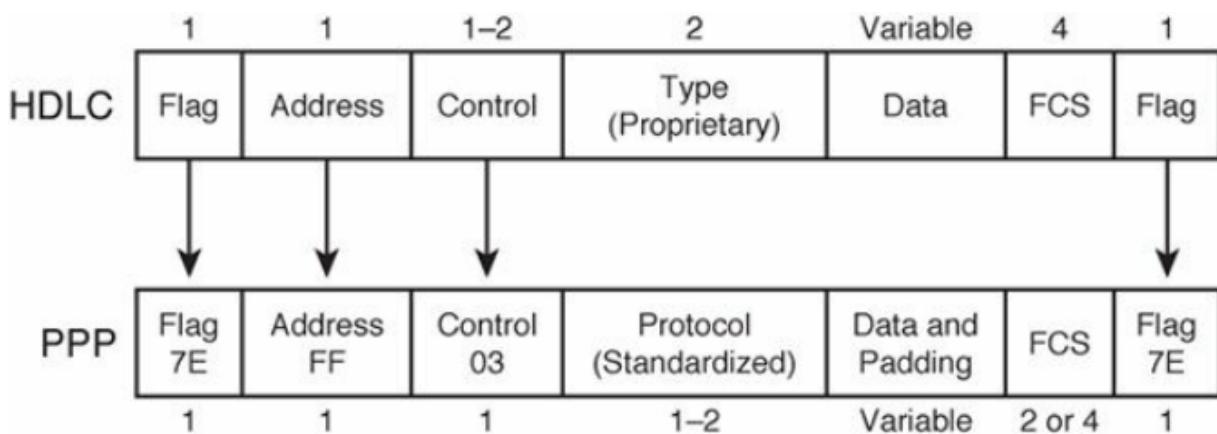
8.2. PPP

Cisco Cloud Routers Configuration guides > WAN > [PPP](#)

- error detection, error recovery, standard protocol Type field, supports synchronous and asynchronous links
- [RFC 1661](#)
- [RFC 1662](#)

8.2.1. PPP Frame Format

- Replaces proprietary type with standard protocol
- Adds padding so the frame has even number of bytes



- FCS:: Frame Checksum calculated over the address, control, protocol, information and payload fields

8.2.2. PPP LCP

- link control protocol
- controls features independent of any Layer 3 protocol
- establishes a PPP session between 2 link partners.

Task: Configure PPP

```
(config-if)# encapsulation ppp
```

LCP Operations:

When a PPP serial link first comes up—for example, when a router senses the Clear to Send (CTS), Data Send Read (DSR), and Data Carrier Detect (DCD) leads come up at the physical layer—LCP begins parameter negotiation with the other end of the link. For example, LCP controls the negotiation of which authentication methods to attempt, and in what order, and then allows the authentication protocol (for example, CHAP) to complete its work. After all LCP negotiation has completed successfully, LCP is considered to be “up.”

LCP Features

- LQM link quality monitoring: drop if % of error frames above a configured value
- looped link detection: drop link if a router receives its own randomly chosen magic number
- layer 2 load balancing: fragment frames over multilink PPP
- authentication: chap, pap

Configuration

- minimal with **encapsulation ppp**
- optional authentication, quality

```
(config-if)# encapsulation ppp  
(config-if)# ppp quality <percent>  
(config-if)# ppp authentication {chap | pap}
```

Task: Drop the Link If Router Receives Its Own Magic Number

LQM

- When LQM is enabled, every keepalive period is sent to Link Quality Reports (LQRs) in place of keepalives. All incoming keepalives are responded to properly.
- If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.
- LQM is incompatible with Multilink PPP

Task: Monitor PPP Link Quality

```
(config-if)# ppp quality <percent>
```

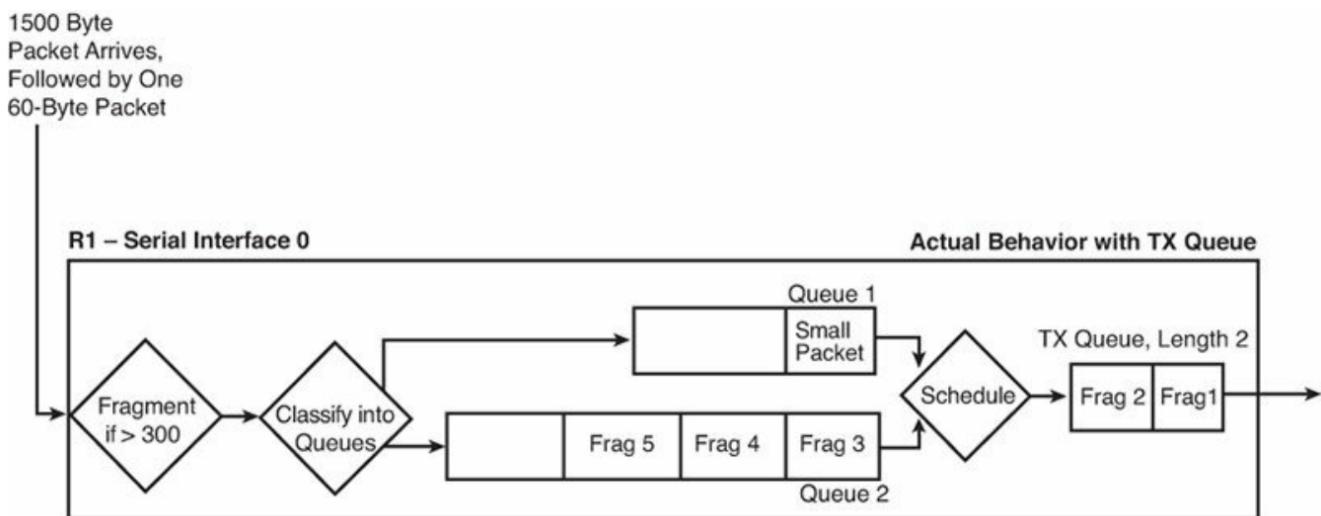
8.2.3. Multilink PPP

- originally intended to combine multiple ISDN B-channels without requiring any Layer 3 load balancing

- now load balance traffic across any type of point-to-point serial link
- configuration with multilink interfaces or virtual templates
- fragmenting each data link layer frame, either based on the number of parallel links or on a configured fragmentation delay.
 - sends the fragments over different links at the same time.
 - adds a header (4 or 2 bytes for Sequence Number and Flags bit) to allow reassembly on the receiving end

LFI

- LFI (link fragmentation and interleaving)
- prevents small, delay sensitive packets from having to wait on longer, delay-insensitive packets to be completely serialized out an interface.
- the queuing scheduler generally LLQ on the multilink interface determines the next packet to send:



Task: Allow Interleave

```
(config-if)# ppp multilink interleave
```

Task: Define LFI Fragment Size

```
(config-if)# ppp multilink fragment-delay <microseconds>
```



defines the fragment size based on "Size" = x * "Bandwidth"

Example

```
interface Multilink1
bandwidth 256
ip address 10.1.34.3 255.255.255.0
encapsulation ppp
ppp multilink
ppp multilink group 1
ppp multilink fragment-delay 10
ppp multilink interleave
service-policy output queue-on-dsc
```

8.2.4. PPP Compression

- uses L2 payload compression (ip + tcp + data + DL) : best with longer packet
- TCP header compression (ip + tcp)
- RTP header compression (ip + udp + rtp)
- payload compression works best with longer packets, and header with shorter packets
- header compression : achieves better compression ration 10:1 to 20:1

Layer 2 Compression

- options: LZS (Lempel-Ziv Stacker), MPPC (microsoft point-to-point compression), Predictor
- LZS use more CPU and less RAM than Predictor algorithm and have better compression ratio
- stacker: supports HDLC, PPP, FR, ATM
- mppc: PPP, ATM
- predictor: PPP, ATM
- configuration with a matching **compress** command under each interface on both end of the links
- once configured, ppp starts ccp (compression control protocol) which is another NCP

Header Compression

- configured with legacy commands or MQC commands
- legacy under the serial (ppp) or multilink interface
- **ip tcp header-compression [passive]**
- **ip rtp header-compression [passive]**
- add also MQC commands

8.2.5. PPP Authentication

Task: Enable PPP Authentication

```
ppp authentication {chap | chap pap | pap chap | pap} [if-needed] [<list-name> |  
default] [callin]
```

Task: Debug Ppp Authentication

```
debug ppp authentication
```

read [understanding debug ppp negotiation](#)

8.2.6. CHAP

Challenge-Handshake Authentication Protocol - RFC 1994

CHAP is an authentication scheme used by Point-to-Point Protocol (PPP) servers to validate the identity of remote clients. CHAP periodically verifies the identity of the client by using a three-way handshake. This happens at the time of establishing the initial link (LCP), and may happen again at any time afterwards. The verification is based on a shared secret (such as the client's password).

1. After the completion of the link establishment phase, the authenticator sends a "challenge" message to the peer.
2. The peer responds with a value calculated using a one-way hash function on the challenge and the secret combined.
3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authenticator acknowledges the authentication; otherwise it should terminate the connection.
4. At random intervals the authenticator sends a new challenge to the peer and repeats steps 1 through 3.

Packets

TODO format

Description	Code (1 byte)	ID (1 byte)	2 bytes	1 byte	Variable	variable
Challenge	1	ID	Length	Challenge Length	Challenge value	Name
Response	2	ID	Length	Response Length	Response value	Name
Success	3	ID	Length		Message	
Failure	4	ID	Length		Message	

8.3. PPPoE

- Ethertype: 0x8863 (Discovery Stage), 0x8864 (PPP Session Stage)
- RFC 2516
- used for digital subscriber line (DSL) Internet access because the public telephone network uses ATM for its transport protocol; therefore, Ethernet frames must be encapsulated in a protocol supported over both Ethernet and ATM.
- The PPP Client feature permits a Cisco IOS router, rather than an endpoint host, to serve as the client in a network. This permits multiple hosts to connect over a single PPPoE connection.
- In a DSL environment, PPP interface IP addresses are derived from an upstream DHCP server using IP Configuration Protocol (IPCP). Therefore, IP address negotiation must be enabled on the router's dialer interface. This is done using the **ip address negotiated** command in the dialer interface configuration.

8.3.1. PPPoE packets



Because PPPoE introduces an 8-byte overhead (2 bytes for the PPP header and 6 bytes for PPPoE), the MTU for PPPoE is usually decreased to 1492 bytes so that the entire encapsulated frame fits within the 1500-byte Ethernet frame. Additionally, for TCP sessions, the negotiated Maximum Segment Size is clamped down to 1452 bytes, allowing for 40 bytes in TCP and IP headers and 8 bytes in the PPPoE, totaling 1500 bytes that must fit into an ordinary Ethernet frame. A MTU mismatch can prevent a PPPoE connection from coming up or from properly carrying large datagrams. Checking the MTU setting is a good first step when troubleshooting PPPoE connections.

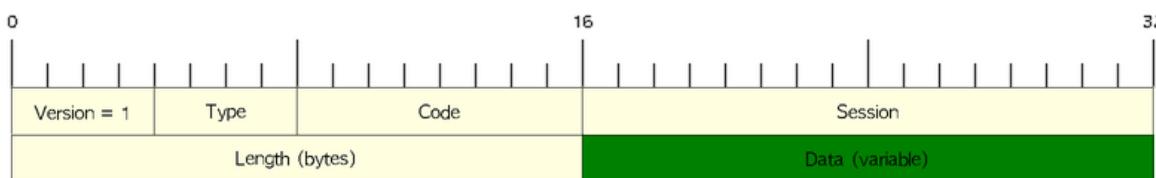


Figure 13. PPPoE Header Format

Length

Size of the data field in bytes

mnemo (ROTIS, SORTI)

PPPoE Active Discovery frames (EtherType = 0x8863)

- PADI: Initiation broadcasted by the clients
- PADO: Offer unicasted by Access Concentrators
- PADR: Request unicast by client
- PADS: Session-confirmation unicast by server with unique Session Id

- PADT: Termination

8.3.2. PPPoE Server

PPPoE Session Stage (Ethertype = 0x8864)

- on the ISP side

TODO: Replace the task below with step-by-step instructions

Task: Create a Broad Band Aggregation Group

```
(config)# bba-group pppoe <name>
(config-bba-group)# virtual-template 1
```

Task: Limit the Number Of Sessions on the Associated MAC

```
(config-bba-group)# sessions per-mac limit <number>
```

Task: Create the Virtual Template Interface

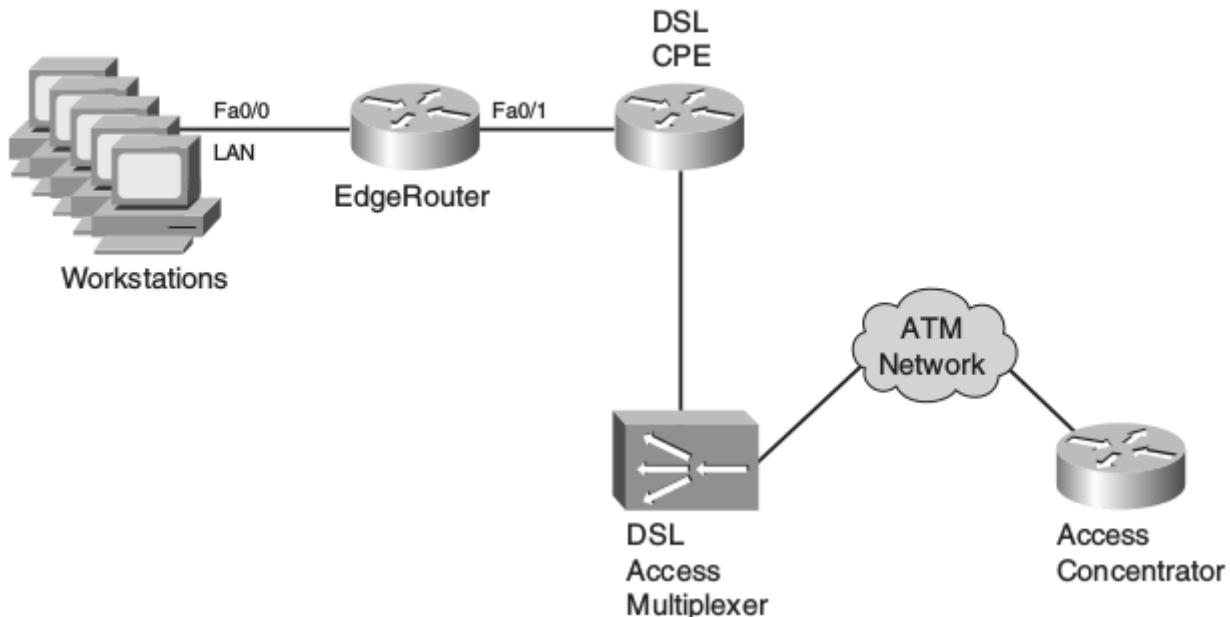
```
(config)# interface virtual-template 1
(config-if)# ip address 10.0.0.1 255.255.255.0
(config-if)# peer default ip address pool <pool-name>

(config)# ip local pool <pool-name> <ip-start> <ip-finish>
```

Task: Enable PPPoE Group on the Interface

```
(config-if)#
(config)# interface f0/0
(config-if)# no ip address
(config-if)# pppoe enable group MyGroup
(config-if)# no shutdown
```

8.3.3. PPPOE Client



Example Of Config on the Edge Router

```
# conf t
(config)# interface fa0/1
(config-if)# ip address 192.168.100.1 255.255.255.0
(config-if)# ip nat inside
(config)# interface fa0/1
(config-if)# pppoe-client dial-pool-number 1
(config-if)# exit
(config)# interface dialer1
(config-if)# mtu 1492
(config-if)# encapsulation ppp
(config-if)# ip address negotiated
(config-if)# ppp authentication chap
```

!The remaining CHAP commands have been omitted for brevity.

```
(config-if)# ip nat outside
(config-if)# dialer pool 1
(config-if)# dialer-group 1
(config-if)# exit
(config)# dialer-list 1 protocol ip permit
(config)# ip nat inside source list 1 interface dialer1 overload
(config)# access-list 1 permit 192.168.100.0 0.0.0.255
(config)# ip route 0.0.0.0 0.0.0.0 dialer1
```

Task: Verify PPPoE Connectivity

```
# show pppoe session
```

Task: Debug

```
# debug pppoe [data | errors | events | packets]
```

8.3.4. PPPoE Authentication

TODO: Add section from configuration guides

8.4. Ethernet WAN

EWAN

- Virtual Private LAN Services (VPLS),
- Multi-Protocol Label Switching (MPLS),
- Any-Transport Over MPLS (ATOM),
- Dot1Q-in-Dot1Q Tunnels (QnQ Tunnels),
- Metro-Ethernet.

8.4.1. VPLS

- Virtual Private LAN Service
- various WAN connections (over either IP or MPLS networks)
- uses QoS for audio and video
- provide multipoint Ethernet LAN services, or Transparent LAN Service (TLS).
 - A multipoint network service is one that allows a customer edge (CE) endpoint or node to communicate directly with all other CE nodes associated with the multipoint service.
 - By contrast, using a point-to-point network service such as ATM, the end customer typically designates one CE node to be the hub to which all spoke sites are connected. In this scenario, if a spoke site needs to communicate with another spoke site, it must communicate through the hub, and this requirement can introduce transmission delay.
- To provide multipoint Ethernet capability, the IETF VPLS drafts describe the concept of linking virtual Ethernet bridges using MPLS Pseudo-Wires (PW). As a VPLS forwards Ethernet frames at Layer 2, the operation of VPLS is exactly the same as that found within IEEE 802.1 bridges in that VPLS will self-learn the source MAC address to port associations, and frames are forwarded based upon the destination MAC address. If the destination address is unknown, or is a broadcast or multicast address, the frame is flooded to all ports associated with the virtual bridge.
- Although the forwarding operation of VPLS is relatively simple, the VPLS architecture needs to be able to perform other operational functions, such as
 - Autodiscover other provider edges (PE) associated with a particular VPLS instance
 - Signaling of PWs to interconnect VPLS virtual switch instances (VSI)
 - Loop avoidance

- MAC address withdrawal

8.4.2. Metro-Ethernet

- Ethernet on the metropolitan-area network (MAN) can be used as pure Ethernet, Ethernet over MPLS, or Ethernet over Dark Fiber, but regardless of the transport medium, we have to recognize that in network deployments requiring medium-distance backhaul or metropolitan (in the same city) connectivity, this Ethernet WAN technology is king. Why do we have so many different types of Metro-E solutions? The answer is that each has advantages and disadvantages. As an example, pure Ethernet-based deployments are cheaper but less reliable and scalable, and are usually limited to small-scale or experimental deployments. Dark Fiber-based deployments are useful when there is an existing infrastructure already in place, whereas solutions that are MPLS based are costly but highly reliable and scalable, and as such are used typically by large corporations.
- MPLS-based Metro-Ethernet network uses MPLS in the service provider's network. The subscriber will get an Ethernet interface on copper (for example, 100BASE-TX) or fiber (such as 100BASE-FX). The customer's Ethernet packet is transported over MPLS, and the service provider network uses Ethernet again as the underlying technology to transport MPLS. So MPLS-based Metro-E is effectively Ethernet over MPLS over Ethernet.
- Label Distribution Protocol (LDP) signaling can be used to provide site-to-site signaling for the inner label (VC label) and Resource Reservation Protocol-Traffic Engineering (RSVP-TE), or LDP can be used to provide the network signaling for the outer label.
- It should also be noted that a typical Metro-Ethernet system has a star network or mesh network topology, with individual routers or servers interconnected through cable or fiber-optic media. This is important when it becomes necessary to troubleshoot Metro-Ethernet solutions.

8.4.3. Ethernet Private Line (EPL)

The Ethernet Private Line service is used to deploy private line WAN connectivity across the Metro network. Typically the Ethernet service will forward packets to a long haul SONET network where Ethernet packets are encapsulated in SONET frames. The Ethernet packets are stripped off (de-encapsulated) at the SONET Provider Edge (PE) equipment and forwarded to the local Metro service provider network. The Ethernet private line is similar to any WAN link where VLAN information isn't sent between routers. The service provider does the rate limiting of traffic based on the CIR selected by the customer. The CIR is the guaranteed data rate service level agreement with the ISP. Traffic shaping or rate limiting of packets should be done at the Customer Edge (CE) on the CPE Ethernet interface to make sure packets are not dropped by the service provider.

8.4.4. Ethernet Virtual Private Line (EVPL)

The Ethernet virtual private line service is deployed for trunking of multiple VLANs across a Metro network (i.e. multiplexing multiple point to point EVCs). The 802.1q encapsulation protocol is the new standard that works with Cisco and other vendor equipment. The customer edge device uses 802.1q protocol to tag each Ethernet packet with VLAN membership before forwarding it across the virtual point to point Metrolink. QoS is applied at the Customer Edge (CE) Ethernet interface using per VLAN or per Class per VLAN traffic shaping.

Further Reading <http://goo.gl/Ffsq5o>

Part II : Layer 3 Technologies

Chapter 9. IPv4

- <https://tools.ietf.org/html/rfc791>[RFC 791]
- IP Ethernet protocol: 0x0800

9.1. IP Packet Format

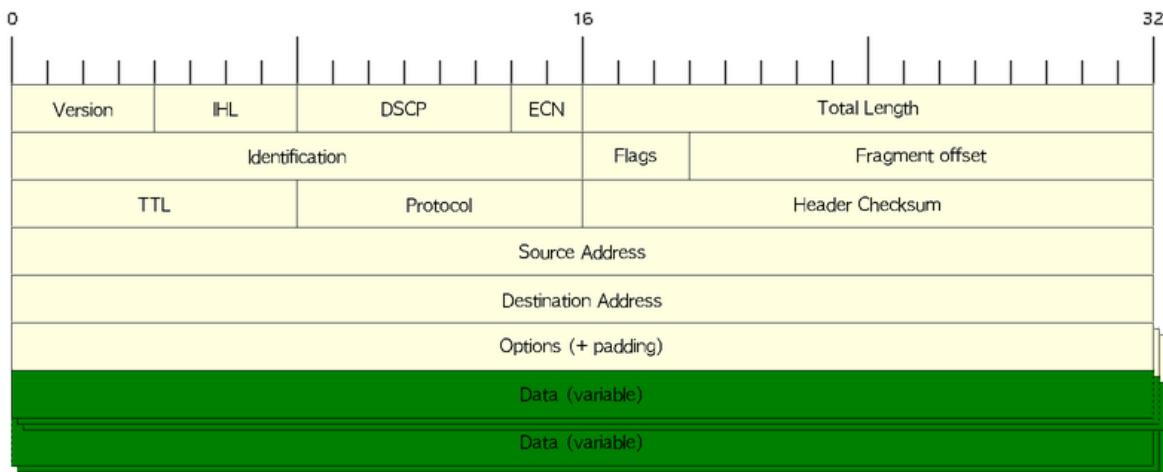


Figure 14. IP Header Format

IP Header Length (IHL)

Datagram header length in 32-bit words. Minimum value = 5

DSCP

Differentiated Service Code Protocol. Previously known as ToS Type of Service. Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

ECN

Explicit Congestion Notification RFC 3168

Total Length

Length, in bytes, of the entire IP packet, including the data and header.

Identification

Integer that identifies the current datagram. Used for fragmentation and re-assembly

Flags

3-bits in order from most significant to least significant

- bit 0: Reserved; must be zero.
- bit 1: Don't Fragment (DF)
- bit 2: More Fragments (MF)

Fragment Offset

Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

- the first fragment has an offset of zero
- this allows a maximum of offset of $(2^{Power13} - 1) * 8 = 65528$ bytes, which would exceed the maximum IP packet length of 65535 bytes with the IHL included ($65,528 + 20 = 65548$ bytes)

Time-to-Live

Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

Protocol

Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

Header Checksum

- equals to the one's complement of the one's complement sum of all 16 bit words in the IP header.
- initialized to all zeros at computation.
- included

Options

Allows IP to support various options, such as security.

Data

Contains upper-layer information.

9.2. IP Address

- 32-bits written in "dotted decimal"
- Classes: A,B,C,D,E
- Classless : prefix + host

Task: Assign an IP Address to an Interface

```
(config-if)# ip address <a.b.c.d> <e.f.g.h> [secondary]
```

Task: Display the IP Parameters for the Interface

```
# show ip interface
```

Task: Display the IP Networks the Device Is Connected To

```
# show ip route connected
```

9.3. CIDR

- Classless interdomain routing
- Defined in RFC 1517-1520
- Administrative assignment of large address blocks and the related summarized routes for the purpose of reducing the size of the Internet routing table
- Enabled by default

Task: Disable Classless Addressing

```
(config)# no ip classless
```

Task: Specify the Format In Which Netmask Appear for the Current Line

```
(config)# line vty <first> <last>
(config-line)# term ip netmask-format {bitcount | decimal | hexadecimal}
```

9.4. Private Addressing

- RFC 1918
- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

9.5. VLSM

- Variable length subnet mask

9.6. Subnet Zero

Task: Allow IP Subnet Zero

```
(config)# ip subnet-zero
```

9.7. Unnumbered Interfaces

- Borrow the IP address of another interface
- Only point-to-point (non-multiaccess) WAN interfaces
- You cannot reboot a IOS image over an ip unnumbered interface

Task: Configure Unnumbered Interfaces on Point-to-Point WAN Interfaces

```
(config-if)# ip unnumbered <interface-type interface-id>
```

9.8. 31-Bit Prefix

- Conserve IP address space
- Since RFC 3021
- Only on point-to-point WAN interfaces

Task: Use a 31-Bit Prefix on Point-to-Point WAN Interfaces

```
(config)# ip classless  
(config-if)# ip address a.b.c.d 255.255.255.254
```

9.9. Checksum

- IP checksum is a 16-bit field in IP header used for error detection for IP header. It equals to the one's complement of the one's complement sum of all 16 bit words in the IP header. The checksum field is initialized to all zeros at computation.
- One's complement sum is calculated by summing all numbers and adding the carries to the result. And one's complement is defined by inverting all 0s and 1s in the number's bit representation.

For example, if an IP header is 0x4500003044224000800600008c7c19acae241e2b.

9.9.1. Sender

First, divide the header hex into 16 bits each and sum them up,

```
4500 + 0030 + 4422 + 4000 + 8006 + 0000 + 8c7c + 19ac + ae24 + 1e2b = 2BBCF
```

Next fold the result into 16 bits by adding the carry to the result,

```
2 + BBCF = BBD1
```

The final step is to compute the one's complement of the one's complement's sum,

```
BBD1 = 1011101111010001
```

```
IP checksum = one's complement(1011101111010001) = 0100010000101110 = 442E
```

Note that IP header needs to be parsed at each hop, because IP addresses are needed to route the packet. To detect the errors at IP header, the checksum is validated at every hop.

9.9.2. Receiver

The validation is done using the same algorithm. But this time the initialized checksum value is 442E.

$$2BBCF + 442E = 2FFFD, \text{ then } 2 + FFFD = FFFF$$

Take the one's complement of FFFF = 0.

At validation, the checksum computation should evaluate to 0 if the IP header is correct.

9.10. Protocol

Number	Protocol
1	ICMP
2	IGMP
6	TCP
17	UDP
45	IRDP
46	RSVP
47	GRE
51	AH IPSec
50	ESP IPSec
58	ICMPv6
88	EIGRP
89	OSPF
103	PIM
112	VRRP

9.11. IP Options

- by default, cisco routers process IP options
- TLV
 - Option-Type: 8bit
 - Option-Length: 8 bit
 - Option-Data: Variable

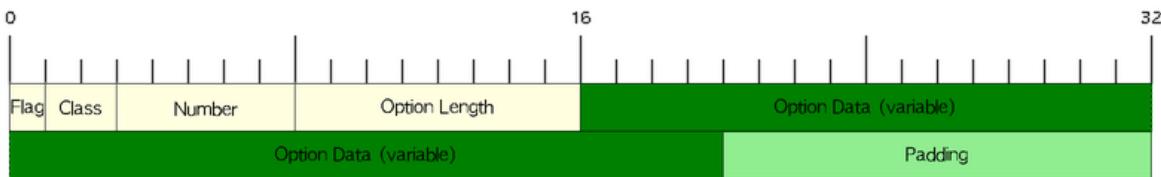


Figure 15. IP Options Format

Copied Flag

- 1 when the option is copied to each fragment

Option Class

- 0 for Control
- 2 for debugging and measurement for Internet Timestamp option

Option Number

- 0 **End of the option list**,
- 1 **No Operation**, again the option field is just one octet with no length or data fields.
- 2 **Security**, the length is 11 octets and the various security codes can be found in RFC 791.
- 3 **Loose Source Route** which is IP routing based on information supplied by the source station where the routers can forward the datagram to any number of intermediate routers in order to get to the destination.
- 4 **Internet Timestamp**
- 7 **Record Route** records the route that a datagram takes.
- 8 **Stream ID** has a length of 4 octets.
- 9 **Strict Source Route** which is IP routing based on information supplied by the source station where the routers can only forward the datagram to a directly connected router in order to get to the next hop indicated in the source route path.

Task: Ignore all IP options

```
(config)# ip options drop
```

Task: Discard any IP datagram containing a source-route option

```
(config)# no ip source-route
```

9.12. IP fragmentation and Re-assembly

TODO

- minimum MTU: 576 bytes (ipv4) 1280 bytes (ipv6)
- When a router receives a packet, it examines the destination address and determines the outgoing interface to use and that interface's MTU. If the packet size is bigger than the MTU, and the Do not Fragment (DF) bit in the packet's header is set to 0, then the router may fragment the packet.

The router divides the packet into fragments. The max size of each fragment is the MTU minus the IP header size (20 bytes minimum; 60 bytes maximum). The router puts each fragment into its own packet, each fragment packet having following changes:

The total length field is the fragment size. The more fragments (MF) flag is set for all fragments except the last one, which is set to 0. The fragment offset field is set, based on the offset of the fragment in the original data payload. This is measured in units of eight-byte blocks. The header checksum field is recomputed. - Fields used: Identification, DF, MF, Offset (and total length of each fragment) - when one fragment is lost, the entire IP datagram is resent **IP doesn't have any timeout or retransmission** TCP or higher layers have

Task: Set the IP MTU packet size for an interface.

```
(config-if)# ip mtu <bytes>
```

Chapter 10. ICMP

- Internet Control Management Protocol
- RFC 792
- Generates error messages
- Protocol Number: 1

10.1. Header

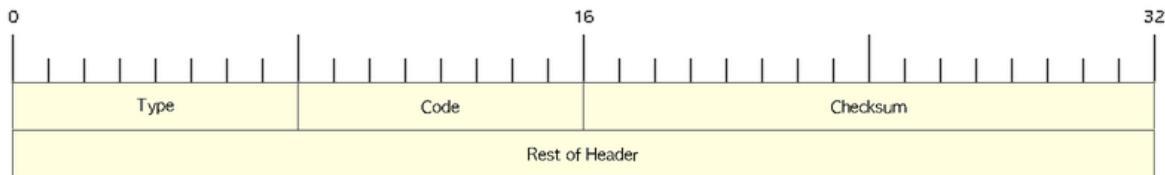


Figure 16. ICMP Header Format

10.2. Control Messages

Type	Code	Description
0	0	Echo reply (used to ping)
1-2	-	unassigned Reserved

10.3. ICMP Unreachable Messages

Type: 3

Code	Description
0	Destination network unreachable
1	Destination host unreachable
2	Destination protocol unreachable
3	Destination port unreachable
4	Fragmentation required, and DF flag set
5	Source route failed
6	Destination network unknown

Code	Description
7	Destination host unknown
8	Source host isolated
9	Network administratively prohibited
10	Host administratively prohibited
11	Network unreachable for ToS
12	Host unreachable for ToS
13	Communication administratively prohibited
14	Host Precedence Violation
15	Precedence cutoff in effect

Task: Disable the sending of ICMP protocol unreachable and host unreachable messages

```
(config-if)# no ip unreachable
```



Disabling the unreachable messages also disables IP Path MTU Discovery because path discovery works by having the software send unreachable messages.

Task: Clear all current ICMP unreachable statistics for all configured interfaces

```
# clear ip icmp rate-limit [<interface type number>]
```

Task: Specify the rate limitation of ICMP unreachable destination messages and the error message log threshold for generating a message

```
(config-if)# ip icmp rate-limit unreachable [df] [<ms>] [log [<packets>] [<interval-ms>]]
```



The default is no unreachable messages are sent more often than once every half second.

Task: Display all current ICMP unreachable statistics for all configured interfaces.

```
# show ip icmp rate-limit [<interface type number>]

Interval (millisecond)      500          500
Interface                  # DF bit unreachables  # All other unreachables
-----
Ethernet0/0                 0            0
Ethernet0/2                 0            0
Serial3/0/3                 0            19
The greatest number of unreachables is on serial interface 3/0/3.
```

10.4. ICMP Source Quench Messages

Type: 4

4 – Source Quench 0 deprecated Source quench (congestion control)

10.5. ICMP Redirect Messages

Type: 5

code	description
0	Redirect Datagram for the Network
1	Redirect Datagram for the Host
2	Redirect Datagram for the ToS & network
3	Redirect Datagram for the ToS & host
6	deprecated Alternate Host Address
7	unassigned Reserved

Task: Disable the sending of ICMP redirect messages to learn routes

```
(config-if)# no ip redirects
```

Task: Display the address of the default router and the address of hosts for which an ICMP redirect message has been received

```
# show ip redirects

Default gateway is 172.16.80.29

Host           Gateway        Last Use    Total Uses Interface
172.16.1.111   172.16.80.240      0:00          9  Ethernet0
172.16.1.4     172.16.80.240      0:00          4  Ethernet0
```

10.6. ICMP Echo Request

8 – Echo Request

10.7. ICMP Router Messages

9 – Router Advertisement 10 – Router Solicitation

10.8. ICMP Time Exceeded

11 – Time Exceeded

0 TTL expired in transit 1 Fragment reassembly time exceeded

10.9. Parameter Problem

12 – Parameter Problem: Bad IP header

0 Pointer indicates the error 1 Missing a required option 2 Bad length

10.10. Timestamp Messages

13 – Timestamp 14 – Timestamp Reply

10.11. Information Request

15 – Information Request 16 – Information Reply

10.12. Address Mask Messages

17 – Address Mask Request 18 – Address Mask Reply

- To request the subnet mask for a particular subnetwork
- Can be used by an attacker to gain network mapping information

Task: Disable the sending of ICMP mask reply messages

```
(config-if)# no ip mask-reply
```

10.13. Ping

- Packet InterNet Groper
- uses two ICMP query messages, ICMP echo requests, and ICMP echo replies to determine whether a remote host is active
 - The ping command first sends an echo request packet to an address, and then it waits for a reply.
 - The ping is successful only if the ECHO REQUEST gets to the destination, and the destination is able to get an ECHO REPLY back to the source of the ping within a predefined time interval.
- measures the amount of time it takes to receive the echo reply

10.14. Traceroute

- records the source of each ICMP "TIME EXCEEDED" message in order to provide a trace of the path the packet took to reach the destination.
- sends out a sequence of UDP datagrams, each with incrementing TTL values, to an invalid port address (Default 33434) at the remote host.
 - First, three datagrams are sent, each with a TTL field value set to 1. The TTL value of 1 causes the datagram to "timeout" as soon as it hits the first router in the path. This router then responds with an ICMP "time exceeded" message which indicates that the datagram has expired.
 - Next, three more UDP messages are sent, each with the TTL value set to 2. This causes the second router in the path to the destination to return ICMP "time exceeded" messages.
 - This process continues until the packets reach the destination and until the system that originates the traceroute receives ICMP "time exceeded" messages from every router in the path to the destination. Since these datagrams try to access an invalid port (Default 33434) at the destination host, the host responds with ICMP "port unreachable" messages that indicate an unreachable port. This event signals the traceroute program to finish.

Chapter 11. TCP

Transmission Control Protocol

- provides a reliable, connection-oriented, byte stream, transport layer service.
 - packetizes the user data into segments,
 - sets a timeout any time it sends data,
 - acknowledges data received by the other end,
 - reorders out-of-order data,
 - discards duplicate data,
 - provides end-to-end flow control, and
 - calculates and verifies a mandatory end-to-end checksum.
- Protocol Number: 6

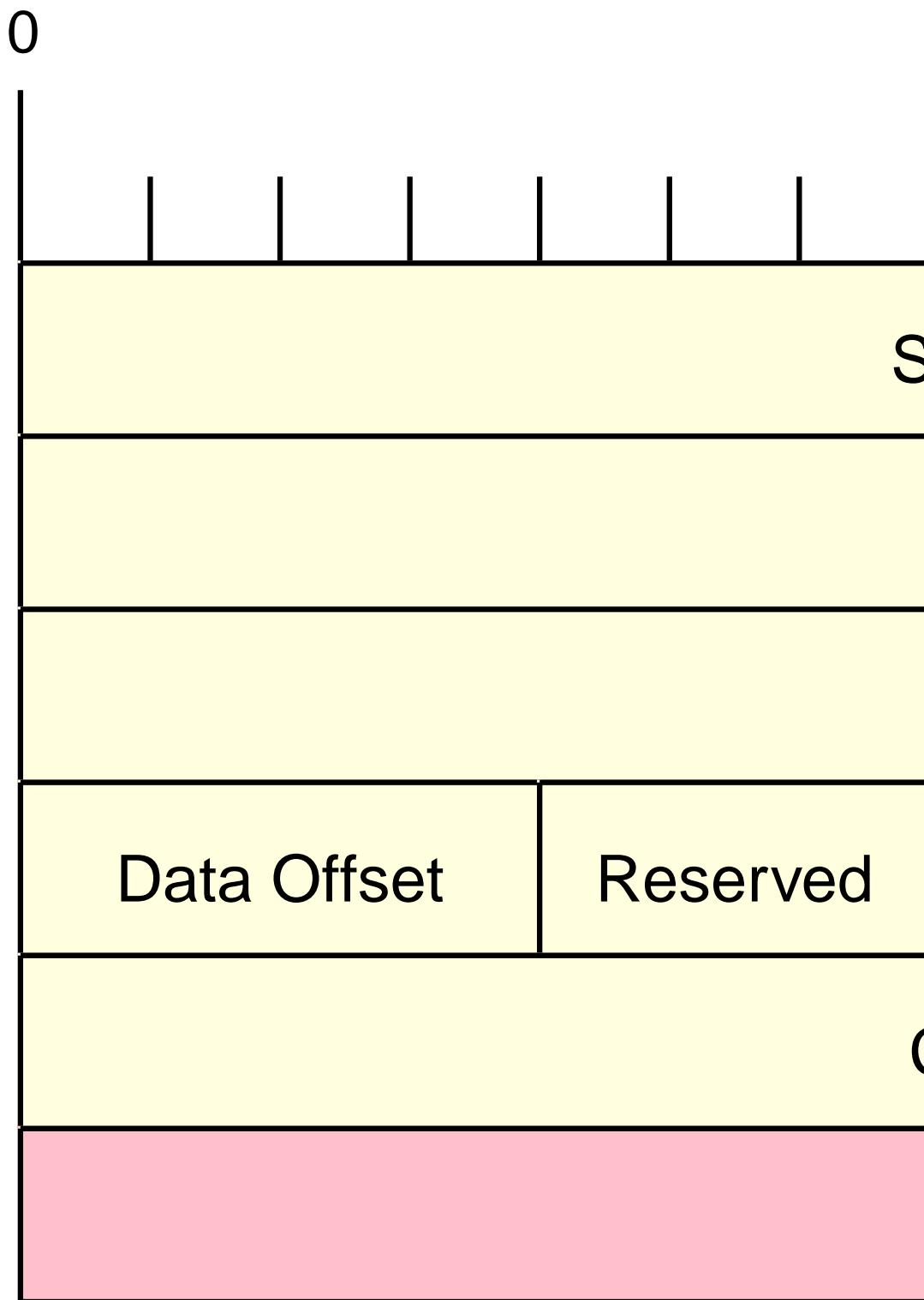


Figure 17. TCP Header Format

11.1. Connection establishment

Three-Hand Shake

SYN seq(A), SYN-ACK seq(B) ack(A+1), ACK seq(B+1)

Task: Set the amount of time before attempting to establish a TCP connection.

```
(config)# ip tcp synwait-time <seconds>
```



The default is 30 seconds.

11.2. TCP MSS

- Maximum Segment Size: max amount of data that a host is willing to receive in a single TCP datagram
- sent as TCP option in TCP SYN segment

11.3. TCP Window

amount of unacknowledged data a sender can send on a particular connection before it gets an acknowledgment back from the receiver, that it has received some of the data.

11.4. Sliding window operations

todo: Excellent lessons <http://www.omnisecu.com/tcpip/tcp-sliding-window.php>

11.5. TCP Options

TODO TODO how tcp options work

what is the ACL IP options selective drop feature

11.6. Ident

- TCP client Identify Protocol
- RFC 1413,
- allows a system to query the identity of a user initiating a TCP connection or a host responding to a TCP connection.
 - When implemented, the Ident service allows a user to obtain identity information by simple connecting to a TCP port on a system, and issuing a simple text string requesting information.
- disabled by default for security reasons

```
(config)# no ip identd
```

11.7. TCP Small Servers

- echo, chargen, daytime , discard services
- disabled by default

The TCP small servers are:

- Echo: (port 7) Echoes back whatever you type through the telnet x.x.x.x echo command.
- Chargen: (port 19) Generates a stream of ASCII data. Use the telnet x.x.x.x chargen command.
- Discard: (port 9) Throws away whatever you type. Use the telnet x.x.x.x discard command.
- Daytime: (port 13) Returns system date and time, if it is correct. It is correct if you run Network Time Protocol (NTP), or have set the date and time manually from the exec level. Use the telnet x.x.x.x daytime command.

Task: Enable TCP small servers

```
(config)# service tcp-small-servers
```

Chapter 12. UDP

- Protocol number: 17
- RFC 768

MSS

- Default 536 bytes ??

12.1. Message Format

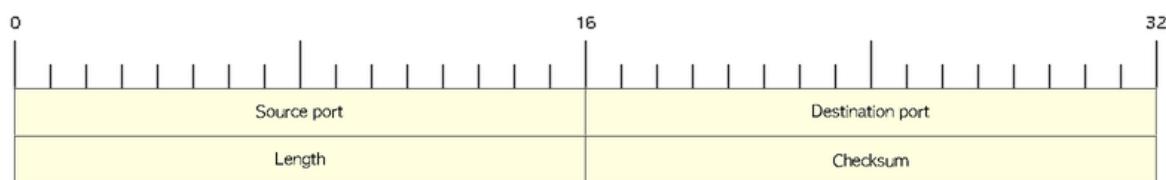


Figure 18. UDP Header Format

12.2. UDP checksum

12.3. UDP dominance

- TCP starvation

12.4. UDP Small Servers

- echo, chargen, discard services
- disabled by default

Task: Enable UDP small servers

```
(config)# service udp-small-servers
```

Chapter 13. ARP

- Configuration Guides > IP Addressing > ARP
- RFC 826
- Finds MAC address of a host given IP address
- maintains ARP cache
- ARP Ethernet protocol: 0x0806
- ARP request for the same address rate-limited to one request every 2 seconds

13.1. Protocol

TODO: better work to show that ARP is encapsulated in Ethernet frame, use color

8	6	6	2	46-1500	4
Preamble	DA	SA	Type=0x806	ARP Request or Reply	FCS

Figure 19. ARP

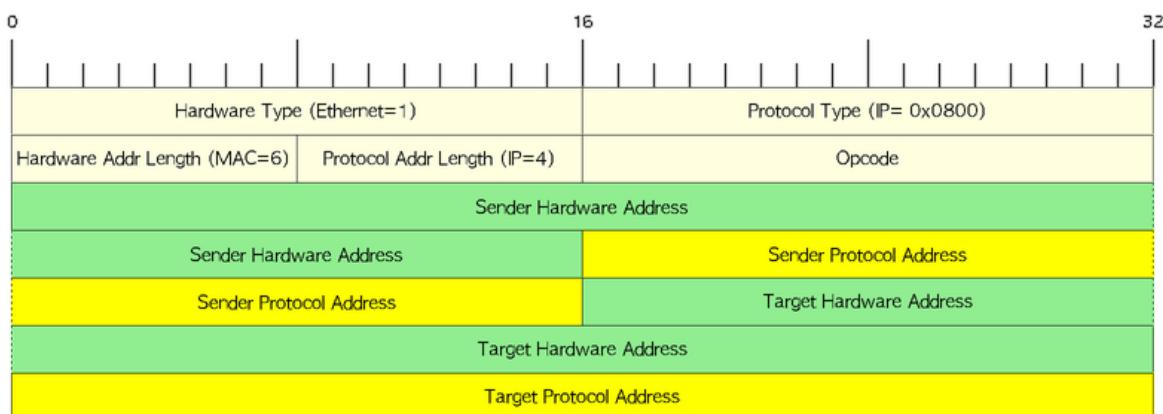


Figure 20. Header Format

OpCode

- 1 for request
- 2 for response

13.2. Static ARP Entries

Task: Enable the Interface Encapsulation

```
(config-if)# arp {arpa | frame-relay | snap}
```

Task: Define Static ARP Entries

```
(config)# arp <ip-address> <hardware-address> <encapsulation-type> [<interface-type>]
```

Task: Define Static ARP Entries for a Specific VRF

```
# arp vrf <name> <hardware-address> <encapsulation-type> [<interface-type>]
```

13.3. Dynamic ARP Entries

- Stored in ARP cache
- Timeout: 4 hours by default

Task: Clear the Entire ARP Cache on an Interface

```
# clear arp interface <type-number>
```

Task: Clear All Dynamic Entries from the ARP Cache, the Fast-Switching Cache, and the IP Route Cache

```
# clear arp-cache
```

Task: Set an Expiration Time for Dynamic Entries In the Arp Cache

```
(config-if)# arp timeout <seconds>
```

Task: Display the Arp Cache

```
# show ip arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.108.42.112	120	0000.a710.4baf	ARPA	Ethernet3
AppleTalk	4028.5	29	0000.0c01.0e56	SNAP	Ethernet2
Internet	110.108.42.114	105	0000.a710.859b	ARPA	Ethernet3
AppleTalk	4028.9	-	0000.0c02.a03c	SNAP	Ethernet2
Internet	10.108.42.121	42	0000.a710.68cd	ARPA	Ethernet3
Internet	10.108.33.9	-	0000.0c01.7bbd	SNAP	Fddi0

Task: Display ARP and RARP Processes

```
# sh processes cpu | include (ARP|PID)
```

13.4. Proxy ARP

- Same message as ARP

- Allows response for IP address in remote subnet.
- RFC 1027
- Mostly replaced by DHCP nowadays

Task: Disable Proxy ARP Globally

```
# ip arp proxy disable
```

Task: Disable Proxy ARP on an Interface

```
(config-if)# no ip proxy-arp
```

High CPU utilization in the ARP input process occurs if the router has to originate an excessive number of ARP requests. The router uses ARP for all hosts, not just those on the local subnet, and ARP requests are sent out as broadcasts, which causes more CPU utilization on every host in the network. ARP requests for the same IP address are rate-limited to one request every two seconds, so an excessive number of ARP requests would have to originate for different IP addresses. This can happen if an IP route has been configured pointing to a broadcast interface (as opposed to next-hop). A most common example is a default route such as:

```
ip route 0.0.0.0 0.0.0.0 Fastethernet0/0
```

In this case, the router generates an ARP request for each IP address that is not reachable through more specific routes, which practically means that the router generates an ARP request for almost every address on the Internet.

When the router needs to route a packet which matches an entry in the routing table with a next-hop value, it performs Layer 3 to Layer 2 resolution for the next-hop address. If it matches an entry in the routing table with just the outgoing/exit local interface, without a next-hop value, it performs Layer 3 to Layer 2 resolution for the final destination of the IP packet.

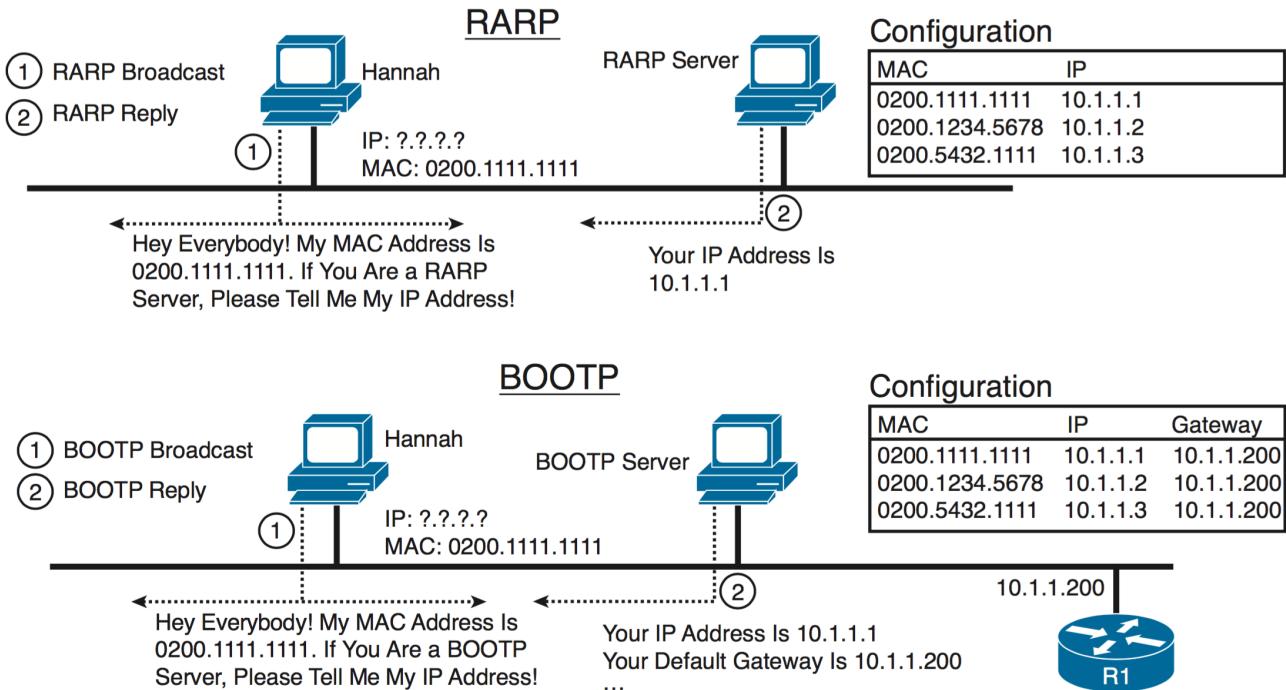
13.5. Gratuitous ARP

- broadcast ARP messages where the SPA=TPA and THA=FF:FF:FF:FF:FF:FF
 - to detect IP address conflict
 - to update other machine ARP table
 - to update mac table of the connected switch



If we see multiple gratuitous ARPs from the same host frequently, it can be an indication of bad Ethernet NICs

13.6. RARP and BootP As DHCP Precursors



- A RARP request is a host's attempt to find its own IP address. So RARP uses the same old ARP message, but the ARP request lists a MAC address target of its own MAC address and a target IP address of 0.0.0.0. A preconfigured RARP server, which must be on the same subnet as the client, receives the request and performs a table lookup in its configuration. If that target MAC address listed in the ARP request is configured on the RARP server, the RARP server sends an ARP reply, after entering the configured IP address in the Source IP address field.
- BOOTP was defined in part to improve IP address assignment features of RARP. BOOTP uses a completely different set of messages, defined by RFC 951, with the commands encapsulated inside an IP and UDP header. With the correct router configuration, a router can forward the BOOTP packets to other subnets—allowing the deployment of a centrally located BOOTP server. Also, BOOTP supports the assignment of many other tidbits of information, including the subnet mask, default gateway, DNS addresses, and its namesake, the IP address of a boot (or image) server. However, BOOTP does not solve the configuration burden of RARP, still requiring that the server be preconfigured with the MAC addresses and IP addresses of each client.

13.7. ARP vulnerabilities

1. Since ARP does not authenticate requests or replies, ARP Requests and Replies can be forged
2. ARP is stateless: ARP Replies can be sent without a corresponding ARP Request
3. According to the ARP protocol specification, a node receiving an ARP packet (Request or Reply) must update its local ARP cache with the information in the source fields, if the receiving node already has an entry for the IP address of the source in its ARP cache.

Typical exploitation of these vulnerabilities:

- ARP poisoning: a forged ARP Request or Reply can be used to update the ARP cache of a remote system with a forged entry
- This can be used to redirect IP traffic to other hosts

Chapter 14. DHCP

Configuration guides > IP Addressing > [DHCP](#)

- [RFC 2131](#)
- Dynamic Host Configuration Protocol
- Based on BOOTP (itself based on RARP)
- Client/agent relay/server model
- UDP port 67 (server), port 68 (client)

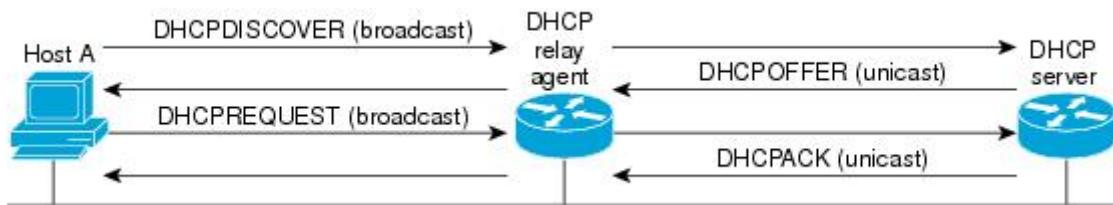


Figure 21. DHCP Request for an IP Address from a DHCP Server

14.1. Protocol Operations



Figure 22. DHCP

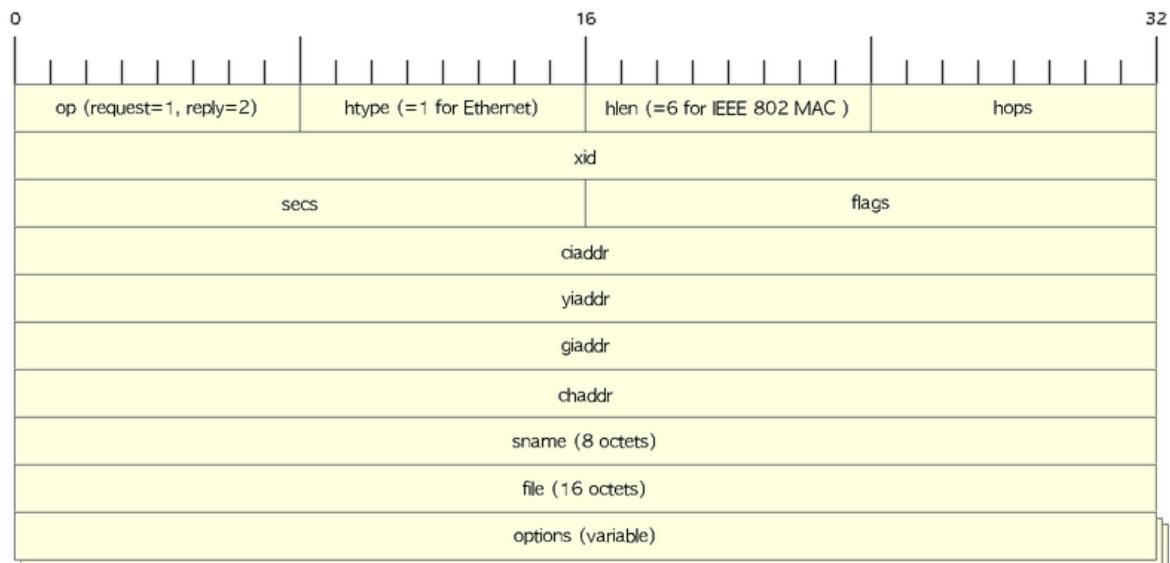


Figure 23. DHCP Message

Field	Octets	Description
op	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Hardware address type for the local network, same value than ARP HRD field
hlen	1	Hardware address length, same value than ARP HLN
hops	1	Set to zero by clients, optionally incremented by 1 by each relay agent
xid	4	Transaction ID, a random number chosen by the client, used by the client and server to match replies with requests
secs	2	Filled in by client, seconds elapsed since client began address acquisition or renewal process. This may be used by a busy DHCP server to prioritize replies when multiple client requests are outstanding.
flags	2	Flags. Actually 1 B bit is used. if Broadcast flag set by a client, the server or relay knows that they should reply with a broadcast.
ciaddr	4	Client IP address; only filled in if client is in BOUND, RENEW or REBINDING state and can respond to ARP requests. The client does not use this field to request a particular IP address in a lease; it uses the Requested IP Address DHCP option.
yiaddr	4	'your' (client) IP address assigned by the DHCP server.
siaddr	4	IP address of next server to use in bootstrap returned in DHCPOFFER, DHCPACK by server which may not be the server sending this reply. The sending server always includes its own IP address in the Server Identifier DHCP option.
giaddr	4	Relay agent IP address, used in booting via a relay agent.
chaddr	16	Client L2 hardware address.
sname	64	Optional server host name, null terminated string.
file	128	Boot file name, null terminated string; "generic" name or null in DHCPDISCOVER, fully qualified directory-path name in DHCPOFFER. Optionally used by a client to request a particular type of boot file in a DHCPDISCOVER message. Used by a server in a DHCPOFFER to fully specify a boot file directory path and filename. This field may also be used to carry DHCP options, using the "option overload" feature, indicated by the value of the DHCP Option Overload option.

Field	Octets	Description
options	var	<p>Optional parameters field. The first four bytes contain the same BOOTP magic cookie (decimal) values 99, 130, 83 and 99. The remainder of the field consists of a list of tagged TLV options.</p> <p><i>Commonly used options</i></p> <ul style="list-style-type: none"> • 0 Pad • 1 Subnet Mask • 3 Router Address • 6 Domain Name Server • 15 Domain Name • 50 Requested IP Address • 51 Address Lease Time • 52 Option Overload • 53 DHCP Message Type • 54 Server Identifier • 55 Parameter Request List • 56 DHCP Error Message • 58 Lease Renewal Time • 59 Lease Rebinding Time • 61 Client Identifier • 82 DHCP Relay • 119 Domain Search List • 255 End



The DHCP Message Type option (53) is a 1-byte-long option that is always used with DHCP messages and has the following possible values: DHCPDISCOVER (1), DHCPOFFER (2), DHCPREQUEST (3), DHCPDECLINE (4), DHCPACK (5), DHCPNAK (6), DHCPRELEASE (7), DHCPINFORM (8), DHCPFORCERENEW (9), DHCPLEASEQUERY (10), DHCPLEASEUNASSIGNED (11), DHCPLEASEUNKNOWN (12), and DHCPLEASEACTIVE (13).

14.2. DHCP Client

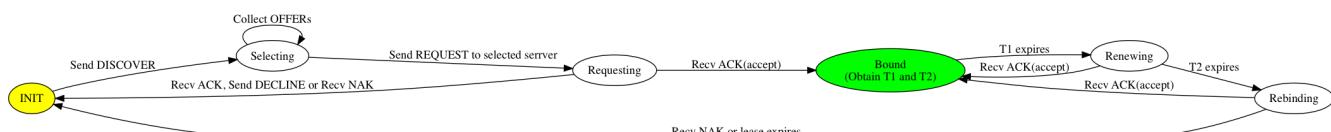


Figure 24. DHCP Client State Machine

Task: Acquire an IP Address on an Interface from DHCP

```
(config-if)# ip address dhcp
```

Task: Display the DHCP Packets Sent and Received During Troubleshooting on the Client Side

```
# debug dhcp detail
```

Task: Force a Release Of a DHCP Lease

```
# release dhcp
```

The **release dhcp** command



- Starts the process to immediately release a DHCP lease for the specified interface.
- Does not deconfigure the **ip address dhcp** command specified in the configuration file for the interface.

Task: Force a Renewal Of a DHCP Lease

```
# renew dhcp
```



- The **renew dhcp** command advances the DHCP lease timer to the next stage, at which point one of the following occurs:
 - If the lease is currently in a BOUND state, the lease is advanced to the RENEW state and a DHCP RENEW request is sent.
 - If the lease is currently in a RENEW state, the timer is advanced to the REBIND state and a DHCP REBIND request is sent.
- If there is no response to the RENEW request, the interface remains in the RENEW state. In this case, the lease timer will advance to the REBIND state and subsequently send a REBIND request.
- If a NAK response is sent in response to the RENEW request, the interface is deconfigured.

14.2.1. Configurable DHCP Client Feature

- Allows a client to use a user-specified client identifier, class identifier or suggested lease time when requesting an address from a DHCP server.
- Options available:
 - Option 33: configure a list of static routes in the client.
 - Option 51: request a lease time for the IP address.
 - Option 55: request certain options from the DHCP server

- Option 60: configure the vendor class identifier string to use in the DHCP interaction.
- Option 61: specify their unique identifier

14.2.2. FORCERENEW Message Handling

TODO: Explain the feature

Task: Configure FORCERENEW Message Handling

```
! Specify the key chain to be used in authenticating a request
(config)# key chain <name>
(config-keychain)# key <id>
(config-keychain-key)# key-string <text>
!
! Specify the type of authentication
(config)# interface <type number>
(config-if)# ip dhcp client authentication key-chain <name>
(config-if)# ip dhcp client authentication mode <type>
!
# ip dhcp-client forcerenew
```

14.3. DHCP Server

- Accepts address assignment requests and renewals from clients
- Assign address, name server, gateways, ...
- Accepts broadcasts from local clients or relay agents
- Database as a tree used for attribute inheritance
 - Root: address pool for natural networks
 - Branches: subnetwork address pools
 - Leaves: manual bindings

Task: Clear DHCP Server Variables

```
clear ip dhcp binding { <address> | * }
clear ip dhcp conflict { <address> | * }
clear ip dhcp server statistics
```

Task: Troubleshoot DHCP IP Address Assignments, Lease Expirations, and Database Changes

```
# debug ip dhcp server events
```

14.3.1. Database Agent

- Host (ftp, tftp, rcp) or storage that stores the DHCP bindings database.

Task: Save Automatic Bindings on a Remote Host

```
ip dhcp database <url> [timeout <seconds>] [ write-delay <seconds>]
```

- **url:** can be ftp,tftp, rcp, flash, disk
- **timeout:** how long the DHCP server wait before aborting database transfer.
default: 5 minutes
- **write-delay:** how soon the DHCP server should send database updates. default: 5 minutes, minimum: 60 seconds



Task: Run DHCP Server Without Database Agent

```
(config)# no ip dhcp conflict logging
```



- Not recommended
- TODO: add the reason

14.3.2. Address Pool

- Specify which DHCP options to use for the client
 - If the client is not directly connected to the DHCP server (the giaddr field of the DHCPDISCOVER broadcast message is nonzero), the server matches the DHCPDISCOVER with the DHCP pool that has the subnet that contains the IP address in the giaddr field.
 - If the client is directly connected to the DHCP server (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pools that contain the subnets configured on the receiving interface. If the interface has secondary IP addresses, subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Task: Create a Pool

```
(config)# ip dhcp pool <name>
```

Task: Specify the Subnet Network Number and Mask Of the Address Pool

```
(dhcp-config)# network <network-number> [mask | prefix-length]
```

Task: Specify the Secondary Subnets

```
(dhcp-config)# network <network-number> [mask | prefix-length] secondary
```

Task: Exclude IP Address

```
(config)# ip dhcp excluded-address <low-address> [<high-address>]
```

Task: Specify the Domain Name

```
(dhcp-config)# domain-name <example.com>
```

Task: Specify the Name Server Per Order Of Preference

```
(dhcp-config)# dns-server <address> [<address2> ... <address8>]
```

Task: Specify the Default Boot Image for a Client

```
(dhcp-config)# bootfile <filename>
```

Task: Specify the Netbios Server

```
(dhcp-config)# netbios-name-server <address> [<address2> ... <address8>]  
(dhcp-config)# netbios-node-type <type>
```

Task: Specify the Gateway

```
(dhcp-config)# default-router <address> [<address2> ... <address8>]
```

Task: Specify a Custom DHCP Code

```
(dhcp-config)# option <code> [instance <number>] {ascii <string> | hex <string> | <ip-address>}
```

Task: Configure the Duration Of the Lease

```
(dhcp-config)# lease <days> [<hours> [<minutes>] ]
```

Task: Specify the Lease for Ever

```
(dhcp-config)# lease infinite
```

The DHCP OFFER message includes the lease time (T), which provides the upper bound on the amount of time the address can be used if it is not renewed. The message also contains the renewal time (T1), which is the amount of time before the client should attempt to renew its lease with the server from which it acquired its lease, and the rebinding time (T2), which bounds the time in which it should attempt to renew its address with any DHCP server. By default, T1 = (T/2) and T2 = (7T/8).



Task: Configure the Utilization Mark Of the Current Address Pool Size

```
(dhcp-config)# utilization mark high <percentage-number> [log]  
(dhcp-config)# utilization mark low <percentage-number> [log]
```

Task: Configure a DHCP Address Pool with Secondary Subnets

```
(dhcp-config)# override default-router ??  
(dhcp-config)# override utilization high <percentage>  
(dhcp-config)# override utilization low <percentage>
```

TODO: add explanation

Task: Verify the DHCP Address Pool Configuration

```
# show ip dhcp pool [name]  
# show ip dhcp binding [address]  
# show ip dhcp conflict [name]  
# show ip dhcp database [url]  
# show ip dhcp server statistics [type-number]
```

14.3.3. Address Bindings

- Mapping between the IP address and MAC address of a client

Task: Display the Current Mapping

```
# show ip dhcp binding
```

Automatic Bindings

- Dynamically maps hardware address to an IP address from a pool.
- Stored in volatile RAM and periodically copied to database agent

Manual Binding

- MAC address of hosts are found in the DHCP database
- Stored in NVRAM
- Can be configured
 - Individually and stored in NVRAM
 - In batch from text files

Task: Specify the IP Address and Subnet Mask Of the Client

```
(dhcp-config)# host <address> [<mask>| </prefix-length>]
```

Task: Specify the Unique Identifier for a DHCP Client

```
(dhcp-config)# client-identifier <unique-identifier>
```

- Send with DHCP option 61
- Unique identifier
 - 7-byte: 1byte for the media , 6 byte for the MAC address
 - 27-byte: vendor, MAC address, source interface of the client

Task: Determine the Client Identifier

```
# debug ip dhcp server packet
```

```
DHCPD:DHCPOPTION received from client 0b07.1134.a029 through relay 10.1.0.253.  
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
```

Task:

```
(dhcp-config)# hardware-address <hw-address> [<protocol-type> | <hw-number>]
```

- For client who can not send a client identifier in the packet

Task:

```
(dhcp-config)# client-name <name>
```

- Do not include the domain name

14.3.4. Static Mapping

- From customer-created text file that DHCP server reads at boot
 - Short configuration: no need for several numerous host pools with manual bindings
 - Reduce space required in NVRAM to maintain address pools
- The file format has the following elements:
 - Database version number
 - End-of-file designator
 - Hardware type
 - Hardware address
 - IP address
 - Lease expiration
 - Time the file was created

Example

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address      Type    Hardware address      Lease expiration
10.0.0.4 /24     1        0090.bff6.081e      Infinite
10.0.0.5 /28     id       00b7.0813.88f1.66  Infinite
10.0.0.2 /21     1        0090.bff6.081d      Infinite
*end*
```

Task: Configure the DHCP Server to Read a Static Mapping Text File

```
(dhcp-config)# origin file <url>
```

14.3.5. Pings

- DHCP server pings an IP address twice before assigning it to a client.
- If the ping is unanswered after waiting for 2 seconds, the server assumes that the address is not in use.

Task: Specify the Number Of Packets Sent to a Pool Address Before Assigning It to a Client

```
(config)# ip dhcp ping packets <number>
```

Task: Specify How Long a DHCP Server Waits for a Ping Reply from an Address Pool

```
(config)# ip dhcp ping timeout <milliseconds>
```

14.3.6. BOOTP Interoperability

Task: Configure the DHCP Server to Not Reply to Any BOOTP Requests.

```
(config)# ip dhcp boot ignore
```

Task: Forward Ignored BOOTP Request Packets to Another DHCP Server

```
(config)# ip helper-address <a.b.c.d>
```

14.3.7. Central DHCP Server

- Updates specific DHCP options for remote DHCP server

Task: Import DHCP Option Parameters from Central DHCP Server

```
(dhcp-config)# import all  
(config)# interface <type> <number>  
(config-if)# ip address dhcp
```

Task: Display the Options That Are Imported from the Central DHCP Server

```
# sh ip dhcp import
```

14.3.8. Option 82

- DHCP option contains information known by the relay agent
- For dynamic IP addresses allocation
- TOBECOMPLETED
- By default, OS DHCP server uses info provided by option 82

Task: Enable DHCP Address Allocation with Option 82

```
(config)# ip dhcp use class
```

Task: Define a DHCP Class and Relay Agent Information Patterns

```
(config)# ip dhcp class <name>  
(dhcp-class)# relay agent information  
(dhcp-class-info)# relay-information hex <pattern> [*] [bitmask <mask>]
```

Task: Display DHCP Class Matching Results

```
# debug ip dhcp server class
```

Static Route with the Next-Hop Dynamically Obtained Through DHCP

TODO: explanation/context

Task: Assign a Static Route for the Default Next-Hop Device When the DHCP Server Is Accessed for an IP Address

```
# ip route <prefix> <mask> {<ip-address> | <interface-number> [<ip-number>]} dhcp  
[<distance>]
```



- Ensure that the DHCP client and server are defined to supply a DHCP device option 3 of the DHCP packet.
- If the DHCP client is not able to obtain an IP address or the default device IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table.

14.3.9. Statistics

Task: Display Server Statistics

```
# show ip dhcp server statistics
```

Task: Reset All DHCP Server Counters to 0

```
# clear ip dhcp server statistics
```

14.4. DHCP Relay Agent

- Forwards requests and replies between clients and servers not on the same physical subnet
- Sets the **giaddr** field and adds option 82
- DHCP server and relay agent are enabled by default

Task: Specify the Packet Forwarding Address

```
(config-if)# ip helper-address <a.b.c.d>
```

Task: Reduce the Frequency with Which DHCP Clients Change Their Addresses and Forwards Client Requests to the Server That Handles the Previous Request.

```
(config-if)# ip dhcp relay prefer known-good-server
```

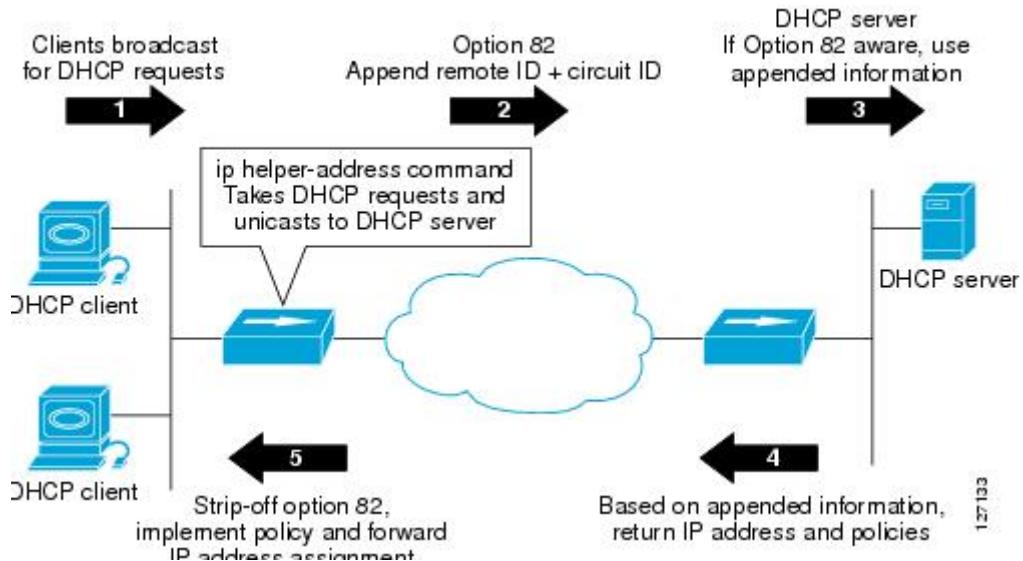


- The relay agent deletes the ARP entries for addresses offered to the client on unnumbered interfaces.

Task: Disable the DHCP Relay Agent Service

```
# no service dhcp
```

14.4.1. Option 82



Task: Insert the DHCP Relay Agent Information Option In BOOTREQUEST Messages Forwarded to a DHCP Server

```
# ip dhcp relay information option
```



- This function is disabled by default

Task: Check Whethers the Relay Agent Information Option Forwarded BOOTREPLY Message Is Valid

```
# ip dhcp relay information check
```

Task: Configure the Reforwarding Policy

```
# ip dhcp relay information policy {drop | keep | replace }
```

Task: Configure All Interfaces As Trusted Sources Of the DHCP Relay Information Option.

```
# ip dhcp relay information trust-all
```

Task: Configure an Interface As Trusted Sources Of the DHCP Relay Information Option.

```
(config-if)# ip dhcp relay information trusted
```

Task: Display All Interfaces That Are Configure to Be a Trusted Source for the DHCP Relay Information Option.

```
# show ip dhcp relay information trusted-sources
```

Task: Configure Per-Interface Support for the Relay Agent Information Option

```
(config-if)# ip dhcp relay information option-insert [none]
(config-if)# ip dhcp relay information check-reply [none]
(config-if)# ip dhcp relay information policy-action {drop | keep | replace}
```

See more optional tasks [here](#)

14.5. Accounting and Security

- Address vulnerability in PWLAN

14.5.1. DHCP Accounting

- add AAA and RADIUS support to DHCP configuration
- sends secure START/STOP accounting messages upon lease assignment/termination
- Restrictions:
 - AAA and RADIUS must be enabled
 - only for network pools with automatic bindings
 - **clear ip dhcp binding** or **no service dhcp** triggers accounting STOP messages

Task: Enable DHCP Accounting If a Specifier Server Group Is Configured to Run RADIUS Accounting

```
(dhcp-config)# accounting <method-list-name>
```

Task: Troubleshoot DHCP Accounting

```
debug radius accounting
debug ip dhcp server events
debug aaa accounting
debug aaa id
```

14.5.2. DHCP Secured IP Address Assignment

- Secures and synchronizes the MAC address of the client to the DHCP binding, preventing hackers from spoofing the DHCP server and taking over a DHCP lease of an authorized client

Task: Secure ARP Table Entries to DHCP Leases In the DHCP Database

```
(dhcp-config)# update arp
```



- Existing active DHCP leases will not be secured until they are renewed.

Task: Configure the Renewal Policy for Unknown Clients

```
(dhcp-config)# renew deny unknown
```



- In some usage scenarios, such as a wireless hotspot, where both DHCP and secure ARP are configured, a connected client device might go to sleep or suspend for a period of time. If the suspended time period is greater than the secure ARP timeout (default of 91 seconds), but less than the DHCP lease time, the client can awake with a valid lease, but the secure ARP timeout has caused the lease binding to be removed because the client has been inactive. When the client awakes, the client still has a lease on the client side but is blocked from sending traffic. The client will try to renew its IP address but the DHCP server will ignore the request because the DHCP server has no lease for the client. The client must wait for the lease to expire before being able to recover and send traffic again.
- To remedy this situation, use the **renew deny unknown** command in DHCP pool configuration mode. This command forces the DHCP server to reject renewal requests from clients if the requested address is present at the server but is not leased. The DHCP server sends a DHCPNAK denial message to the client, which forces the client back to its initial state. The client can then negotiate for a new lease immediately, instead of waiting for its old lease to expire.

14.5.3. DHCP Per Interface Lease Limit and Statistics

- Allows an ISP to limit the number of DHCP leases allowed on an interface.

Task: Configure a DHCP Lease Limit to Control the Number Of Subscribers on an Interface

```
(config)# ip dhcp limit lease log  
(config-if)# ip dhcp limit lease <max-users>
```

Task: Verify the DHCP Lease Limit Configuration

```
# show ip dhcp limit lease
```

Task: Clear the Stored Lease Violation Entries

```
# clear ip dhcp limit lease
```

14.5.4. DHCP Authorized ARP

Task: Disable Dynamic ARP Learning on an Interface

```
(config-if)# arp authorized
```

Task: Configure How Long an Entry Remains In the ARP Cache

```
(config-if)# arp timeout <seconds>
```

Task:

```
# show arp
```

14.5.5. ARP Auto-Logoff

- enhances DHCP authorized ARP by providing finer control and probing authorized clients to detect a logoff.

Task: Configure an Interval and Number Of Probe Retries for ARP

```
(config-if)# arp probe interval <seconds> count <number>
```

14.6. DHCP Snooping

- Prevent rogue DHCP servers from answering before the real DHCP server(s). Rogue DHCP servers would likely be interested in handing out a malicious default gateway that could intercept information before handing it off to the real default gateway.
- Prevent a malicious "client" from requesting hundreds of addresses and using up the entire pool; a DOS-style attack, where new clients would be unable to get an address.
- IP Source Guard (discussed later)
- Dynamic ARP Inspection (discussed later)

TODO

add information about option 82

Chapter 15. NAT

- Configuration Guides > IP Addressing > [NAT](#)
- [RFC 1631](#)

15.1. Purpose

- IPv4 address conservation
- can be static, dynamic or pat

15.2. Inside and Outside Address

Inside local address

Private IP address that is assigned to a host on the inside network.

Inside global address

Public IP address that represents one or more inside local IP addresses to the outside world.

Outside local address

Private IP address of an outside host as it appears to the inside network.

Outside global address

Public IP address assigned to a host on the outside network by the owner of the host.

Task: Display NAT Translation Information

```
show ip nat translations [verbose]  
show ip nat statistics
```

15.3. Static NAT

- Statically correlates the same local host to the same public IP address.
- Does not conserve IP addresses.

Task: Configure Static Translation Of Inside Source Address

```
conf t
ip nat inside source static <local-ip> <global-ip>

interface <type number>
  ip address <ip-address> <mask> [secondary]
  ip nat inside

interface type number
  ip address <ip-address> <mask>
  ip nat outside
```

15.4. Dynamic NAT Without PAT

- One local host uses an available public IP address in a pool.
- Does not conserve IP addresses.
- Timeout after period of nonuse

Task: Configure Dynamic Translation Of Inside Source Address

```
ip nat pool <name> <start-ip> <end-ip> {netmask <mask> | prefix-length <length>}
access-list <acl> permit source [<w.i.l.d>]
ip nat inside source list <acl> pool <name>

interface <type number>
  ip address <ip-address> <mask>
  ip nat inside

interface <type number>
  ip address <ip-address> <mask>
  ip nat outside
```

Task: Change Timeouts Value

```
ip nat translation <seconds>
ip nat translation udp-timeout <seconds>
ip nat translation dns-timeout <seconds>
ip nat translation tcp-timeout <seconds>
ip nat translation finrst-timeout <seconds>
ip nat translation icmp-timeout <seconds>
ip nat translation syn-timeout <seconds>
```

15.5. PAT

- Like dynamic NAT but multiple local hosts share a single public address by multiplexing TCP/UDP ports.

- Conserves IP addresses.

15.6. NAT for Overlapping Address

- Can be done with any of the first three types.
- Translates both source and destination addresses, instead of just the source (for packets going from enterprise to the Internet).

15.7. TCP Load Distribution for NAT

- Round-robin allocation of a virtual host that coordinates load sharing among real hosts.

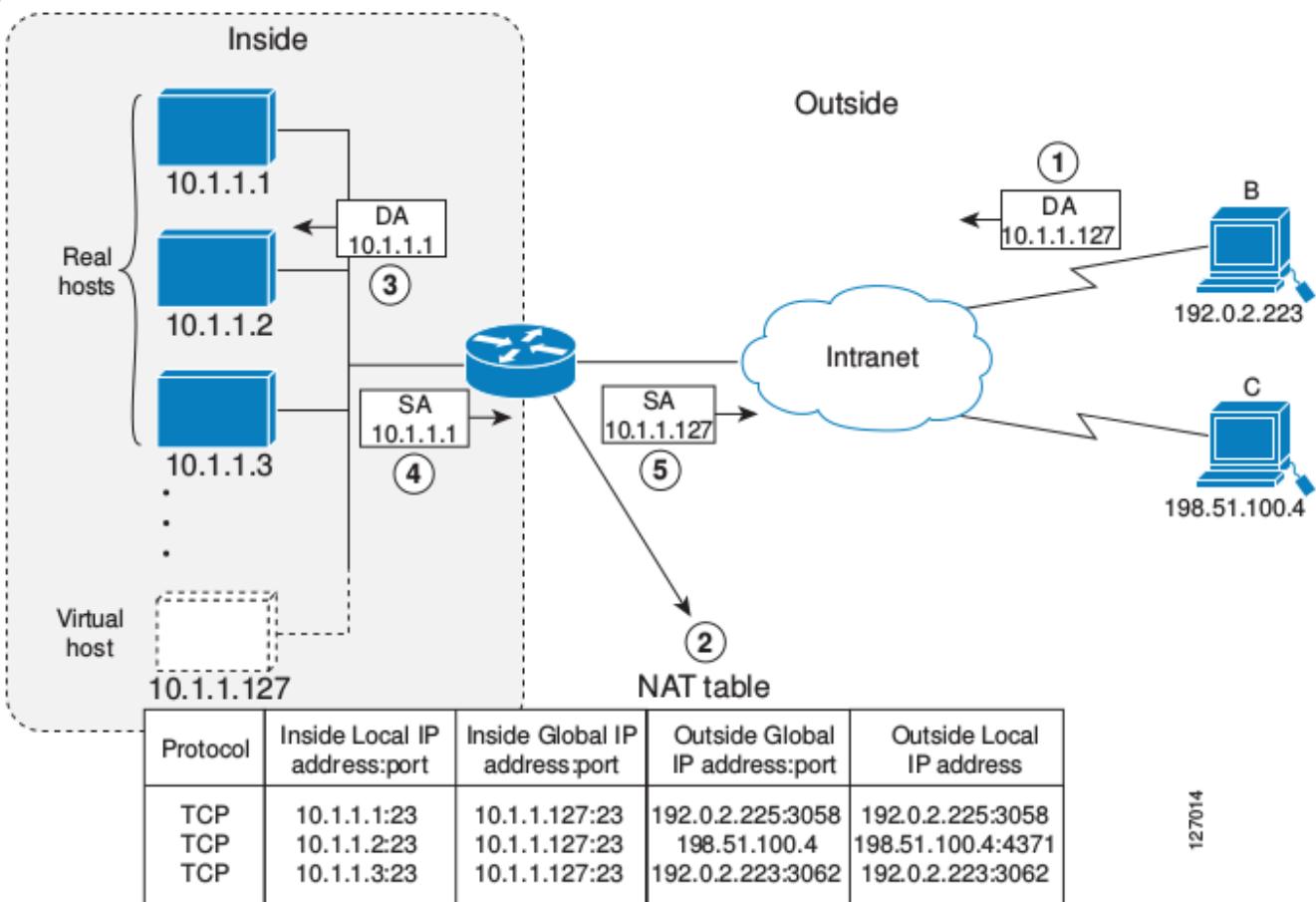


Figure 25. NAT TCP Load Distribution

127014

Task: Allow Internal Users Access to the Internet

```
ip nat pool <name> <start-ip> <end-ip> {netmask netmask | prefix-length prefix-length}
access-list number permit a.b.c.d [e.f.g.h]
ip nat inside source list number pool name overload

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside
end
```

15.8. Overlapping Networks

Configure dynamic translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using dynamic translation.

Task: Configure Overlapping Network

```
ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length}
access-list access-list-number permit source [source-wildcard]
ip nat outside source list access-list-number pool name

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside
```

15.9. Server TCP Load Balancing

```

ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length} type
rotary
access-list access-list-number permit source [source-wildcard]
ip nat inside destination-list access-list-number pool name

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside

```

Task: Clear NAT Entries Before the Timeout

```

clear ip nat translation inside global-ip local-ip outside local-ip global-ip
clear ip nat translation outside global-ip local-ip
clear ip nat translation protocol inside global-ip global-port local-ip local-port
outside local-ip local-port-global-ip global-port
clear ip nat translation {* | [forced] | [inside global-ip local-ip] [outside local-ip
global-ip]}

```

Task: Enable Syslog for Logging NAT Translations

```

ip nat log translations syslog
no logging console

```

15.10. NAT Order Of Operations

15.10.1. Inside-to-Outside

1. If IPSec Then Check Input Access List
2. Decryption - for CET (Cisco Encryption Technology) or IPSec
3. Check Input Access List
4. Check Input Rate Limits
5. Input Accounting
6. Redirect to Web Cache
7. Policy Routing
8. Routing
9. NAT Inside to Outside (Local to Global Translation)
10. Crypto (Check Map and Mark for Encryption)
11. Check Output Access List

12. Inspect (Context-Based Access Control (CBAC))
13. TCP Intercept
14. Encryption
15. Queueing

15.10.2. Outside-to-Inside

1. If IPSec Then Check Input Access List
2. Decryption - for CET or IPSec
3. Check Input Access List
4. Check Input Rate Limits
5. Input Accounting
6. Redirect to Web Cache
7. NAT Outside to Inside (Global to Local Translation)
8. Policy Routing
9. Routing
10. Crypto (Check Map and Mark for Encryption)
11. Check Output Access List
12. Inspect CBAC
13. TCP Intercept
14. Encryption
15. Queueing

Read more: [Order of operations](#)

Chapter 16. NHRP

TIP: if dmvpn phase 3, the tunnel key must be the same as the tunnel key ???

```
!! DMVPN HUB
int f0/0.123
    enc dot1q 123
    ip address 10.0.0.1 255.255.255.0
    no shut
int t123
    ip add 129.99.123.1 255.255.255.0
    tunnel source f0/0.123
    tunnel mode gre multipoint
    tunnel key 123
    ip nhrp network-id 123
    ip nhrp map multicast dynamic
    ip nhrp network-id 321

!! DMVPN SPOKE
int f0/0.123
    desc ospf
    enc dot1q 123
    ip address 10.0.0.2 255.255.255.0
int t123
    ip add 129.99.123.2 255.255.255.0
    tunnel source f0/0.123
    tunnel destination 10.0.0.1
    tunnel key 123
    ip nhrp network-id 123
    ip nhrp nhs 129.99.123.1
    ip nhrp map multicast 10.0.0.1
    ip nhrp map 129.99.123.1 10.0.0.1
```

Task: Verify That NHRP Registration Has Been Sent from Spokes to the Hub

```
R1#sh ip nhrp

129.99.123.2/32 via 129.99.123.2
    Tunnel123 created 00:08:18, expire 01:54:55
    Type: dynamic, Flags: unique registered
    NBMA address: 10.0.0.2
129.99.123.3/32 via 129.99.123.3
    Tunnel123 created 00:09:22, expire 01:54:57
    Type: dynamic, Flags: unique registered
    NBMA address: 10.0.0.3
```

Task: Troubleshoot nhrp

```
# debug nhrp
```

Chapter 17. IPv6

Configuration Guides > IP > **IPv6** - <https://tools.ietf.org/html/rfc2460>[RFC 2460]

17.1. IPv6 Header

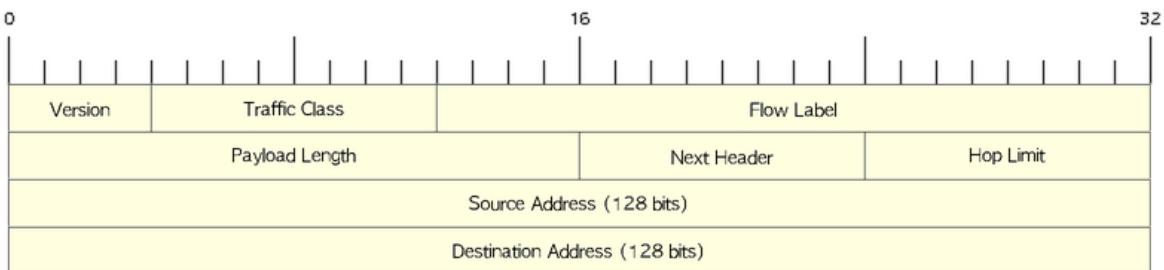


Figure 26. IPv6 Base Header Format

Comparison with IPv4

Streamlined

- Fixed length header + optional extension header
- Fragmentation fields moved out of base header
- IP options moved out of base header
- Header checksum eliminated
- Header length field eliminated
- Length field excludes IPv6 header
- Alignment changed from 32 to 64 bits

Revised

- Time to live → hop limit
- Protocol → next header
- Precedence and TOS → traffic class
- Addresses increased 32 bits → 128 bits

Extended

- Flow label field added

17.2. Traffic Class

- 6 bits for ToS
- 2 bits for ECN

17.3. Flow Label

- Sequence in a particular flow
- Originally created for giving real-time applications special service.
- when set to a non-zero value now serves as a hint to routers and switches with multiple outbound paths that these packets should stay on the same path so that they will not be reordered.
- may be used to help detect spoofed packets

17.4. Payload Length

- Size in bytes of the payload including any extension header
- Set to zero when a **Hop-by-Hop** extension header carries a **Jumbo Payload** option.

17.5. Next Header

- Specifies the type of the next header.
- usually specifies the transport layer protocol used by a packet's payload.
- When extension headers are present in the packet this field indicates which extension header follows.
- The values are shared with those used for the IPv4 protocol field
- Extension headers only examined at the destination, except for the 'hop-by-hop options'.
- If a node does not recognize a specific extension header, it should discard the packet and send a Parameter Problem message (ICMPv6 Type 4, Code 1).
- When a Next Header value 0 appears in a header other than the fixed header a node should do the same.

Table 11. Recommended order of extension header

Next Header Type	Value	Description
Hop-by-Hop Options Header	0	Read by all devices in transit network
Destination Option Header	60	Read by the final destination device
Routing Header	43	Support routing decision making
Fragment Header	44	Contains parameters of datagram fragmentation
Authentication Header	51	
Encapsulating Security Payload	50	Carries encrypted data for secure communication.
Destination Option Header	60	Read by the final destination device
Upper-Layer Header	6	TCP
Upper-Layer Header	17	UDP

Next Header Type	Value	Description
Mobility Header (currently without upper layer)	135	Used with Mobile IPv6

- Value 59 means **No Next Header**

17.5.1. Hop-by-hop options and destination options

- Hop-by-Hop Options extension headers examined by all nodes on the packet's path, including sending and receiving nodes.
- The Destination Options extension header need to be examined by the destination node(s) only.
- The extension headers are both at least 8 octets in size;
 - if more options are present than will fit in that space, blocks of 8 octets are added to the header repeatedly—containing options and padding—until all options are represented.
- the **header ext len** is the size of this header in bytes excluding the first 8 octets

TODO jumbo-grams TODO router alert option, mld multicast listener discovery

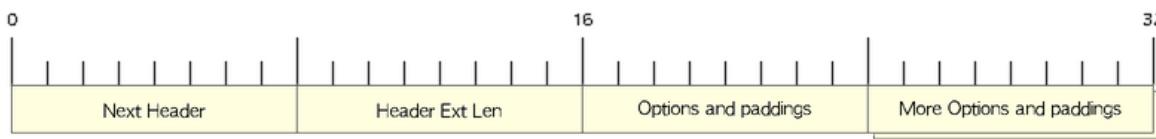


Figure 27. IPv6 Hop-By-Hop Options

17.5.2. Routing Extension Header

- Directs a packet to one or more intermediate nodes before being sent to its destination.
- At least 8 octets in size;
 - if more Type-specific Data is needed than will fit in 4 octets, blocks of 8 octets are added to the header repeatedly, until all Type-specific Data is placed.
- Routing types:
 - 0 deprecated, because of DoS
 - 1 used by the Nimrod project
 - 2 for IPv6 Mobile

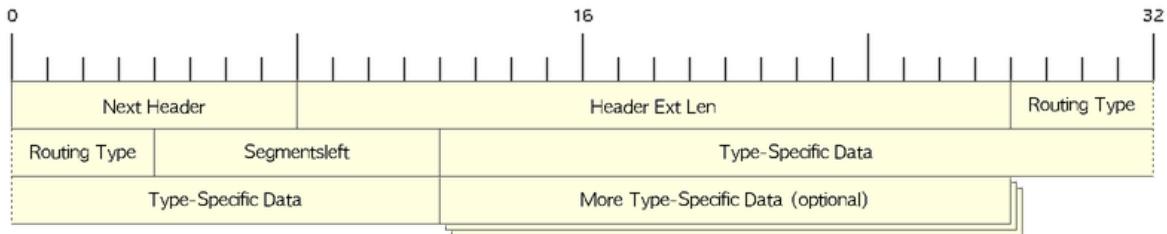


Figure 28. IPv6 Routing Options

TODO used in ipv6 mobility TODO disable ipv6 source routing to protect against DDoS

TODO .Task:

```
(config-if)# no ipv6 source-route
```

17.5.3. Fragment Extension Header

- In order to send a packet that is larger than the path MTU, the sending host splits the packet into fragments.
- The Fragment extension header carries the information necessary to reassemble the original (unfragmented) packet.

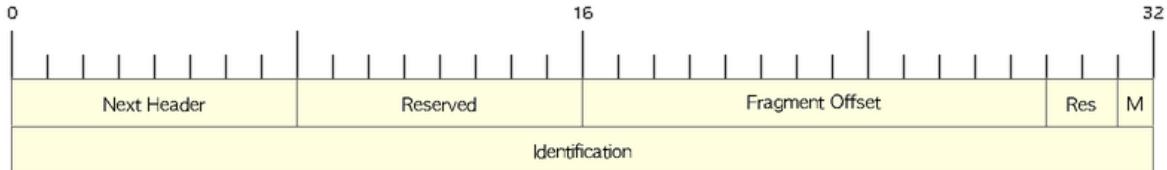


Figure 29. IPv6 Fragment Options

17.6. Fragmentation And Reassembly

- Unlike in IPv4, IPv6 routers never fragment IPv6 packets.
 - Packets exceeding the size of the maximum transmission unit of the destination link are dropped
 - The router sends a **Packet too Big ICMPv6 Type 2** message to the originating node , similarly to the IPv4 method when the Don't Fragment bit is set.
- End nodes in IPv6 are expected to perform path MTU discovery to determine the maximum size of packets to send, and the upper-layer protocol is expected to limit the payload size. However, if the upper-layer protocol is unable to do so, the sending host may use the Fragment extension header in order to perform end-to-end fragmentation of IPv6 packets. Any data link layer conveying IPv6 data must be capable of delivering an IP packet containing **1280 bytes** without the need to invoke end-to-end fragmentation at the IP layer.

! if a data link layer physically cannot deliver an ipv6 datagram of 1280 bytes in a single frame then the link layer must provide its own fragmentation and reassembly mechanism, separate from IP's own fragmentation mechanism, to ensure that a 1280-byte IP datagram can be delivered intact to the IP layer.

17.6.1. Fragmenting

- A packet containing a fragment of an original (larger) packet consists of two parts: the unfragmentable part of the original packet (which is the same for all fragments), and a piece of the fragmentable part of the original packet, identified by a Fragment Offset. The Fragment Offset of the first ("leftmost") fragment is 0.
- The unfragmentable part of a packet consists of the fixed header and some of the extension headers of the original packet (if present): all extension headers up to and including the Routing extension header, or else the Hop-by-Hop extension header. If neither extension headers are present, the unfragmentable part is just the fixed header.
- The Next Header value of the last (extension) header of the unfragmentable part is set to 44 to indicate that a Fragment extension header follows. After the Fragment extension header a fragment of the rest of the original packet follows.
- The first fragment(s) hold the rest of the extension headers (if present). After that the rest of the payload follows. Each fragment is a multiple of 8 octets in length, except the last fragment.
- Each Fragment extension header has its M flag set to 1 (indicating more fragments follow), except the last, whose flag is set to 0.

17.6.2. Re-Assembly

- The original packet is reassembled by the receiving node by collecting all fragments and placing each fragment at the right offset and discarding the Fragment extension headers of the packets that carried them. Packets containing fragments need not arrive in sequence; they will be rearranged by the receiving node.
- If not all fragments are received within 60 seconds after receiving the first packet with a fragment, reassembly of the original packet is abandoned and all fragments are discarded. If the first fragment was received (which contains the fixed header), a Time Exceeded message (ICMPv6 type 3, code 1) is returned to the node originating the fragmented packet, if the packet was discarded for this reason.
- Receiving hosts must make a best-effort attempt to reassemble fragmented IP datagrams that, after reassembly, contain up to 1500 bytes. Hosts are permitted to make an attempt to reassemble fragmented datagrams larger than 1500 bytes, but they are also permitted to silently discard any datagram after it becomes apparent that the reassembled packet would be larger than 1500 bytes. Therefore, senders should avoid sending fragmented IP datagrams with a total reassembled size larger than 1500 bytes, unless they have previous assurance that the receiver is capable of reassembling such large datagrams

17.6.3. Security

- Research has shown that the use of fragmentation can be leveraged to evade network security

controls.

- As a result, RFC 7112 requires that the first fragment of an IPv6 packet contains the entire IPv6 header chain, such that some very pathological fragmentation cases are forbidden.
- Additionally, as a result of research on the evasion of RA-Guard in RFC 7113, RFC 6980 has deprecated the use of fragmentation with Neighbor Discovery, and discouraged the use of fragmentation with Secure Neighbor Discovery (SEND).

17.7. Addressing

- 128 bits
- Represented in hexadecimal and uses 8 colon-separated fields of 16 bits.

17.7.1. IPv4 Vs IPv6

- Multiple ipv6 addresses on a logical or physical interface with equal precedence on IOS (only one primary ipv4 with optional secondary address)
- Automatic configuration of globally unique address (without the need of DHCP)
- Built-in neighbor discovery of neighbors, routers and gateways

17.7.2. Address Abbreviation Rules

- Whenever one or more successive 16-bit groups in an IPv6 address consist of all 0s, that portion of the address can be omitted and represented by two colons (::). The two-colon abbreviation can be used only once in an address, to eliminate ambiguity.
- When a 16-bit group in an IPv6 address begins with one or more 0s, the leading 0s can be omitted. This option applies regardless of whether the double-colon abbreviation method is used anywhere in the address.

2001:0001:0000:0000:00A1:0CC0:01AB:397A

2001:1:0:0:A1:CC0:1AB:397A

2001:0001::00A1:0CC0:01AB:397A

2001:1::A1:CC0:1AB:397A

17.7.3. Address Types

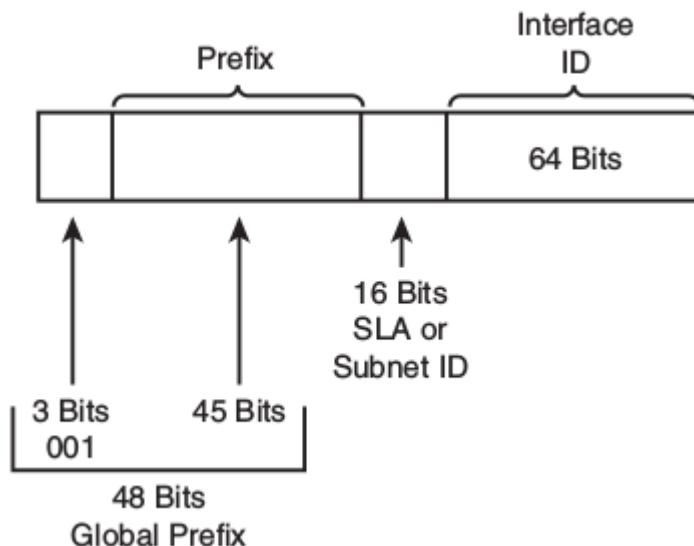
Address Type	Range	Application
Aggregatable global unicast	2000::/3	Host-to-host communication; same as IPv4 unicast.
Multicast	FF00::/8	One-to-many and many-to-many communication; same as IPv4 multicast.

Address Type	Range	Application
Anycast	Same as Unicast	Application-based, including load balancing, optimizing traffic for a particular service, and redundancy. Relies on routing metrics to determine the best destination for a particular host.
Link-local unicast	FE80::/10	Connected-link communications.
Solicited-node multicast	FF02::1:FF00:0/104	Neighbor solicitation.
Site Local	FEC0::/10	Deprecated RFC 3879
Unique Local	FC00::/7	RFC 4193 for private use (~ RFC 1918), non-routable via global BGP

Unicast

Aggregatable Global Addresses

- Begin with binary 001 (hexadecimal= 2000::/3)



Link-Local Addresses

- Starts with FE80::/10
- Equivalent of Ipv4 169.254.0.0/16
- Followed by 54 bits set to 0
- Interface ID
- only one link-local address per interface
- Routers do not forward link-local traffic to other segments.
 - non-routable between interfaces
- used for
 - SLAAC Stateless Address Auto-Configuration
 - ND Neighbor Discovery

- RD Router Discovery

IPv4-Compatible Addresses

- One option is to have first 96 bits set to 0

```
0:0:0:0:0:10:10:100:16
::10:10:100:16
::A:A:64:10
```

- ::ffff:0:0/96 prefix is designated as an IPv4-mapped IPv6 address. With a few exceptions, this address type allows the transparent use of the Transport Layer protocols over IPv4 through the IPv6 networking.

Assign an IPv6 Unicast Address to a Router Interface

Task: Enable Ipv6 on the Router

```
(config)# ipv6 unicast-routing
```

Task: Configure a Global Unicast Address

```
(config-if)# ipv6 address 2014:10:12::19:66/64
```

Router automatically configure a link local address on all IPv6 enabled interfaces. However, you can explicitly configure one

```
(config-if)# ipv6 address fe80::1 link-local
```

Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast and anycast address assigned to the interface
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2
- IPv6 redistribution ignores the “local” routes in the IPv6 routing table (the /128 host routes for a router’s own interface IPv6 addresses) whereas IPv4 has no such concept.

Multicast

IPv6 Multicast Address Format

- Begin with FF as the first octet, or FF00::/8
- The second octet specifies lifetime (permanent or temporary) and the scope (node, link, site, organization, global)

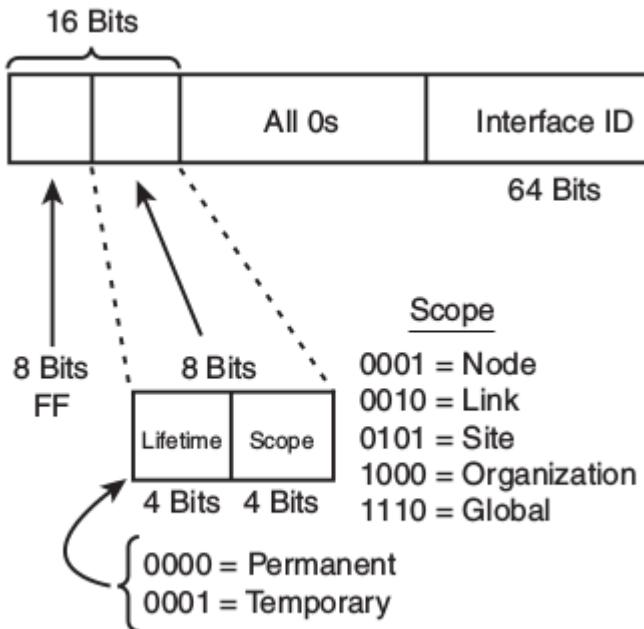


Table 12. IPv6 Multicast Well-Known Addresses

Function	Multicast Group	IPv4 Equivalent
All hosts	FF02::1	Subnet broadcast address
All Routers	FF02::2	224.0.0.2
OSPFv3 routers	FF02::5	224.0.0.5
OSPFv3 designated routers	FF02::6	224.0.0.6
EIGRP routers	FF02::A	224.0.0.10
PIM routers	FF02::D	224.0.0.13

Each router must join the **solicited-node group** (FF02::1:FF00:0000/104) for all unicast and anycast traffic. The last 24 bits come from the corresponding last 24 bits of the unicast or anycast address. The **neighbor discovery** process uses solicited-node addresses.

Anycast

Anycast addresses can be assigned to any number of hosts that provide the same service; when other hosts access this service, the specific server they hit is determined by the unicast routing metrics on the path to that particular group of servers. This provides geographic differentiation, enhanced availability, and load balancing for the service.

```
(config-if)# ipv6 address 3001:ffff::104/64 anycast
```

All IPv6 routers additionally must support the subnet router anycast address. This anycast address is a prefix followed by all 0s in the interface ID portion of the address. Hosts can use a subnet router anycast address to reach a particular router on the link identified by the prefix given in the subnet router anycast address.

The Unspecified Address

- Represented by ::
- Used as source address by an interface that has not yet learned its unicast addresses.
- Cannot be assigned to an interface
- Cannot be used as a destination address

How to Embed an RP Address Within a Multicast Group Address

RFC 2373

Given address 2001:DB*:0717::A, Follow the structure FF7X:0Y30:2001:DB8:0717::group

- FF for a multicast address
- 7 indicates that the RP address is embedded in the multicast address
- X for the multicast scope
 - 1 node-local
 - 2 link-local
 - 5 site-local
 - 8 organization-local
 - E global
 - F reserved
- 0 in the first character of the second hextet
- Y for the RP interface ID from 1 to F
- 30 for the mask for the network (0x30 = decimal 48)
- Remaining hextets for the network prefix

17.7.4. IPv6 Address Autoconfiguration

Stateful autoconfiguration

- Assigns a host its entire 128-bit address using DHCP

Stateless autoconfiguration

- Assigns a host a 64-bit prefix, and the host derives the last bit using EUI-64 process.

EUI-64 Address

- Split 48-bit MAC address in two 24-bit parts
- Place FFFE in the middle
- Set to 1 the universal/local bit (7th bit in the interface id)

Given the IPv6 prefix 2001:128:1f:633 and MAC address 00:07:85:80:71:B8, the resulting EUI-address is 2001:128:1f:633:2:07:85FF:FE80:71B8/64

```
(config-if)# ipv6 address 2001:128:1f:633::/64 eui-64
```

Modified EUI-64 Address

RFC 4941 Privacy Extensions to SLAAC

TODO check microsoft web site

17.8. Basic IPv6 Functionality Protocols

17.8.1. Neighbor Discovery

- [RFC 4861](#)
- Discover and track other IPv6 hosts on connected interfaces
- Uses ICMPv6 messages and Solicited-node multicast addresses
- Major roles
 - Stateless address autoconfiguration (detailed in [RFC 2462](#))
 - Duplicate address detection (DAD)
 - Router discovery
 - Prefix discovery
 - Parameter discovery (link MTU, hop limits)
 - Neighbor discovery
 - Neighbor address resolution (replaces ARP, both dynamic and static)
 - Neighbor and router reachability verification

Neighbor Advertisements

ICMPv6 Messages Used by ND

- Host advertises their pr
- Source addresses
- Destination addresses
- Icmp type, code: 134,0

Neighbor Solicitation

- NS messages to find the link-layer of a specific neighbor
- Source address: manual assigned or ::
- Destination address: target address or solicited-node multicast address
- ICMP type, code: 135,0
- Uses in 3 operations: duplicate address detection, neighbor reachability verification, layer 3 to

layer 2 address resolution.



IPv6 does not include ARP as a protocol but rather integrates the same functionality into ICMP as part of neighbor discovery. The response to an NS message is an NA message .

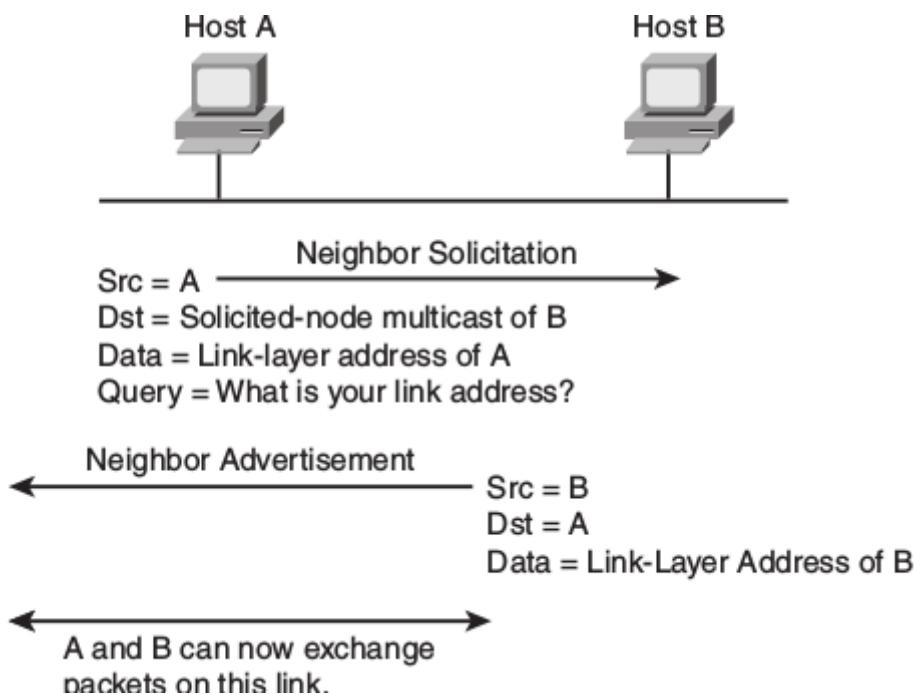


Figure 30. Neighbor Discovery Between Two Hosts

Router Advertisement

- Routers advertise their presence and link prefixes, MTU, hop limits
- Source address: router's link-local address
- Destination address: all-nodes FF02::1 for periodic broadcasts, querying host address for response
- Icmp type, code: 134,0

A Cisco IPv6 router begins sending RA messages for each of its configured interface prefixes when the **ipv6 unicast-routing** command is configured. You can change the default RA interval (200 seconds) using the command **ipv6 nd ra-interval**. Router advertisements on a given interface include all of the 64-bit IPv6 prefixes configured on that interface. This allows for stateless address autoconfiguration using EUI-64 to work properly. RAs also include the link MTU, hop limits, and whether a router is a candidate default router.

IPv6 routers send periodic RA messages to inform hosts about the IPv6 prefixes used on the link and to inform hosts that the router is available to be used as a default gateway. By default, a Cisco router running IPv6 on an interface advertises itself as a candidate default router.

Task: Prevent Router to Advertise Itself As a Default Candidate but Do Not Hide Its Presence

```
ipv6 nd ra-lifetime 0
```

Task: Hide Presence Of a Router Running IPv6

```
ipv6 nd suppress-ra
```

Router Solicitation

- Host query for the presence of routers on the link
- Source address: querying host interface, or :: if not assigned
- Destination address: FF02::2
- Icmp type, code : 133,0

At startup, IPv6 hosts can send RS messages to the all-routers multicast address. Hosts do this to learn the addresses of routers on a given link, as well as their various parameters, without waiting for a periodic RA message. If a host has no configured IPv6 address, it sends an RS using the unspecified address as the source. If it has a configured address, it sources the RS from the configured address.

Duplicate Address Detection

To verify that autoconfigured or statically address is unique, the host sends an NS message to its own autoconfigured address's corresponding solicited-node multicast address. This message is sourced from the unspecified address ::. In the target address field in the NS is the address the host seeks to verify. If an NA from another host results, the sending host knows that the address is not unique

Neighbor Unreachability Detection

2 options:

- a host sends a probe to the desired host's solicited-node multicast address and receives an RA or an NA in response.
- a host, in communication with the desired host, receives a clue from higher-layer protocol (e.g. TCP ACK)

17.8.2. ICMPv6

- RFC 2463
- Two groups of messages: error reporting messages and informational messages
- IOS implements ICMP rate limiting by setting the minimum interval between error messages and build a token bucket

Limit ICMPv6 error messages with default interval 100 ms , and default token-bucket size 10.

```
(config)# ipv6 icmp error-interval seconds ???
```

Unicast Reverse Path Forwarding

- Protects router from DoS attacks from spoofed IPv6 host address.
- Performs a recursive lookup in the ipv6 routing table to verify that the packet came in on the correct interface.

```
(config-if)# ipv6 verify unicast reverse-path
```

17.8.3. DNS

- Provides resolution of domain names
- DNS records: AAAA (RFC 1886), A6 (RFC 2874)

17.8.4. CDP

- Cisco Discovery Protocol
- Provides extensive information about the configuration and functionality of Cisco devices.

Task: Display IPv6 Information Transmitted In CDP

```
# show cdp neighbors detail
```

17.8.5. DHCPv6

- RFC 3315

Two conditions can cause a host to use DHCPv6:

- The host is explicitly configured to use DHCPv6 based on an implementation-specific setting.
- An IPv6 router advertises in its RA messages that it wants hosts to use DHCPv6 for addressing. Routers do this by setting the M flag (Managed Address Configuration) in RAs.

To use stateful autoconfiguration, a host sends a DHCP request to one of two well-known IPv6 multicast addresses on UDP port 547:

- FF02::1:2, all DHCP relay agents and servers
- FF05::1:3, all DHCP servers

The DHCP server then provides the necessary configuration information in reply to the host on UDP port 546. This information can include the same types of information used in an IPv4 network, but additionally it can provide information for multiple subnets, depending on how the DHCP server is configured.

To configure a Cisco router as a DHCPv6 server, you first configure a DHCP pool, just as in IPv4 then enable the DHCPv6 service using the **ipv6 dhcp server pool-name**

TODO stateless vs stateful managed-config-flag vs other-config-flag

17.8.6. Access Lists

Similar with IPv4 access lists except that:

- Because Neighbor Discovery is a key protocol in IPv6 networks, access lists implicitly permit ND traffic. This is necessary to avoid breaking ND's ARP-like functionality. You can override this implicit-permit behavior using deny statements in IPv6 access lists.

Task: Configure an Interface to Filter Traffic Using an Access List

```
ipv6 traffic-filter access-list-name {in | out}
```

- IPv6 access lists are always named; they cannot be numbered (unless you use a number as a name).
- IPv6 access lists are configured in named access-list configuration mode, which is like IPv4 named access-list configuration mode. However, you can also enter IPv4-like commands that specify an entire access-list entry on one line. The router will convert it to the correct configuration commands for named access-list configuration mode.

17.9. IPv6 tunneling

17.9.1. 6in4

- mechanism for migrating from IPv4 to IPv6 (RFC 4213)
- uses tunneling to encapsulate IPv6 traffic over explicitly-configured IPv4 links
 - The 6in4 traffic is sent over the IPv4 Internet inside IPv4 packets whose IP headers have the IP protocol number set to 41.
 - In 6in4, the IPv4 packet header is immediately followed by the IPv6 packet being carried. This means that the encapsulation overhead is simply the size of the IPv4 header of 20 bytes. With an Ethernet MTU of 1500 bytes, one can thus send IPv6 packets of 1480 bytes without fragmentation.
- Also referred to as proto-41 static because the endpoints are configured statically.
- generally manually configured

17.9.2. 6to4

- encapsulates the IPv6 packets into IPv4 which allows remote IPv6 networks to communicate across the IPv4 infrastructure(core network or Internet).
- The main difference between the manual tunnels and automatic 6to4 tunnels is that the tunnel is not point-to-point but it is point-to-multipoint.
- In automatic 6to4 tunnels, the IPv4 infrastructure is treated as a virtual non-broadcast multi-access (NBMA). The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.

- Point-to-multipoint 6to4 tunnels that can be used to connect isolated IPv6 sites can use addresses from the 2002::/16 prefix.

17.9.3. ISATAP

- automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6.
- Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border devices or between a border device and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks.

IPv6 supports the following types of overlay tunneling mechanisms: - Manual - GRE - IPv4-compatible - 6to4 - Intra-site Automatic Tunnel Addressing Protocol (ISATAP)

17.9.4. 6RD

IPv6 Rapid Deployment (6rd) is a stateless tunneling mechanism which allows a Service Provider to rapidly deploy IPv6 in a lightweight and secure manner without requiring upgrades to existing IPv4 access network infrastructure. While there are a number of methods for carrying IPv6 over IPv4, 6rd has been particularly successful due to its stateless mode of operation which is lightweight and naturally scalable, resilient, and simple to provision.

[Further Reading](#)

17.9.5. 6VPE

The 6PE feature is particularly applicable to Service Providers who already run an MPLS network or plan to do it. One of the Cisco 6PE advantages is that there is no need to upgrade the hardware, software or configuration of the core network. Thus it eliminates the impact on the operations and the revenues generated by the existing IPv4 traffic. MPLS has been chosen by many Service Providers as a vehicle to deliver services to customers. MPLS as a multi-service infrastructure technology is able to provide layer 3 VPN, QoS, traffic engineering, fast re-routing and integration of ATM and IP switching. It is in a very natural manner that MPLS is put to contribution to ease IPv6 introduction in existing production networks.

MPLS decoupling of the control plane and data plane provide an interesting alternative to the integration and coexistence of IPv4, IPv6 and ATM over a single infrastructure, thus fulfilling environments such as 3G networks where UMTS Release 5 needs in terms of transport: Cisco 6PE for IPv6 traffic, ATM over MPLS and regular IPv4 switching with its VPN, traffic engineering and QoS extensions. From an operational standpoint, new CEs introduction is straightforward and painless as it leverages the Layer 3 VPN scalability. Using tunnels on the CE routers is the simplest way to deploy IPv6 over MPLS networks. It has no impact on the operation or infrastructure of MPLS, and requires no changes to either the P routers (they don't have to be IPv6 aware) in the core or the PE routers connected to the customers.

6VPE is a technology that allows IPv6 VPN customers to communicate with each other over an IPv4 MPLS Provider without any tunnel setup, by having the customer VPNv6 prefixes using a v4-

mapped IPv6 address as next-hop inside the provider's network and using IPv4 LSPs between the 6VPEs. In 6VPE, labels must be exchanged between the 6VPEs for their VPv6 prefixes, which means that the VPv6 address-family must be activated on the IPv4 iBGP session between the 6VPEs.

By default, the **mpls ip propagate-ttl** command is enabled and the IP TTL value is copied to the MPLS TTL field during label imposition. To disable TTL propagation for all packets, use the **no mpls ip propagate-ttl** command. To disable TTL propagation for only forwarded packets, use the **no mpls ip propagate forwarded** command. Disabling TTL propagation of forwarded packets allows the structure of the MPLS network to be hidden from customers, but not the provider.

Further Reading <http://goo.gl/vuPAXm> <http://goo.gl/Hu78Cr>

Further Reading <http://goo.gl/xEL1XF>

17.10. IPv6 Routing

17.10.1. Static Routes

Similar to IPv4 static routes except that:

- An IPv6 static route to an interface has an administrative distance of 1, not 0 as in IPv4.
- An IPv6 static route to a next-hop IP address also has an administrative distance of 1, like IPv4.
- Floating static routes work the same way in IPv4 and IPv6.
- An IPv6 static route to a broadcast interface type, such as Ethernet, must also specify a next-hop IPv6 address because
 - IPv6 does not use ARP
 - There is no concept of proxy ARP

```
(config)# ipv6 route 2001:128::/64 2001::207:85FF:FE80:7208
```

```
show ipv6 route
```

17.10.2. OSPFv3

[implementing OSPF for IPv6](#)

17.10.3. EIGRPv6

17.11. Ospfv3

- Router id is highest ipv4 loopback, highest ipv4, or **router-id id** command

17.12. Readings

Implement tunnels

17.12.1. IPv6 General Prefix

- as shortcut
 - e.g. if organization is assigned a /32, then all prefixes should be derived from this /32
 - helps in renumbering scenarios

Task: Define an IPv6 general prefix

```
(config)# ipv6 general-prefix <name> <X:X:X:X::X>/<0-128>
```

Task: Apply an IPv6 general prefix to a link

```
(config-if)# ipv6 address <name> 0:0:0:1::1/64
```

17.13. misc

Chapter 18. CEF

- enabled by default
- can be central or distributed across multiple line cards
- Uses the FIB and the adjacency table
- Improves over process switching and fast switching methods

Switching Path	Forwarding Information stores	Load-Balancing Method
Process switching	Routing table	Per packet
Fast switching	Fast-switching cache (per flow route cache)	Per destination IP address
CEF	FIB tree and adjacency table	Per a hash of the packet source and destination

Task: Disable CEF

```
(config)# no ip cef [distributed]
```

Task: Disable CEF on an Interface

```
(config-if)# no ip route-cache cef [distributed]
```

Task: Verify That CEF Is Enabled

```
# sh cef interface <type number> [detail]  
# sh ip interface <type number>
```

18.1. RIB

- routing information base for each protocol(RIP, EIGRP, OSPF, ...)
- may contain many routes going to the same destination prefix

18.2. FIB

- best routes from possibly many routing protocols in the RIB
- contains the prefixes and next-hop address from each entry in the IP routing table structured in a way that is optimized for forwarding.
- no need for route cache maintenance because there is a one-to-one correlation between FIB entries and routing table entries
- one FIB for centralized forwarding platforms or one for each line card for distributed systems

Task: Display the FIB Contents

```
# sh ip cef
```

Prefix	Next Hop	Interface
[...]		
10.2.61.8/24	192.168.100.1	FastEthernet1/0/0
	192.168.101.1	FastEthernet6/1
[...]		

18.3. Adjacency Table

- 'adjacent' devices (single hop across a link layer)
- stores outbound interface and MAC header rewrite for adjacent nodes

Task: Display the Contents Of the Adjacency Table

```
# show adjacency [detail]
```

Protocol	Interface	Address
IPV6	Serial0/0/0	point2point(12)
IP	Serial0/0/0	point2point(13)
IP	Serial0/0/1	point2point(15)
IPV6	Serial0/0/1	point2point(10)
IPV6	FastEthernet0/0/2	FE80:24::4(12)
...		

18.3.1. Adjacency Discovery

- adjacent nodes are discovered automatically (ARP) or added manually

Table 13. Adjacency Types That Required Special Handling

Adjacency Type	Actions
Null adjacency	Packets destined for a Null0 interface are dropped. Null adjacency can be used as an effective form of access filtering.
Glean adjacency	When a device is connected to a multiaccess medium , the FIB table on the device maintains a prefix for the subnet rather than for the individual host prefixes. The subnet prefix points to a glean adjacency. A glean adjacency entry indicates that a particular next hop should be directly connected , but there is no MAC header rewrite information available. When the device needs to forward packets to a specific host on a subnet , CEF requests an ARP entry for the specific prefix , ARP sends the MAC address , and the adjacency entry for the host is built.

Adjacency Type	Actions
Punt adjacency	The device forwards packets requiring special handling or packets sent by features not yet supported in CEF switching paths to the next higher switching level for handling.
Discard adjacency	The device discards the packets.
Drop adjacency	The device drops the packets.

18.3.2. Unresolved Adjacency

When a link-layer header is prepended to a packet, the FIB requires the prepended header to point to an adjacency corresponding to the next hop. If an adjacency was created by the FIB and not discovered through a mechanism such as ARP, the Layer 2 addressing information is not known and the adjacency is considered incomplete or unresolved. Once the Layer 2 information is known, the packet is forwarded to the RP, and the adjacency is determined through ARP. Thus, the adjacency is resolved.

18.4. CEF Load Balancing

- per-destination (default)
- per-packet: round-robin method over multiple links

Task: Disable Per-Destination

```
(config-if)# no ip load-sharing per-destination
```

Task: Enable Per-Packet Load Balancing

```
(config-if)# ip load-sharing per-packet
```

Task: Select CEF Load Balancing Algorithm

```
(config)# ip cef load-sharing algorithm {original | universal | tunnel | include-ports  
[source | destination | source destination] }
```

Original algorithm

- produces distortions in load sharing across multiple routers because the same algorithm is used on every router.

Universal algorithm

- allows each router on the network to make a different load sharing decision for each source-destination address pair,
- avoids original CEF polarization
- Use a randomly generated Universal ID as seed for the hash function
- default



Tunnel algorithm

- when there are only a few source and destination pairs

Include-ports algorithm

- uses Layer 4 source and destination ports in the load-balancing decision.
- benefits traffic streams running over equal cost paths that are not load shared because the majority of the traffic is between peer addresses that use different port numbers, Real-Time Protocol (RTP) streams.

GTP-U TEID-Based ECMP Load-Balancing Algorithm

- for Cisco IOS XE Software
- for mobile devices

Task: Specify custom ID to be used as in the Hash function of the universal algorithm

```
(config)# ip cef load-sharing algorithm universal <id>
```

18.5. Misc

C4K_L3HWFORWARDING-2-FWDCAMFULL

sh platform hardware ip route summary

%MLSCEF-DFC4-7FIB_EXCEPTION: FIB TCAM exception

when BGP PIC enabled, CEF recursion is disabled when next-hop is learned via /32 mask or next-hop is directly connected.

Chapter 19. BFD

Configuration Guides > IP Routing > Bidirectional Forwarding Detection

- [RFC 5880](#)

independently of media, data protocols, and routing protocols.

- provides a low-overhead, sub-second method of detecting failures in the forwarding path between two adjacent routers, including the interfaces, data links, and forwarding engines.
- detection protocol enabled at the interface and protocol levels.
 - supports BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers.
 - must be configured on both systems (or BFD peers)
- benefits of implementing BFD over reduced timer mechanisms for EIGRP, IS-IS, and OSPF:
 - sub-second failure detection (vs 1 or 2 seconds)
 - not tied to any particular routing protocol or media
 - less CPU-intensive because some parts of BFD can be distributed to the data plane (vs control plane)
- always run in a unicast, point-to-point mode.
- UDP
 - BFD control packets sourced from 49152 and sent to 3784.
 - BFD echo packets sourced from 3785 and sent to 3785

19.1. BFD Control Packet Format

- Mandatory Control section + Optional Authentication section

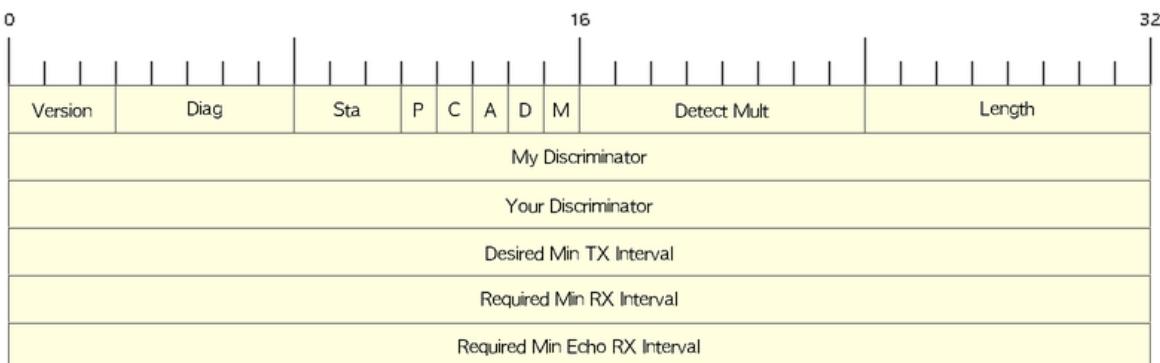


Figure 31. BFD Control Packet Format

Diagnostic Code(Diag)

local system's reason for the last change in session state.

- 0 No Diagnostic
- 1 Control Detection Time Expired
- 2 Echo Function Failed
- 3 Neighbor Signaled Session Down
- 4 Forwarding Plane Reset
- 5 Path Down
- 6 Concatenated Path Down
- 7 Administratively Down
- 8 Reverse Concatenated Path Down
- 9-31 Reserved for future use

State (Sta)

Current BFD session state as seen by the transmitting system.

- 0 AdminDown
- 1 Down
- 2 Init
- 3 Up

Poll (P)

- If set, the transmitting system is requesting verification of connectivity, or of a parameter change, and is expecting a packet with the Final (F) bit in reply.

Fnal (F)

- If set, If set, the transmitting system is responding to a received BFD Control packet that had the Poll (P) bit set.

Control Plane Independent ©

- If set, the transmitting system's BFD implementation does not share fate with its control plane (in other words, BFD is implemented in the forwarding plane and can continue to function through disruptions in the control plane).

Authentication Present (A)

- If set, the Authentication Section is present and the session is to be authenticated

Demand (D)

- If set, Demand mode is active in the transmitting system (the system wishes to operate in Demand mode, knows that the session is Up in both directions, and is directing the remote system to cease the periodic transmission of BFD Control packets).

Multipoint (M)

- Reserved for future point-to-multipoint extensions to
- Must be zero on both transmit and receipt.

Detection Time Multiplier (Detect Mult)

- The negotiated transmit interval, multiplied by this value, provides the Detection Time for the receiving system in Asynchronous mode.

Length

- Length of the BFD Control packet, in bytes.

My Discriminator

Unique, nonzero discriminator value generated by the transmitting system

- Used to demultiplex multiple BFD sessions between the same pair of systems.

Your Discriminator

- The discriminator received from the corresponding remote system. This field reflects back the received value of My Discriminator, or is zero if that value is unknown.

Desired Min TX Interval

minimum interval in microseconds, that the local system would like to use when transmitting BFD Control packets, less any jitter applied

- The value zero is reserved.

Required Min RX Interval

minimum interval, in microseconds, between received BFD Control packets that this system is capable of supporting, less any jitter applied by the sender.

- If this value is zero, the transmitting system does not want the remote system to send any periodic BFD Control packets.
- Required Min Echo RX Interval: minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting, less any jitter applied by the sender .
- If this value is zero, the transmitting system does not support the receipt of BFD Echo packets.

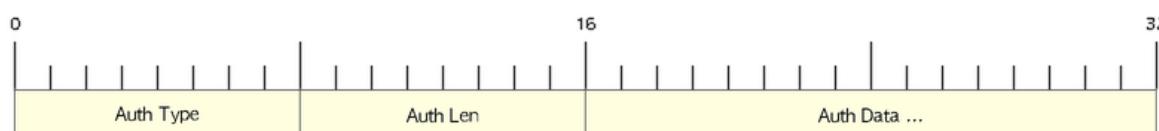


Figure 32. Optional Authentication Section

Auth Type

- 0 Reserved
- 1 Simple Password

- 2 Keyed MD5
- 3 Meticulous Keyed MD5
- 4 Keyed SHA1
- 5 Meticulous Keyed SHA1
- 6-255 Reserved for future use

Auth Len

length, in bytes, of the authentication section, including the Auth Type and Auth Len fields.

19.2. BFD operating modes

Asynchronous mode

The systems periodically send BFD Control packets to one another, and if a number of those packets in a row are not received by the other system, the session is declared to be down.

- requires half as many packets to achieve a particular Detection Time as does the Echo function.
- used when the Echo function cannot be supported for some reason.

Demand mode

It is assumed that a system has an independent way of verifying that it has connectivity to the other system. Once a BFD session is established, such a system may ask the other system to stop sending BFD Control packets, except when the system feels the need to verify connectivity explicitly, in which case a short sequence of BFD Control packets is exchanged, and then the far system quiesces. Demand mode may operate independently in each direction, or simultaneously.

- useful in situations where the overhead of a periodic protocol might prove onerous, such as a system with a very large number of BFD sessions.
- useful when the Echo function is being used symmetrically.
- may not be used when the path round-trip time is greater than the desired Detection Time, or the protocol will fail to work properly

Echo mode

- Enabled by default,
 - can be used with both Asynchronous and Demand mode
 - can be disabled to run independently in each direction.
 - Described as *without asymmetry* when it is running on both sides (both BFD neighbors are running echo mode).
- works asynchronously between 2 BFD neighbors R1 and R2
 - R1 sends an echo packet (instead of a control packet) to R2, formatted as:

```
L3 Source: R1 (192.168.12.1)
L3 Destination: R1 (192.168.12.1)
MAC Source: Itself (000c.298f.aca3)
MAC Destination: (000c.29cf.21ff)
```

- R2's receives this packet, sees this packet, and CEF-switches it straight back to R1!
- In this fashion, R1 knows that R2 is reachable.
- R2 would perform similar behavior towards R1, for its own echo process.
- Control-plane CPU-intensive packets are still sent, but they are sent at the "slow timers" speed.
 - Before using BFD echo mode, disable ICMP redirect messages (**no ip redirects** command) to avoid high CPU utilization.
 - Don't enable both BFD echo mode and uRPF. The session will flap

Task: Disable BFD Echo Mode Without Asymmetry

```
(config)# no bfd echo
```



No echo packets will be sent by the router, and the router will not forward BFD echo packets that are received from any neighbor routers.

Task: Configure the BFD slow timer.

```
(config-if)# bfd slow-timer <milliseconds>
```

19.3. BFD Session Parameters on the Interface

interval

Rate in milliseconds (50..999) at which BFD control packets will be sent to BFD peers.

min_rx

Rate in milliseconds (50..999) at which BFD control packets will be expected to be received from BFD peers.

multiplier

Number (3..50) of consecutive BFD control packets that must be missed from a BFD peer before BFD declares that the peer is unavailable and the Layer 3 BFD peer is informed of the failure.

Task: Enables BFD on the interface.

```
(config-if)# bfd interval <milliseconds> min_rx <milliseconds> multiplier <interval-multiplier>
```

- The bfd interval configuration is removed when the subinterface on which it is configured is removed.
- The bfd interval configuration is not removed when:
 - an IPv4 address is removed from an interface
 - an IPv6 address is removed from an interface
 - IPv6 is disabled from an interface
 - an interface is shutdown
 - IPv4 CEF is disabled globally or locally on an interface
 - IPv6 CEF is disabled globally or locally on an interface



19.4. BFD Support for Dynamic Routing

- BFD has no neighbor detection.
 - When the routing protocol needs to monitor a neighbor, it informs BFD, and BFD establishes the neighbor relationship at that point.
- Various routing protocols can piggyback a single BFD session.
 - If you have BGP and EIGRP running between the same two subnets on the same two routers, there's no need to have two BFD sessions for checking the same exact topology.
- enabled globally at the router level or a per interface basis

Task: Enable BFD for all interfaces participating in the routing process

```
(config-router)# bfd all-interfaces
```

Task: Enable BFD for all interfaces participating in the routing process. Use address-family interface configuration mode

```
(config-router-af-interface)# bfd
```

Task: Configure BFD for BGP

```
(config-router)# neighbor <ip-address> fall-over bfd
```

Task: Enable HSRP support for BFD on the interface.

```
(config-if)# standby bfd
```



CEF must be enabled

Task: Display a line-by-line listing of existing BFD adjacencies

```
# sh bfd neighbors
```

Sample Output



OurAddr	NeighAddr	LD/RD	RH	Holdown(mult)	State	Int
172.16.10.1	172.16.10.2	1/6	1	260 (3)	Up	Fa0/1

Task: Display a line-by-line listing of existing BFD adjacencies with details

```
# sh bfd neighbors details
```

Sample Output



```
NeighAddr LD/RD RH/RS State Int
10.1.1.2 1/1 1(RH) Up Et0/0
Session state is UP and not using echo function.
OurAddr: 10.1.1.1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3 Received MinRxInt:
50000, Received
Multiplier: 3 Holddown (hits): 150(0), Hello (hits): 50(2223) Rx Count:
2212, Rx Interval
(ms) min/max/avg: 8/68/49 last: 0 ms ago Tx Count: 2222, Tx Interval
(ms) min/max/avg:
40/60/49 last: 20 ms ago Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: CEF Stub
Uptime: 00:01:49
Last packet: Version: 0 - Diagnostic: 0
I Hear You bit: 1 - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 3 - Length: 24
My Discr.: 1 - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 50000
```



Read [Jeff Kronlage's blog on BFD](#)

19.5. BFD Support for Static Routing

- Supports RIPv2, EIGRP, OSPF, IS-IS, BGP, HSRP

Task: Specifies a static route BFD neighbor.

```
(config)# ip route static bfd <interface-type number> <ip-address> [group <group-name> [passive]]
```



The **interface-type**, **interface-number**, and **ip-address** arguments are required because BFD support exists only for directly connected neighbors.

Task: Displays information about the static BFD configuration from the configured BFD groups and nongroup entries

```
# sh ip static route bfd
```

19.6. BFD Templates for Multi-Hop

- Template can be used to define timers and authentication independently from the interface
- use **bfd map** to associate the template with unique source-destination address pairs for multihop BFD sessions.

Task: Configure a BFD template

```
configure terminal  
bfd-template multi-hop <template-name>  
interval min-tx <milliseconds> min-rx <milliseconds> multiplier <multiplier-value>  
authentication <authentication-type> keychain <keychain-name>
```

Task: Configure a BFD Map

```
(config)# bfd mapipv4 vrf <name> <destination> <length> <source-address> <length>  
<template-name>
```

19.7. BFD Multihop Support for IPv4 Static Routes



The following section can be skipped

- Enables detection of IPv4 network failure between paths that are not directly connected.
 - If a BFD session is up, IPv4 static routes that are associated with IPv4 static BFD configuration are added to a routing table. If the BFD session is down, the routing table removes all associated static routes from the routing table.
- Applicable on different kinds of interfaces such as physical, subinterface, and virtual tunnels and across intra-area and interarea topologies.

19.7.1. BFDv4 Associated Mode

In BFDv4 associated mode, an IPv4 static route is automatically associated with an IPv4 static BFDv4 multihop destination address if the static route next hop exactly matches the static BFDv4 multihop destination address.

The state of the BFDv4 session is used to determine whether the associated IPv4 static routes are added in the IPv4 RIB. For example, static routes are added in the IPv4 RIB only if the BFDv4 multihop destination is reachable, and the static routes are removed from the IPv4 RIB if the BFDv4 multihop destination subsequently becomes unreachable.

19.7.2. BFDv4 Unassociated Mode

In unassociate mode, a BFD neighbor is not associated with a static route, and the BFD sessions are requested if the IPv4 static BFD is configured.

Unassociated mode is useful in the following scenario:

- Absence of an IPv4 static route—This scenario occurs when a static route is on device A, and device B is the next hop. In associated mode, you must create both a static BFD multihop destination address and a static route on both devices to bring up the BFDv4 session from device B to device A. Specifying the static BFD multihop destination in unassociated mode on device B avoids the need to configure an unwanted static route.

Task: Configuring BFD Multihop IPv4 Static Routes

```
# ip route static bfd <multihop-destination-address> <multihop-source-address>
unassociate
```

Before you begin the configuration



- Specify a BFD destination address which is same as the IPv4 static route next hop or gateway address.
- Configure a BFD map and a BFD multihop template for an interface on the device. The destination address and source address configured for a BFD map must match the BFD static multihop configuration and the source address must be a valid IP address configured for an interface in the routing table.

19.8. BFD on Multiple Hops

- for a destination more than one hop, and up to 255 hops, away
- Cisco IOS Release 15.1(3)S and later
- set up between a unique source-destination address pair provided by the client.
- need to configure the **bfd-template** and **bfd map** commands to create a multihop template and associate it with one or more maps of destinations and associated BFD timers. You can enable authentication and configure a key chain for BFD multihop sessions.

19.9. BFD dampening

- configures exponential delay mechanism to suppress the excessive effect of remote node reachability events flapping with BFD.
- improves the convergence time and stability throughout the network
- can be applied to all types of BFD sessions, including IPv4/single-hop/multihop, MPLS-TP, and Pseudo Wire (PW) Virtual Circuit Connection Verification (VCCV).
- can be configured at the BFD template level (both single-hop and multihop templates).
 - Dampening is applied to all the sessions that use the BFD template.
 - If you do not want a session to be dampened, you should use a new BFD template without dampening for the new session.
 - not enabled by default

Task: Configure a device to dampen a flapping BFD session.

```
(config-bfd)# dampening [ <half-life-period> <reuse-threshold> <suppress-threshold> <max-suppress-time>]
```

Chapter 20. RIP

Configuration Guides > IP Routing > Routing Internet Protocol

- Distance vector protocol
- Transport: UDP 520
- Update destination:
 - Broadcast 255.255.255.255 for RIPv1
 - Multicast 224.0.0.9 for RIPv2
- Full updates every 30 seconds
- Triggered updates
- Multiple routes to the same subnet with equal metric:
 - Default = 4
 - Configured with **ip maximum-paths *n***
- Metric: hop count with
 - 1 signifying a directly connected network of the advertising router
 - 16 signifying an unreachable network.
- AD: 120
- Supports CIDR, VLSM, authentication
- Subnet mask included in route entry
- Split horizon (without poison reverse on Cisco)
- Administrative distance: 120
- Route tags when routes are redistributed into RIP
- Can advertise a next-hop router that is different from itself
 - Not implemented in Cisco IOS
- Does not keep a separate topology table
- Does not form neighbor relationship

20.1. RIP Messages

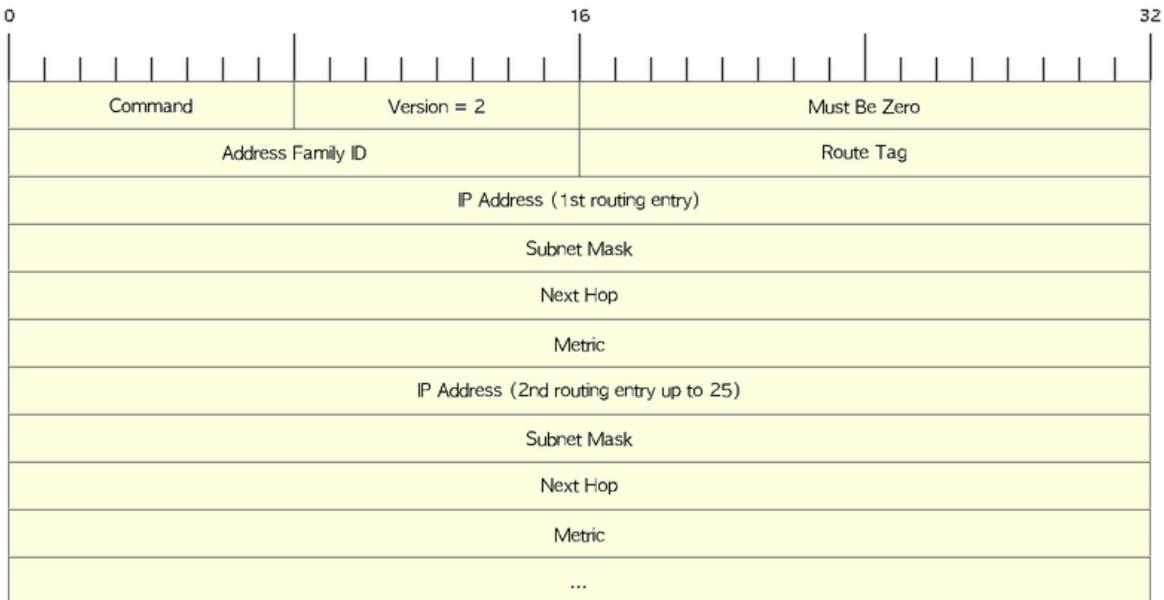


Figure 33. RIP Header Format

20.2. Request Message

- Command value = 1
- ask a neighbor to send a partial or a full RIP update immediately,
- Do not wait for the Update timer to expire
- Full RIP update
 - if one routing entry with AF = 0 and metric = 16
 - sent when RIP process start, RIP-enabled interfaces comes up, or **clear ip route**
- Partial update
 - if one or more route entry
 - Seldom used in Cisco IOS

20.3. Response Message

- Command value = 2

20.4. Default RIP Configuration

- Version : 1
- Auto-summary : enable
- Authentication : disable
- Authentication mode: text
- Split-horizon : enable

- Interpacket delay : no

20.5. Basic Configuration

```
(config)#router rip
(config-router)#version 2
(config-router)#network 10.0.0.0
(config-router)#no auto-summary
```

20.6. Version

Task: Specify the RIP Version Globally

```
(config-router)# version {1 | 2}
```

Task: Configure an Interface to Send Only a RIPv2 Packets

```
(config-if)ip rip send version [1] [2]
```

Task: Configure an Interface to Receive Only a RIPv2 Packets

```
(config-if)ip rip receive version [1] [2]
```

20.7. Authentication

When authentication is enabled,

- The maximum number of advertised prefixes is reduced to 24.
- The first route entry in each RIPv2 message would be carrying 20 bytes of authentication data.
- If cryptographic authentication methods are used, further authentication data is placed after the entire RIPv2 message.

Task: Enable RIP Authentication

```
(config-if)# ip rip authentication key-chain <name>
(config-if)# ip rip authentication mode {text | md5}
```



Use **show key chain** to spot invisible blank space after passwords

20.8. Summarization

- Default: auto-summarization
 - Summarizes prefixes to the classful network boundaries when classful network boundaries

are crossed.

- Supernet advertisement not allowed
 - E.g. **ip summary-address rip 10.0.0.0 252.0.0.0**

Task: Disable Automatic Route Summarization

```
(config-router)# no auto-summary
```

Task: Summarize a Prefix

```
(config-if)# ip summary-address rip <ip-address> <mask>
```

20.9. Route Updates

Task: Disable Sending RIP Updates on an Interface but Continue to Receive the Update

```
(config-if)# passive-interface { default | <type number>}
```

Task: Disable the Validation Of the Source IP Address Of Incoming RIP Routing Updates

```
(config-router)# no validate-update-source
```

Task: Send Updates As Broadcast

```
(config-if)# ip rip v2-broadcast
```

Task: Send Updates As Unicast

```
(config-router)# neighbor <ip-address>
```



the **neighbor** statement does not automatically suppress the sending of the broadcast or multicast update. The additional **passive-interface** is required.

20.10. Route Filtering

Task: Stop Advertising a Route with a Prefix-List

```
(config-router)# distribute-list prefix-list <name> {in | out}
```

Task: Filter Out RIP Routes with Extended Access Lists

```
(config-router)# distribute-list <extended-acl> {in|out} [<interface-id>]
```



- The source field in the ACL matches the update source of the route
- The destination field represents the network address

20.11. Route Metric

- 16 unreachable network
- RIPv2 adds 1 to the route metric while sending updates.
 - RIPNg and EIGRP increment metric when they receive updates
- maximum routes with same metric to the same subnet
 - 4 by default

Task: Add an Offset to Incoming and Outgoing Metrics to RIP Routes

```
(config-router)# offset-list <acl> {in | out } <offset> [<interface-type-number>]
```

20.12. Split Horizon

Task: Disable Split Horizon

```
(config-if)# no ip split-horizon
```

20.13. Interpacket Delay for RIP Updates

- Useful when high-end router send RIP updates to low-end router
- Default: 0 in range 8 to 50 milliseconds

Task: Configure Interpacket Delay

```
(config-if)# output-delay <milliseconds>
```

20.14. Rip Optimization Over WAN

Task: Enable Triggered Extensions for RIP

```
(config)# int serial <controller-number>
(config-if)# ip rip triggered
```

20.15. Timers

Task: Configure RIP Timers

```
(config-router)# timers basic <update> <invalid> <holddown> <flush> [<sleepetime>]
```

Update timer

- Interval between updates.
- Default: 30 seconds

Invalid After timer

- Time in seconds after which a route is declared invalid.
- Default: 180 seconds
- Reset after update is received
- Should be at least 3 times the update timer.
- Invalid routes are still used for forwarding packets

Holdown timer

- Interval during which routing information about better paths is suppressed.
- Default: 180 seconds
- Should be at least 3 times the update timer
- The route is marked inaccessible and advertised as unreachable.
- Holdown routes are still used for forwarding packets

Flush After timer

- Amount of time that must pass before a route is removed from the RIB.
- Default: 240 seconds
- Starts at the same time than Invalid After timer
- Cisco IOS checks this timer only after the Invalid After timer expired
 - No consequence If Flush timer < Invalid Timer

Sleep time

- Amount of time for which routing updates will be postponed.

Task: Specify a Default Update Interval on an Interface

```
(config-if)# ip rip advertise <seconds>
```



- The command above overrides the update timers set by **timers basic** command.

Chapter 21. EIGRP

- classless protocol (VLSM, summarization)
- multiprotocol support (ipv4, ipv6, ipx, appletalk,)
- uses its own transport protocol
 - IP protocol 88: RTP
 - Uses multicast to 224.0.0.10 and unicast
- Administrative distance : 90 internal routes, 5 summary routes , 170 external routes
- Forms active neighbor adjacencies
- DUAL for loop-free topology and fast convergence
- granular metric
- unequal cost load balancing
- summarization
- Supports MD5 and SHA based authentication
- [rfc7868](#)

21.1. EIGRP Packet Format

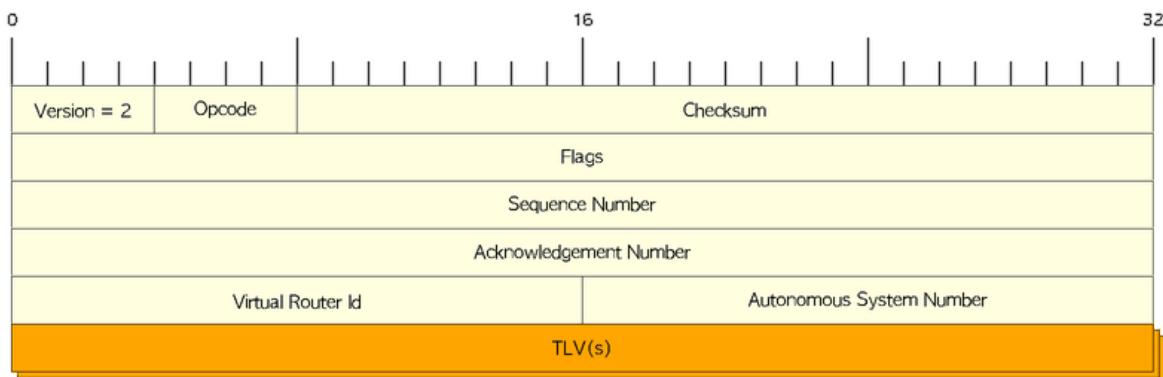


Figure 34. EIGRP Packet Format

Opcode

EIGRP packet type

- 1 = Update, 3 = Query, 4 = Reply, 5 = Hello/Ack, 10 = SIA Query, 11 = SIA Reply.
- Other types have been allocated for different, mostly unimplemented purposes, or are obsolete

Checksum

- based on the entire EIGRP packet excluding the IP header.

Flags

- 0x1 = Init (used during initial adjacency buildup)
- 0x2 = Conditional Receive (used by RTP to allow this message to be received only by a subset of receivers)
- 0x4 = Restart (indicates that a router has restarted)
- 0x8 = End-of-Table (indicates that the transmission of the entire EIGRP database is complete).

Sequence

- Facilitates orderly delivery of packets

Acknowledgement

- sequence number of the last packet heard from the neighbor to which this packet is being sent.
- A Hello packet with a nonzero ACK field will be treated as an ACK packet rather than as a Hello.
- only unicast because acknowledgments are never multicasted.

Virtual Router Id

- 0x1 = Unicast Address Family
- 0x2 = Multicast Address Family
- 0x8000 = Unicast Service Address Family (in service advertisement framework)

TLV

- Field used to carry route entries as well as provide EIGRP DUAL information.
 - 0x0001 EIGRP Parameters (General TLV Types)
 - 0x0002 Authentication Type (General TLV Types)
 - 0x0003 Sequence (General TLV Types)
 - 0x0004 Software Version (General TLV Types)
 - 0x0005 Next Multicast Sequence (General TLV Types)
 - 0x0102 IPv4 Internal Routes (IP-Specific TLV Types)
 - 0x0103 IPv4 External Routes (IP-Specific TLV Types)
 - 0x0402 IPv6 Internal Routes (IP-Specific TLV Types)
 - 0x0403 IPv6 External Routes (IP-Specific TLV Types)
 - 0x0602 Multi Protocol Internal Routes (AFI-Specific TLV Types)
 - 0x0603 Multi Protocol External Routes (AFI-Specific TLV Types)

21.2. EIGRP Messages

- Unreliable packets: Hello and Ack (with Seq=0)
- Reliable packets: Update, Query/Reply, SIA-Query/SIA-Reply
 - sent with non-zero SEQ

- Must be ACK
- are retransmitted at most 16 times for a max window of 5 seconds

Hello

- Opcode = 5
- Multicast to 224.0.0.10 or FF02::A
- unicast to static neighbors
- Do not require acknowledgement
- Can be used as Ack if sent without data
- every 5 seconds or 60 seconds on NBMA interfaces with < 1 Mbps bandwidth
- Non-reliable

Ack

- unicast in response to Update, Query, Reply, SIA-Query, and SIA-Reply packets
- contains a nonzero acknowledgement number set to the Sequence number of the reliable packet being acknowledged.
- uses the same Opcode as the Hello packet
- Non-reliable

it is allowed to use any unicast reliable packet to also carry an acknowledgment number. If a router has both a unicast reliable packet to send to a neighbor and also needs to acknowledge a previously received reliable packet from that neighbor, the sequence number of the received reliable packet can be sent along with the outbound reliable packet in its Acknowledgment number field. It is not necessary to send a standalone ACK in this case; the unicast reliable packet carrying a nonzero Acknowledgment number field will be processed by its recipient both by its true type and as an ACK.



Update

- reliable
- unicast during a new adjacency buildup, Update packets are unicasted between the newly discovered neighbors.
 - In specific cases, when multiple new neighbors are detected on a single multiaccess interface in a short time span, EIGRP might choose to synchronize to them using multicasts for efficiency reasons (for example, when a hub router in a DMVPN network starts and detects tens or hundreds of spoke routers).
- multicast after routers have fully synchronized
- unicast if a neighbor does not acknowledge the arrival of an Update packet
- always unicasts on point-to-point interfaces and for statically configured neighbors

Query

- Opcode = 3

- reliable
- multicast unless in response to a received query

Reply

- Opcode = 4
- unicast
- indicates that it does not need to go into Active state because it has a FS

Request

- unicast or multicast
- get specific info from neighbors
- used in route server applications

SIA-Query

- Opcode = 10
- unicast
- used during a prolonged diffusing computation to verify whether a neighbor that has not yet sent a Reply to a Query is truly reachable and still engaged in the corresponding diffusing computation. The SIA-Query packet is used to ask a particular neighbor to confirm that it is still working on the original Query. If the neighbor is reachable and is still engaged in the diffusing computation for the destination specified in the SIA-Query, it will immediately respond with an SIA-Reply packet. As a result, the timer that governs the maximum time a diffusing computation is allowed to run is reset, giving the computation extra time to finish

SIA-Request

- Opcode = 11
- unicast

Task: Show Statistics About Messages Sent and Received

```
# show ip eigrp traffic

EIGRP-IPv4 VR(CCIE) Address-Family Traffic Statistics for AS(1)
Hellos sent/received: 1132/6090
Updates sent/received: 169/428
Queries sent/received: 0/0
Replies sent/received: 0/0
Acks sent/received: 74/191
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0
Hello Process ID: 246
PDM Process ID: 244
Socket Queue: 0/10000/7/0 (current/max/highest/drops)
Input Queue: 0/2000/7/0 (current/max/highest/drops)
```

Task: Debug EIGRP

```
debug ip eigrp packet [hello | ack | update } quey | reply]
```

21.3. Neighbors

- Discovered with Hello packets
- can be set manually
- must agree on
 - Primary IPv4 subnet
 - Autonomous System Number
 - Authentication
 - K values
- Do not need to agree on timers
 - The hold time is included in the hello packets so each neighbor should stay alive even though the hello interval and hold timers do not match.

! After a static neighbor is defined, all EIGRP multicasts on the interface through which the neighbor is reachable will be disabled. As a result, EIGRP-enabled routers will not establish an adjacency if one router is configured to use unicast (static) while another uses multicast (dynamic) on the same link. Here's another way of putting this rule: Either all neighbors on a common network segment are statically configured for each other, or none of them are.

Task: Adjust EIGRP Hello Interval

```
(config-if)# ip hello-interval eigrp <asn> <seconds>
```

Task: Adjust EIGRP Holdown Time

```
(config-if)# ip hold-time eigrp <asn> <seconds>
```

i Changing the Hello interval does not result in automatic recalculation of the Hold time. This can, under certain circumstances, result in problems with flapping adjacencies if the Hello interval is manually configured to be close or even higher than the default Hold time, without changing the Hold timer itself.

Task: Verify Neighbor Adjacencies

```
# sh ip eigrp neighbors [detail]
```

IP-EIGRP neighbors for process 1								
H	Address	Interface	Hold	Uptime	SRTT	RT0	Q	Seq
			(sec)		(ms)		Cnt	Num
1	10.10.10.3	Fa0/0	11	00:00:08	87	522	0	6
0	10.10.10.2	Fa0/0	14	00:01:54	1300	5000	0	3

! Q Cnt indicates the number of enqueued reliable packets, that is, packets that have been prepared for sending and even possibly sent but for which no ACK has been received yet from the neighbor. In a stable network, the Q Cnt value must be zero; non-zero values are normal during initial router database synchronization or during network convergence. If the Q Cnt value remains nonzero for prolonged periods of time, however, it indicates a communication problem with the neighbor.

Task: Exchange EIGRP Packets Only As Unicast

```
(config-router)# neighbor <a.b.c.d> <interface-id>
```

Task: Exchange EIGRP Packets Only As Unicast In Named Configuration

```
(config-router-af-interface)# neighbor <a.b.c.d> <interface-id>
```

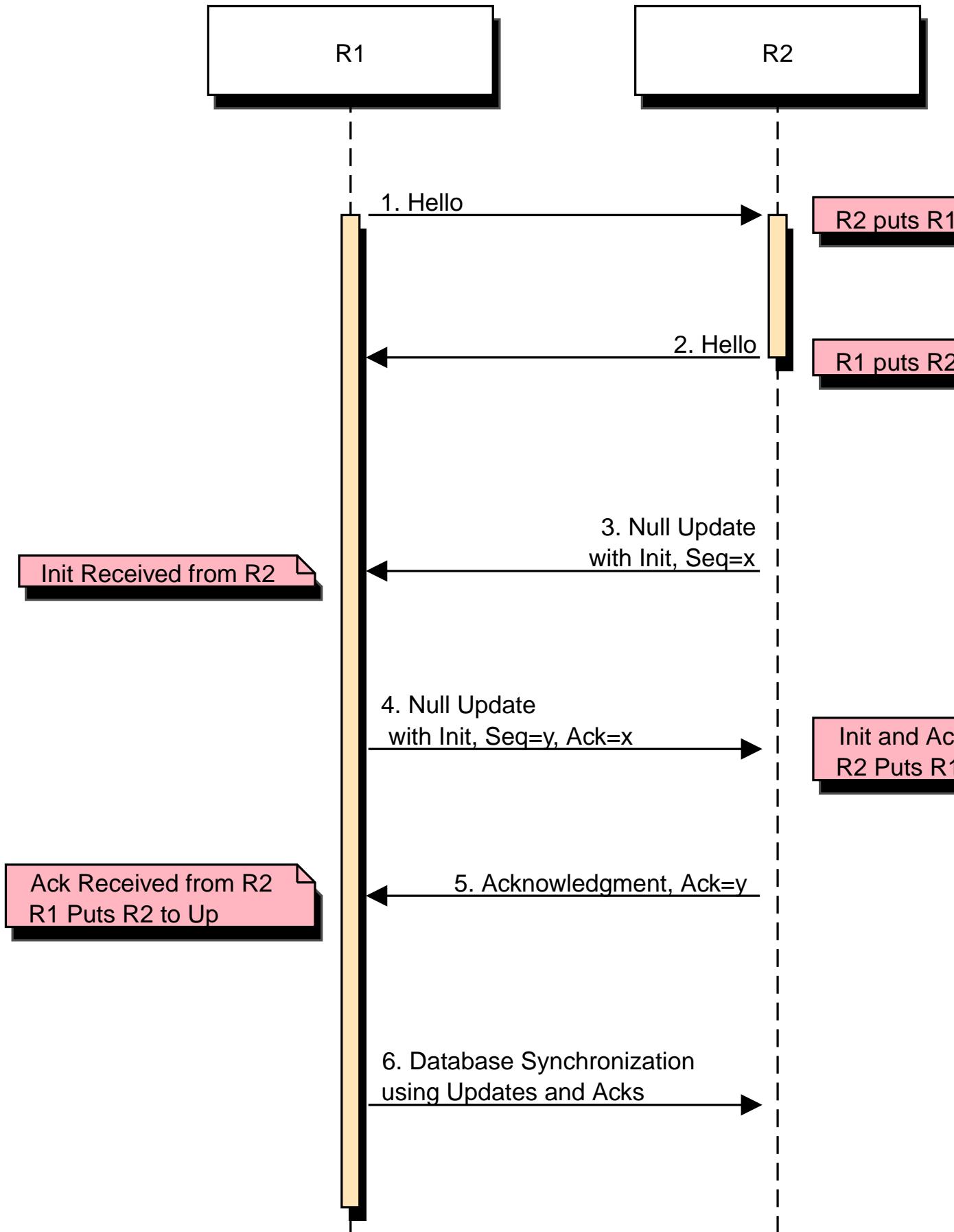


Figure 35. Adjacency Formation



EIGRP does not build peer relationships over secondary addresses. All EIGRP traffic is sourced from the primary address of the interface.

21.4. EIGRP Loop Prevention Techniques

21.4.1. Split Horizon

- Enabled by default on all interfaces

Task: Disable Split Horizon for EIGRP

```
(config-if)# no ip split-horizon eigrp <asn>
```

Task: Disable Split Horizon In Named Configuration

```
(config-router-af-interface)# no split-horizon
```

21.5. Classic Metric

Metric = $256 * ((k_1 * B \backslash a \backslash n \backslash dwidth + (k_2 * B \backslash a \backslash n \backslash dwidth) / (256 - Load) + k_3 * Delay)) * (k_5 / (Reliability + k_4))$

- Default Values: $k_1, k_2, k_3, k_4, k_5 = 1, 0, 1, 0, 0$
- The values of K must match for the neighbors to become adjacents
- EIGRP uses integer division while calculating the metric

Task: Description

```
(config-router)# metric weights
```

21.5.1. Bandwidth Metric Component

- frac {10^(7)} { "minimum Bandwidth in Kbps"}
- Range: 1 Mbps to 10 Gbps

Task: Configure the Bandwidth Of an Interface

```
(config-if)# bandwidth <kbps>
```

21.5.2. Delay Metric Component

- in tens-of-microseconds
- sum of delay on the path to the destination
- Range: 1..16_777_214

- EIGRP split horizon with poison reverse, route withdrawal uses max delay 16,777,215 to indicate an unreachable network



show ip interface displays delay in micro-seconds

Task: Configure the Delay Of an Interface

```
(config-if)# delay <tens-of-microseconds>
```

21.5.3. Reliability Metric Component

- likelihood of successful packet transmission with 0 means 0% and 255 means 100%
- Minimum value along the path
- EIGRP does not send a new update every time the reliability changes along the path
- The reliability metric of a route is just a snapshot of its then-current reliability when it was last advertised.

21.5.4. Load Metric Component

- Maximum effective Txload of the route with 255 means 100% loading
- To account for large differences in the momentary load caused by bursty traffic, IOS actually computes an exponentially weighted average over the momentary load that smooths out short-lived load swings.
- Because an interface can be differently utilized in the ingress and egress data flow direction, IOS maintains two independent load metric counters, the Txload for outgoing traffic and Rxload for incoming traffic.
- EIGRP does not send a new update every time the load changes along the path
- The load metric of a route is just a snapshot of its then-current load when it was last advertised.

21.5.5. MTU Metric Component

- minimum Maximum transmission unit
- advertised but not factored into the composite metric calculation and does not impact the best-path selection in any way

21.5.6. Hop Count Metric Component

- Default max value: 100, can be set to 255
- not factored into the composite metric calculation and does not impact the best-path selection in any way

Task: Change the max hop count

```
(config-router)# metric maximum-hops <1-255>
```

21.5.7. Routing Metric Offset Lists

TODO

When trying to manually influence EIGRP path selection through interface bandwidth/delay configuration, the modification of bandwidth is discouraged for following reasons:



- The change will only affect the path selection if the configured value is the lowest bandwidth over the entire path. Changing the bandwidth can have impact beyond affecting the EIGRP metrics. For example, QoS also looks at the bandwidth on an interface.
- EIGRP by default throttles to use 50 percent of the configured bandwidth. Lowering the bandwidth can cause problems like starving EIGRP neighbors from getting packets because of the throttling back. Configuring an excessively high bandwidth can lead EIGRP to consume more bandwidth than physically available, leading to packet drops.
- Changing the delay does not impact other protocols nor does it cause EIGRP to throttle back, and because, as it's the sum of all delays, has a direct effect on path selection.

21.6. Wide Metric

TODO Narbick figure 8.2

Metric = ((k₁* Throughput + (k₂* (Throughput)/ (256-Load) + (k₃*TotalLatency) + (k₆*ExtendedAttributes)*(k₅/(k₄ + Reliability)))

- uses by default in Named Configuration Mode
- Use one of the following commands to confirm wide metric support:
 - **sh eigrp plugins**
 - **sh eigrp tech-support**
 - **sh ip protocols**
- needs to be downscaled because the RIB can only handle 32-bit metric
 - default value: 128

Task: Change the Scale

```
(config-router)# metric rib-scale <1..255>
```



The downscaled value is not used by EIGRP in any way. EIGRP makes all its path selections based on the Wide Metrics composite value; only after a best path toward a destination is selected, its composite metric value is downscaled as the route is installed to the RIB.

21.6.1. Throughput

- ~ bandwidth
- $65536 * 10^7 / \text{bandwidth_in_kbps}$

21.6.2. Latency Metric Component

- ~ delay
- On interfaces physically operating on speeds of 1 Gbps and lower without bandwidth and delay commands, the interface delay is simply its IOS-based default delay converted to picoseconds.
- On interfaces physically operating on speeds over 1 Gbps without bandwidth and delay commands, the interface delay is computed as $10^{13} / \text{interface default bandwidth}$.
- On interfaces configured with the explicit bandwidth command and without the delay command, regardless of their physical operating speed, the interface delay is the IOS-based default delay converted to picoseconds.
- On interfaces configured with explicit delay command, regardless of their physical operating speed and the bandwidth setting, the interface delay is computed as its specified delay value converted to picoseconds, that is, $10^7 * \text{value of the delay command}$ (recall that the delay command defines the delay in tens of microseconds)

21.6.3. Reliability

- same than Classic Reliability

21.6.4. Load

- same than Classic Load

21.6.5. MTU

- same than classic MTU
- advertised but unused

21.6.6. Hop Count

- same than classic Hop Count metric component
- advertised but unused

21.6.7. Extended Metrics

- placeholders for future extensions to the composite metric computation.
- As of this writing, three extended metrics were defined: Jitter, Energy, and Quiescent Energy.
- Uses K6

21.7. Reliable Transport Protocol

- guarantees delivery in order
- Update, Query, Reply, SIA-Query, SIA-Request packets
- uses Conditional Receive for reliable and efficient multicast
 - partition all its neighbors on a multiaccess interface into two groups: a group of well-behaved neighbors that have been able to acknowledge all multicast messages sent so far and a group of “lagging” routers that have failed to acknowledge at least one transmitted reliable EIGRP packet and that must be handled individually. If EIGRP wants to continue sending the multicast packets in parallel with retransmitting the unacknowledged packets to the lagging routers as unicasts, it has to send the in-order multicast packets with a special flag saying “this packet is only for those routers that have received all multicast packets so far.”
 - accomplished by the sender first transmitting a Hello packet with two specific TLVs called the Sequence TLV and the Next Multicast Sequence TLV, often called a Sequenced Hello. The Next Multicast Sequence TLV contains the upcoming sequence number of the next reliable multicasted message. The Sequence TLV contains a list of all lagging neighbors by their IP address, in effect saying “whoever finds himself in this list, ignore the next multicast message with the indicated sequence number.” A neighbor receiving this Sequenced Hello packet and not finding itself in the Sequence TLV will know that it is expected to receive the upcoming multicast packet, and will put itself into a so-called Conditional Receive mode (CR-mode). A neighbor receiving this Sequenced Hello packet and finding itself in the Sequence TLV, or a neighbor not receiving this Hello packet at all for whatever reason will not put itself into the CR-mode. Afterward, the sending router will send the next multicast packet with the CR flag set in its Flags field. Routers in CR-mode will process this packet as usual and then exit the CR-mode; routers not in CR-mode will ignore it. As a result, the router is able to continue using multicast with those routers that have no issues receiving and acknowledging it, while making sure that the lagging neighbors won’t process the multicasts until they are able to catch up. Each lagging neighbor that has not acknowledged one or more multicast packets will be sent these packets as unicasts in their proper sequence.
 - multicast flow timer: time to wait for an ACK before declaring a neighbor as lagging and switching from multicast to unicast
 - RTO (Retransmission Time Out): the time between the subsequent unicasts
 - SRTT (Smooth Round Trip Time): average elapsed time in milliseconds, between the transmission of a reliable packet to the neighbor and the receipt of an acknowledgment.

21.8. EIGRP Autonomous System Configuration

- created with the command **router eigrp <autonomous-system-number>**
- EIGRP VPNs can be configured only under IPv4 address family. A VRF instance and route distinguisher must be defined before the address family session can be created.
- recommendation: configure the asn when the address family is configured by **router eigrp <asn> address-family** or separately using the **autonomous-system** command.

21.9. EIGRP Named Configuration

- Global params under SAFI or in **config-router-topology base** mode
- interface params in **config-router-af-interface** mode
- wide-meric scaling automatic enabled
- can be configured in IPv4 and IPv6 named configuration
- VRF instance and a RD are optional
- EIGRP IPv6 VRF-lite feature is available only in EIGRP named configuration
- EIGRP VPNs can be configured. A VRF and RD must be defined before the address-family session can be created.
- a single EIGRP routing process can support multiple VRFs. However, a single VRF can be supported by each VPN . Redistribution between VRFs is not supported.

Task: Configure a Basic EIGRP Named Configuration

```
(config)# router eigrp <virtual-instance-name>
(config-router)# address-family ipv4 [multicast] [unicast] [vrf <vrf-name>]
autonomous-system <asn>
(config-router-af)# network <a.b.c.d>
```

Task: Convert Classic Configuration to EIGRP Named Configuration

```
# eigrp upgrade-cli name
```

21.9.1. Address Family Section

```
(config-router-af)# ?
Address Family configuration commands:
  af-interface      : Enter Address Family interface configuration
  default          : Set a command to its defaults
  eigrp             : EIGRP Address Family specific commands
  exit-address-family : Exit Address Family configuration mode
  maximum-prefix    : Maximum number of prefixes acceptable in aggregate
  metric            : Modify metrics and parameters for advertisement
  neighbor          : Specify an IPv4 neighbor router
  network           : Enable routing on an IP network
  shutdown          : Shutdown address family
  timers            : Adjust peering based timers
  topology          : Topology configuration mode
```

21.9.2. Per-AF-Interface Section

```
(config-router-af-interface)# ?
Address Family Interfaces configuration commands:

add-paths          : Advertise add paths
authentication     : authentication subcommands
bandwidth-percent : Set percentage of bandwidth percentage limit
bfd                : Enable Bidirectional Forwarding Detection
dampening-change   : Percent interface metric must change to cause update
dampening-interval: Time in seconds to check interface metrics
default            : Set a command to its defaults
exit-af-interface  : Exit from Address Family Interface configuration mode
hello-interval     : Configures hello interval
hold-time          : Configures hold time
next-hop-self      : Configures EIGRP next-hop-self
passive-interface  : Suppress address updates on an interface
shutdown           : Disable Address-Family on interface
split-horizon      : Perform split horizon
summary-address    : Perform address summarization
```

21.9.3. Per-AF-Topology Configuration Section

Within the context of Multi Topology Routing, a topology is defined as a subset of routers and links in a network for which a separate set of routes is calculated. The entire network itself, for which the usual set of routes is calculated, is known as the base topology. The base topology is the default routing environment that exists prior to enabling MTR. Any additional topologies are known as class-specific topologies and are a subset of the base topology. Each class-specific topology carries a class of traffic and is characterized by an independent set of Network Layer Reachability Information (NLRI) that is used to maintain separate routing tables and FIB databases. This design allows the router to perform independent route calculation and forwarding for each topology. Multiple topologies can be used to segregate different classes of traffic, such as data, voice, and video, and carry them over different links in the same physical network, or to keep separate and independent topologies for IPv4 and IPv6 routing. Multiple topologies are not equivalent to Virtual Routing and Forwarding (VRF) tables because they share the common address space, and they are not intended to provide address conservation or reuse.

EIGRP is capable of keeping separate routing information for different topologies, and its behavior per specific topology within an address family can be configured in the per-AF-topology section. On routers without MTR support, only the topology base command will be available; on routers supporting MTR, the topology command will allow referencing a particular separate topology table definition by its name.

```
(config-router-af-topology)# ?  
Address Family Topology configuration commands:
```

auto-summary	: Enable automatic network number summarization
default	: Set a command to its defaults
default-information	: Control distribution of default information
default-metric	: Set metric of redistributed routes
distance	: Define an administrative distance
distribute-list	: Filter entries in eigrp updates
eigrp	: EIGRP specific commands
exit-af-topology	: Exit from Address Family Topology configuration mode
maximum-paths	: Forward packets over multiple paths
metric	: Modify metrics and parameters for advertisement
offset-list	: Add or subtract offset from EIGRP metrics
redistribute	: Redistribute IPv4 routes from another routing protocol
snmp	: Modify snmp parameters
summary-metric	: Specify summary to apply metric/filtering
timers	: Adjust topology specific timers
traffic-share	: How to compute traffic share over alternate paths
variance	: Control load balancing variance

Task: Modify administrative distance

```
(config-router)# distance eigrp <internal-routes> <external-routes>
```

Task: Modify the administrative distance on a per-prefix basis

```
(config-router)# distance <1-255> <source-ip> <source-wild-card> [<acl>]
```



The AD for EIGRP internal routes can be changed on a per-prefix basis, but external EIGRP routes cannot

21.10. DUAL

Diffusing Computation

- A distributed computation in which a single starting node commences the computation by delegating subtasks of the computation to its neighbors that may, in turn, recursively delegate sub-subtasks further, including a signaling scheme allowing the starting node to detect that the computation has finished while avoiding false terminations.
- In DUAL, the task of coordinated updates of routing tables and resulting best path computation is performed as a diffusing computation.

Diffusing Update Algorithm (DUAL)

- A loop-free routing algorithm used with distance vectors or link states that provides a diffused computation of a routing table.

- works very well in the presence of multiple topology changes with low overhead.

21.10.1. Topology Table

- stores information about every known destination
- network prefix/length, FD, CD, RD and route state

Task: Display EIGRP Topology Table

```
# show ip eigrp topology [as-number | [[ip-address] mask]] [active | all-links | pending | summary | zero-successors]

IP-EIGRP Topology Table for process 77

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.16.90.0 255.255.255.0, 2 successors, FD is 0
    via 172.16.80.28 (46251776/46226176), Ethernet0
    via 172.16.81.28 (46251776/46226176), Ethernet1
    via 172.16.80.31 (46277376/46251776), Serial0
P 172.16.81.0 255.255.255.0, 1 successors, FD is 307200
    via Connected, Ethernet1
    via 172.16.81.28 (307200/281600), Ethernet1
    via 172.16.80.28 (307200/281600), Ethernet0
    via 172.16.80.31 (332800/307200), Serial0
```

P - Passive

No EIGRP computations are being performed for this destination.

A - Active

EIGRP computations are being performed for this destination.

U - Update

Indicates that an update packet was sent to this destination.

Q - Query

Indicates that a query packet was sent to this destination.

R - Reply

Indicates that a reply packet was sent to this destination.

r - Reply

status Flag that is set after the software has sent a query and is waiting for a reply.

RD

Reported Distance

CD

- Computed Distance = RD + link cost
- Total metric along a path from the current router to a destination network through a particular neighbor computed using that neighbor's Reported Distance (RD) and the cost of the link between the two routers.
- Exactly one CD is computed and maintained per the [Destination, Advertising Neighbor] pair.

FD

- Feasible Distance
- least-known total metric to a destination from the current router since the last transition from the Active to Passive state.
- not necessarily equal to the current best CD to a destination.
 - There is exactly one FD per each destination, regardless of the number of neighbors.
 - FD is an internal variable maintained for each network known to EIGRP whose value is never advertised to another router.
- lowest bandwidth on the path to this destination as reported by the upstream neighbor
- total delay
- path reliability
- path loading
- minimum path maximum transmission unit (MTU)
- feasible distance
- reported distance
- route source (external routes are marked)

21.10.2. Feasibility Condition

- Feasibility condition: $RD < FD$
 - sufficient but not necessary condition
 - not every loop-free path satisfies the FC
 - proven by Dr. J. J. Garcia-Luna-Aceves
 - also called the Source Node Condition
- Feasible Successor: Neighbor that satisfy the FC
- successor: Feasible Successor with the least CD

SDAG

- Successor—Directed Acyclic Graph
- For a particular destination, a graph defined by routing table contents of individual routers in the topology, such that nodes of this graph are the routers themselves and a directed edge from router X to router Y exists if and only if router Y is router X's successor.

- After the network has converged, in the absence of topological changes, SDAG is a tree.

21.10.3. Topology Changes

- A topology change occurs whenever the distance to a network changes or a new neighbor comes online that advertises the network.
 - The distance change can be detected either through receiving an Update, Query, Reply, SIA-Query, or SIA-Reply packet from a neighbor that carries updated metric information about the network, or because a local interface metric has changed.
 - Also, the event of a neighbor going down is processed by setting the CD/RD of all networks reachable through that neighbor to infinity.
- Whenever EIGRP detects a topology change,
 - it first records the change into the topology table and updates the RD and CD of the neighbor that advertised the change (in case of a received EIGRP message) or was influenced by it (in case of a link metric change).
 - From among all neighbors that advertise the network, EIGRP identifies the one that provides the least CD, taking into account the updated CDs. Note that the FC is not invoked at this step.
- Only after identifying the neighbor offering the least CD, EIGRP verifies whether this neighbor meets the FC and is therefore a Feasible Successor. If it is, EIGRP will promote it to the Successor and start using it right away. If, however, that neighbor does not meet the FC, EIGRP will put the route into the Active state and send out Queries, asking its neighbors to assist in locating the best route.

21.10.4. Local Computation

- After a topology changes, if the best path is through a Feasible Successor, do the following:
 1. the Feasible Successor Providing the Least CD Is Made the New Successor.
 2. If the CD Over the New Successor Is Less Than the Current FD, the FD Will Be Updated to the New CD; Otherwise It Stays at Its Current Value.
 3. the Routing Table Is Updated to Point Toward the New Successor.
 4. If the Current Distance to the Destination Has Changed As a Result Of Switching to a New Successor, an Update Packet Is Sent to All Neighbors, Advertising the Router's Updated Distance to the Destination.

21.10.5. Diffusing Computation

If after a topology changes , if the router finds out that the new shortest path is provided by a neighbor that is not a Feasible Successor, do the following:

1. The entry in the routing table, still pointing to the current unchanged Successor, is locked: It must not be removed nor its next hop changed until the diffusing computation is finished and the route has been moved to the Passive state again.
2. The FD is set to the current (possibly increased) CD through the current unchanged Successor. Also, if this router ever needs to advertise its distance to the network while in the Active state, it

will also use the value of the current CD through the Successor.

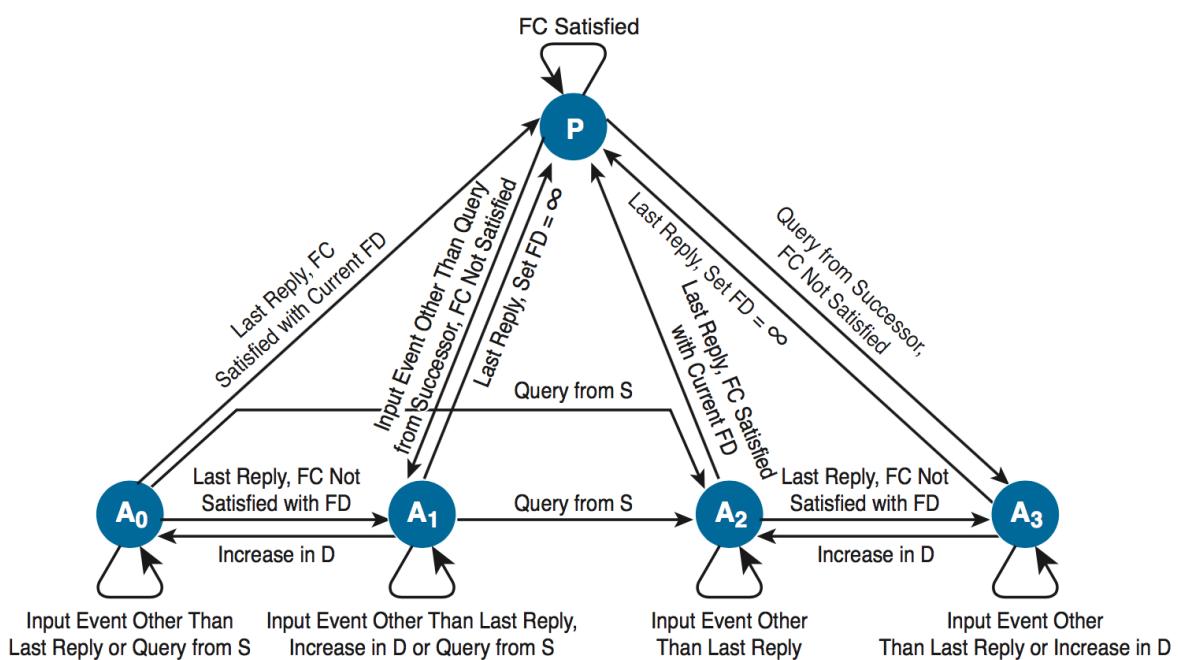
3. The network is put into the Active state and the router sends out a Query packet to all its neighbors. This Query packet contains the Active network's prefix and the router's current CD toward it.

One Single Topology Change

Each neighbor receiving a Query packet will process it by updating its own topology table using the distance information advertised in the Query and reevaluating its own choice of Successors and Feasible Successors. Two possibilities now exist: Either the neighbor still has its own Feasible Successor or a Successor that provides it with the least-cost loop-free path, or the information contained in the Query causes the neighbor to stop considering the path through its current Successor the shortest available and none of its own neighbors that offer the shortest path are a Feasible Successor.

21.10.6. Multiple Topology Changes

- Uses DUAL Finite State Machine to handle multiple topology changes occurring a simple diffusing computation



States

- P : Passive
- A0: Local Origin with Distance Increase
- A1: Local Origin
- A2: Multiple Origins
- A3: Successor Origin

Rules

- Unless a change in distance occurs such that the neighbor providing the least CD fails to meet the FC, the route remains passive.

- If a Query is received from the current Successor and, after processing the distance indicated in this Query, the neighbor that provides the least CD fails to meet the FC, the route will enter the A3 active state.
 - The router will send out Queries and wait for Replies.
 - If no further distance increase is detected while waiting for the Replies, the last Reply allows the router to transition back to the Passive state, reinitialize the FD, and choose any neighbor that provides the least CD as the new Successor.
- If a distance change caused by other means than a Query from a Successor is detected (this can be caused by receiving an Update, changing an interface metric, or losing a neighbor) and after processing the change, the neighbor that provides the least Computed Distance fails to meet the Feasibility Condition, the route will enter the A1 active state, also called the Local Origin Active State. The router will send out Queries and wait for Replies. If no further distance increase or Query from the current Successor is received while waiting for the Replies, the last Reply allows the router to transition back to the Passive state, reinitialize the Feasible Distance, and choose any neighbor that provides the least Computed Distance as the new Successor.
- If during the stay in the A3 (Successor Origin) or A1 (Local Origin) active states, another distance increase caused by other means than the Successor's Query is detected, another topology change during the diffusing computation has occurred. Because the router cannot advertise this updated distance while it is in the Active state, other routers might not be informed about it and their Replies might not take this new increased distance into account. Therefore, extra scrutiny is applied to the received Replies instead of simply choosing the neighbor that provides the least Computed Distance. This is accomplished first by changing the state from A3 (Successor Origin) to A2 (called Multiple Origins), or from A1 (Local Origin) to A0 (no official name; we will call it Local Origin with Distance Increase) states. In A2 or A0 states, the router waits to receive all remaining Replies. When the last Reply arrives, the router will first check whether the neighbor providing the least Computed Distance passes the Feasible Condition check using the Feasibility Distance value set when the route entered the Active state (recall that it was set to the increased distance through the current Successor at the moment of transitioning to the Active state). This extra check essentially mimics a situation in which the router is actually using the path through the current Successor and has just detected the distance increase, so it uses the current value of Feasibility Distance to verify whether the neighbor providing the least Computed Distance passes the Feasibility Condition. If it does, the route becomes Passive again, and the neighbor is chosen as the Successor. If it does not, however, the route will return from A0 (Local Origin with Distance Increase) to A1 (Local Origin) or from A2 (Multiple Origins) to A3 (Successor Origin) and the router will commence another diffusing computation by again sending a Query.
- If during the stay in A1 (Local Origin) or A0 (Local Origin with Distance Increase) active states a Query from the Successor is received, another topology change during the diffusing computation has occurred. Because the router cannot advertise this updated distance while it is in the Active state, other routers might not be informed about it and their Replies might not take this new increased distance into account. Therefore, extra scrutiny is applied to the received Replies. This is accomplished by changing the state to A2 (Multiple Origins) and then proceeding from that state just like in the previous case

Task: Display Details on EIGRP Active States

```
# sh ip eigrp topology active
```

21.10.7. Stuck-In-Active

- when all expected Replies are not received before the **Active** timer (default= 3 minutes) expires after first Query
 - The neighbors that did not reply will be removed from the neighbor table and their adjacencies torn down, and the diffusing computation will consider these neighbors to have responded with an infinite metric.
- If a neighbor does not respond to a Query message with its Reply within half of the Active timer time, the router will send the neighbor a SIA-Query message. The SIA- Query stands for a message saying “Are you still working on my Query?” If the neighbor is able to receive and process this SIA-Query, it will immediately respond with the SIA-Reply message. The contents of the SIA-Reply can either say “Yes, I still expect my own neighbors to send me the Replies I’ve asked them for” or “No, the computation is finished; this is my current metric to the destination.” In any case, the SIA-Reply is sent immediately as a response to the SIA-Query message; there is nothing to wait for. Receiving an SIA-Reply allows the Active timer to be reset, giving the diffusing computation an additional time to complete. At most three SIA-Queries can be sent, each after half of the Active timer. If the diffusing computation is not finished by the time the third SIA-Query was replied to by an SIA-Reply and the half of the Active timer expired again, the adjacency to the neighbor will be dropped. The same will happen if an SIA-Query is not responded to by an SIA-Reply within the next half of the Active timer. With the default setting of the Active timer to 180 seconds, three consecutive SIA-Query packets allow extending the diffusing computation to a maximum of $4 \times 90 = 360$ seconds (90 seconds to the first SIA-Query, plus each SIA-Query buying another 90 seconds).

Task: Control the Time That the Router Waits (After Sending a Query) Before Declaring the Route to Be In the Stuck In Active State.

```
(config-router)# timers active-time [<minutes>| disabled]
```



default wait time = 3 minutes

- Reasons a router doesn’t respond to EIGRP Query:
 - The neighbor router’s CPU is overloaded and the router either cannot respond in time or is even unable to process all incoming packets including the EIGRP packets.
 - Quality issues on the link are causing packets to be lost.
 - Low-bandwidth links are congested and packets are being delayed or dropped.
 - The network topology is excessively large or complex, either requiring the Query to propagate to a significant depth or causing an inordinate number of prefixes to be impacted by a single link or node failure.
- Troubleshooting SIA routes is generally a three-step process:

1. Find the Routes That Are Consistently Being Reported As SIA.
2. Find the Router That Is Consistently Failing to Answer Queries for These Routes
3. Find the Reason That Router Is Not Receiving or Answering Queries.

The first step should be fairly easy. If you are logging console messages, a quick perusal of the log indicates which routes are most frequently marked SIA.

The second step is more difficult. The command to gather this information is show ip eigrp topology active:

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status

```
A 10.2.4.0/24, 0 successors, FD is 512640000, Q
  1 replies, active 00:00:01, query-origin: Local origin
    via 10.1.2.2 (Infinity/Infinity), Serial1
  1 replies, active 00:00:01, query-origin: Local origin
    via 10.1.3.2 (Infinity/Infinity), r, Serial3
  Remaining replies:
    via 10.1.1.2, r, Serial0
```

Any neighbors that show an R have yet to reply (the active timer shows how long the route has been active). Note that these neighbors may not show up in the Remaining replies section; they may appear among the other RDBs. Pay particular attention to routes that have outstanding replies and have been active for some time, generally two to three minutes. Run this command several times and you begin to see which neighbors are not responding to queries (or which interfaces seem to have a lot of unanswered queries). Examine this neighbor to see if it is consistently waiting for replies from any of its neighbors. Repeat this process until you find the router that is consistently not answering queries. You can look for problems on the link to this neighbor, memory or CPU utilization, or other problems with this neighbor.

If you run into a situation where it seems that the query range is the problem, it is always best to reduce the query range rather than increasing the SIA timer.

21.11. Stub Routing

TODO Better explanation of this feature

- improves network scalability and stability.
- commonly used in hub-and-spoke networks.
- configured only on spoke routers.
- announces its stub router status using an additional TLV in its EIGRP Hello messages.

The results of configuring a router as a stub are multifold:

- A stub router does not propagate routes learned through EIGRP to its neighbors, with the exception of **leak-map** routes . This prevents a stub router from ever being considered a

Feasible Successor for remote networks by its neighbors and possibly becoming a transit router at some point in the future.

- A stub router advertises only a subset of its own EIGRP-enabled networks to its neighbors. This subset can be defined in the **eigrp stub** command using the **summary**, **connected**, **static**, **redistributed**, and **receive-only** keywords.
- Neighbors of a stub router aware of its stub status (thanks to the specific TLV in the stub router's Hello packets) will never send a Query packet to a stub router. This prevents the neighbors from converging through a stub router to reach networks that are remote to the stub router.

The following rules summarize the stub router behavior with respect to handling Query packets:

- Originating Query packets is not modified in any way. Rules for entering the Active state and sending Queries are precisely the same.
- Processing received Query packets depends on what network was queried for. If the network in the received Query is a network the stub router is allowed to advertise, meaning that it falls under the configured category of summary, connected, static, or redistributed, the router will process the Query normally (even possibly causing the stub router to become Active itself) and send back an appropriate Reply. The same is valid for an EIGRP-learned network that is allowed to be further advertised using a leak-map—a Query for such a network would be processed and responded to in the usual way. If the Query contains a network that the stub router knows about but is not allowed to advertise (the network does not fall under the configured category, or is learned through EIGRP but not allowed for further advertisement by a leak-map), it will be processed in the usual way as described earlier, but the Reply will always indicate infinite distance, regardless of what the stub router truly knows about the network. Receiving a Query for an unknown network will immediately cause the router to respond with a Reply and an infinite distance; however, this is regular EIGRP behavior not related to the stub feature.
- At this point, you might ask why a stub router would receive a Query, as its stub status should instruct its neighbors to avoid sending Queries to it. There are two primary reasons why even a stub router might receive a Query. First, a stub router's neighbor might be running an old IOS that does not recognize the stub TLV yet. Such a neighbor will create an adjacency to a stub router just fine, but it will also happily send Queries to it, not knowing that the router is a stub router. Second, if there are multiple routers on a common segment and all of them are configured as stub routers, if any of these stub routers need to send a Query, it will also send it to all its stub neighbors. This is done to support multihomed branch offices that usually have two branch routers configured as stubs. Each of these branch routers is connected to the headquarters through its own uplink, and they are also connected together by a common intra-site link. If the uplink on one of the branch routers fails, the affected router needs to converge through its neighbor branch router, and this might require a permission to send Queries to its fellow stub neighbor. Therefore, on a common segment with all routers configured as stubs, Queries are sent as usual.
- In case of multiaccess segments with mixed neighbors (stub and nonstub), EIGRP solves the problem of sending Queries only to nonstub neighbors in two ways: Either it sends the Queries as unicasts to the nonstub neighbors or it uses the Conditional Receive mode in RTP to send multicast Queries in such a way that only nonstub routers will process them. The choice of a particular mechanism depends on the number of nonstub neighbors. While mixing stub and

nonstub routers on a common segment is not a recommended practice, it is inevitable, for example, in cases where the hubs and spokes are interconnected by a DMVPN or a VPLS service.

Task: Configure EIGRP Stub

```
(config-router)# eigrp stub {[received-only] | [connected] [static] [ leak-map <name>] [redistributed] [summary]}
```

receive-only

does not advertise any prefixes.

- only receives prefixes advertised to it by its neighbors.
- either static routing on its neighbors or NAT/PAT on the stub router is required in this case to allow the networks behind the stub router to communicate with the outside world.
- cannot be used with any other keywords when configuring stub routing.

leak-map

Allows some prefix to be advertised

- crucial in scenarios where a branch office uses a pair of interconnected routers configured as stub routers. If these routers are to provide backup connectivity to each other, they must be allowed to readvertise EIGRP-learned routes to each other, even in stub mode.

connected



Advertises connected subnets.

- directly connected interfaces will not be advertised automatically; it is still necessary to add them to EIGRP using the usual **network** command
- option enabled by default

static

Advertises static routes.

- The static routes need to be redistributed into EIGRP to be advertised.

summary

Advertises Summary routes

- summary routes can be created manually (**summary-address**) or automatically at a major network border router (**auto-summary**).
- option enabled by default

redistributed

Advertises redistributed routes

i the stub router feature has no impact on what routes the hub router will advertise to its stub spokes. Without an additional configuration on the hub router, the spokes will be populated with full routing tables. Considering the fact that in a hub-and-spoke network, any other network beyond the branch networks is reachable through the hub, having full routing tables on spoke routers with most of their entries pointing toward the hub router is not particularly useful. Therefore, in these networks, the stub feature on spokes is usually combined with route filtering and summarization on the hub router. The hub router can be configured to advertise only the default route to the spoke router(s), filtering out all other more specific route entries, effectively reducing the routing table on the spoke to a single EIGRP-learned default route entry.

21.12. EIGRP Stub Routing Leak Map Support

21.13. Protocol-Dependent Modules

TODO

21.14. Goodbye Message and Graceful Shutdown

- broadcast when an EIGRP routing process is shut down
- Speeds convergence as peers don't have to wait the hold timer expiration
- Hello Message with all K-values set to 255
- Normal message displayed by routers that support Good Bye message

*Apr 26 13:48:42.523: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1 (Ethernet0/0) is down: Interface Goodbye received

- Misleading message displayed by router which doesn't support the Goodbye message

*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor (Ethernet0/0) is down: K-value mismatch

- The receipt of a goodbye message by a non supporting peer does not disrupt normal network operations.
- The nonsupporting peer will terminate the session when the hold timer expires
- The sending and receiving routers will converge normally after the sender reloads

21.15. Summarization

- All subnets are suppressed
- Creates boundary for Query propagation

- If a router receives a Query for a network it does not have in its topology table, it will immediately send back a Reply indicating an unreachable destination, without itself going active and propagating the Query further.

Task: Enable Auto-Summarization

```
(config-router)# auto-summarization
```



- Cannot be used in divergent networks
- create null0 summary

Task: Advertise a Single Summary In EIGRP Classic Mode

```
(config-if)# ip summary-address eigrp <asn> <prefix> <mask>
```

Task: Advertise a Single Summary In EIGRP Named Mode

```
(config-router-af-interface)# summary-address <prefix> <mask>
```

Task: Configure Summarization to Advertise a Default Route Into EIGRP

```
(config-if)# ip summary-address eigrp <asn> 0.0.0.0 0.0.0.0
```



- All subnets will be suppressed because all IPv4 networks are subnet of 0/0

Task: Configure a Fixed Metric for EIGRP Summary Address

```
(config-router)# summary-metric <network-address> <subnet-mask>
          { <bandwidth> <delay> <reliability> <load> <mtu> [
distance <ad> ] | distance <ad> }
```



When EIGRP creates a summary route, it includes a metric with the route in order to advertise it. EIGRP searches for components of the summary to be suppressed and represented by the summary. EIGRP finds the component with the best metric and copies the metric from the component into the summary. Components of the summary may change often, which means that every time the best component metric changes, the summary needs to be readvertised to all its peers. Even if the best component metric is not the one that changed, EIGRP still has to search every topology entry to make sure the summary is not affected. This can add a significant processing overhead.

21.15.1. Leak Map

Task: Advertise Specific Subnets Of a EIGRP Summary

```
(config-if)# ip summary-address eigrp <asn> <prefix> <mask> leak-map <route-maps>
```

21.15.2. Floating Summary Routes

TODO - By default, summarization install a route to Null0 to match the summary to prevent forwarding traffic for unreachable destinations. -

21.15.3. Poisoned Floating Summarization

TODO

21.16. EIGRP Route Authentication

- Supports MD5 in classic mode
- Supports MD5 and SHA-256 in multi-af mode

Task: Use MD5 Password In EIGRP Classic Mode

```
(config-if)# ip authentication mode eigrp <asn> md5  
(config-if)# ip authentication key-chain eigrp <asn> <password>
```

Task: Use MD5 Password In EIGRP Named Mode

```
(config-router-af-interface)# authentication mode md5  
(config-router-af-interface)# authentication key-chain <sesame>
```

Task: Authenticate EIGRP Neighbor with SHA-256 Password

```
(config-router-af-interface)# authenticate mode hmac-sha-256 <password>
```

- Can be applied at the **af-interface-default** in multi-af mode

21.17. Link Bandwidth Percentage

- by default, EIGRP packets consume max 50% of the link bandwidth as configured by the **bandwidth** command
- bandwidth configured by **bandwidth** in AS configuration and **bandwidth-percent** for named configuration

21.18. EIGRP Autonomous System Configuration

Task: Create a Basic EIGRP AS System Configuration

```
(config)# router eigrp asn  
(config-router)# network a.b.c.d [e.f.g.h]
```

- A maximum of 30 EIGRP can be configured
- EIGRP sends updates only interfaces in the specified networks

Task: Verify Eigrp Topology

```
show ip eigrp topology [all-links]  
show ip eigrp topology [prefix/len]
```

21.19. Router ID

- Used to avoid routing loops
- Advertised inside internal and external routes (in later IOS)
- same rule as OSPF

Task: Specify the EIGRP Router ID

```
(config-router)# eigrp router-id <a.b.c.d>
```



0.0.0.0 and 255.255.255.255 are not allowed

21.20. Unequal Load Balancing

If CD is the Computed Distance, then the eligible Feasible successor must satisfy the inequality below:

```
CD via Successor < CD via Feasible Successor < variance * CD via Successor
```

The amount of traffic flowing over a particular path can be computed as this ratio:

```
Highest Installed Path Metric / Path Metric
```

- The unequal-cost paths installed into the routing table also count toward the maximum number of parallel paths to a destination configured using the maximum-paths command. Depending on your network topology and requirements, it might be necessary to modify this setting.

Task: Enable EIGRP Unequal Load Balancing

```
(config-router)# variance <number>
```

Task: Enable EIGRP Unequal Load Balancing In Named Configuration

```
(config-router-topology)# variance <number>
```

21.21. Add-Path Support

- Allow a Hub (dual-homed in DMVPN) to advertise multiple-equal cost routes to the same destination
 - must have the multiple equal-cost installed in its routing table
 - must disable Split Horizon on the tunnel towards the spokes
 - must have variance = 1, no unequal load balancing on the hub and the spokes
 - must deactivate **next-self-hop [no-ecmp-mode]**
 - must be configured in the af-interface section of the named mode configuration
 - In certain scenarios, such as DMVPN deployments in which multiple branch offices are dual homed, hub routers usually have information about both routes to a particular dual-homed branch office, and can perform equal-cost load balancing on their end. However, without an additional mechanism, a hub is unable to advertise these equal-cost routes to other spoke routers. As a result, the other spokes only see a single route to the dual-homed branch office without an ability to perform load balancing over multiple paths, and if the single route they know about fails, they need to go over the usual reconvergence process in EIGRP to learn about the other route.
 - Spoke routers do not need to be specifically configured for the Add-Path feature, apart from possible tuning of the maximum-paths command to be allowed to insert multiple equal-cost paths into their routing tables.

21.22. Passive Interface

- Suppresses EIGRP hello packets and routing updates on interfaces
 - Doesn't form adjacencies
 - Includes the interface addresses in the topology database

Task: Configure EIGRP Passive Interfaces

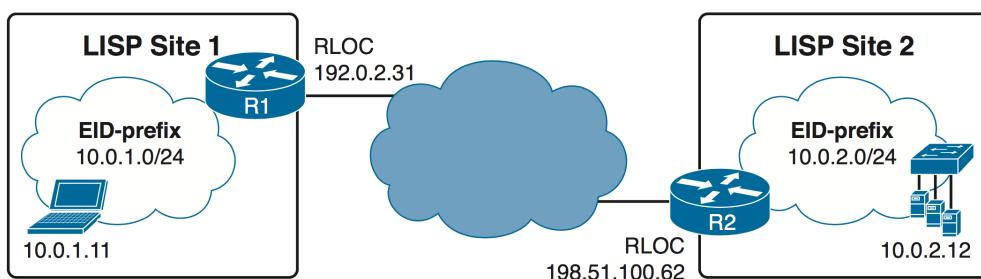
```
(config-router)# passive-interface [default] [<interface-type> <interface-number>]
```

21.23. EIGRP Over the Top

- Enables a single end-to-end routing domain between two or more EIGRP sites that are connected using a private or a public WAN connection.
- Relies on LISP
- Benefits:

- no dependency on the type of WAN connection used.
- no dependency on the service provider to transfer routes.
- no security threat because the underlying WAN has no knowledge of enterprise routes.
- simplifies dual carrier deployments and designs by eliminating the need to configure and manage EIGRP-BGP route distribution and route filtering between customer sites.
- allows easy transition between different service providers.
- supports both IPv4 and IPv6 environments.

21.23.1. LISP



- Locator/Identifier Separation Protocol
- Separate the identity and location into two independent entities, each of them represented by a complete address, and provide a mapping service so that the address representing the identity of a host can be resolved into the address that represents its location.
- Uses EID (EndPoint Identifiers) and RLOC (Routing Locator)
- LISP hence has both a control and a data plane.
 - The control plane in LISP comprises the registration protocol and procedures by which the tunnel routers R1 and R2 register the EIDs they are responsible for along with their RLOCs in a LISP-mapping service, and using these registrations they map EIDs into RLOCs.
 - The data plane defines the actual tunnel encapsulation used between Routers R1 and R2 when two hosts from each LISP sites communicate.
- In OTP, EIGRP serves as the replacement for LISP control plane protocols. Instead of doing dynamic EID-to-RLOC mappings in native LISP-mapping services, EIGRP routers running OTP over a service provider cloud create targeted sessions, use the IP addresses provided by the service provider as RLOCs, and exchange routes as EIDs.
- OTP is based on creating targeted EIGRP sessions between customer edge routers, and using the routing information carried by EIGRP to populate both routing tables and LISP mapping tables. The edge routers do not exchange any routing information with the service provider routers. Thus, this solution is fully controlled by a customer and requires no cooperation with the service provider, apart from providing full IP connectivity between customer routers

21.23.2. OTP CE

Task: Configure EIGRP OTP on CE

```
(config)# router eigrp test
(config-router)# address-family ipv4 unicast autonomous-system 100
(config-router-af)# neighbor 10.0.0.2 gigabitethernet 0/0/1 remote 3 lisp-encap 1
(config-router-af)# network 192.168.0.0
(config-router-af)# network 192.168.1.0
```

21.23.3. OTP Route Reflectors

Task: Configure EIGRP Route Reflectors

```
(config)# router eigrp test
(config-router)# address-family ipv4 unicast autonomous-system 100
(config-router-af)# af-interface gigabitethernet 0/0/1
(config-router-af-interface)# no next-hop-self
(config-router-af-interface)# no split-horizon
(config-router-af-interface)# exit
(config-router-af)# remote-neighbors source gigabitethernet 0/0/1 unicast-listen lisp-encap 1
(config-router-af)# network 192.168.0.0
```

More [WAN virtualization with OTP](#)

21.23.4. EIGRP Logging and Reporting

Task: Display the Contents Of the EIGRP Log

```
# sh ei address-family {ipv4 | ipv6} events
```

Task: Configure EIGRP Logging

```
Router(config-router)# eigrp ?
event-log-size : Set max log size (default=500)
event-logging : Log IP-EIGRP routing events (default)
log-neighbor-changes : enable IP-EIGRP neighbor logging (default)
log-neighbor-warnings : Enable/Disable IP-EIGRP neighbor warnings (default=every
10seconds)
```

21.23.5. SoO

TODO

Chapter 22. OSPF

Configuration Guides > IP Routing > [OSPF](#)

- link-state interior gateway protocol
- <https://tools.ietf.org/html/rfc2328>[RFC 2328]
- Dijkstra short path first algorithm
- classless protocol
- Transport via IP protocol 89
 - multicasts to 224.0.0.5 for AllSPF routers and to 224.0.0.6 for Designated Routers
 - unicasts
- equal-cost multipath
- hierarchical design to reduce traffic
- authentication updates

22.1. Neighbors

To form adjacency neighbors must agree on

- unique router ID
- unique interface IP address
 - primary IP address for OSPFv2
 - link-local address for OSPFv3
- common attributes
 - interface area-id
 - authentication
 - hello and dead timers
 - area type (normal, stub, NSSA,)
 - interface MTU
 - other optional capabilities

22.2. Router Id

Determined by these rules in order of preference at boot or ospf process restart:

- manually configured router id
- highest IP address of an up/up loopback not used by other OSPF process
- highest IP address of an up/up non-loopback interfaces not used by other OSPF process

Task: Set the Router-Id

```
(config-router)# router-id <a.b.c.d>
```

22.3. DR Election

- There is no pre-emption in OSPF
 - Router must wait for the failure of the current DR
 - Use the WAIT timer = DEAD timer
- on hub-and-spoke, best practice is to have hub as DR and spokes not eligible as DR with priority=0

Task: Priority

```
(config-if)# ip ospf priority <0-255>
```

Task: Set the WAIT Timer

```
(config-if)# ip ospf dead-timer <seconds>
```

22.4. Ospf Cost

"Cost" = $10^8 / \text{Bandwidth(bps)}$ "

Task: Description

```
(config-router)# auto-cost reference-bandwidth <bps>
```

22.5. OSPF Packet Format

22.5.1. Common OSPF Packet Header

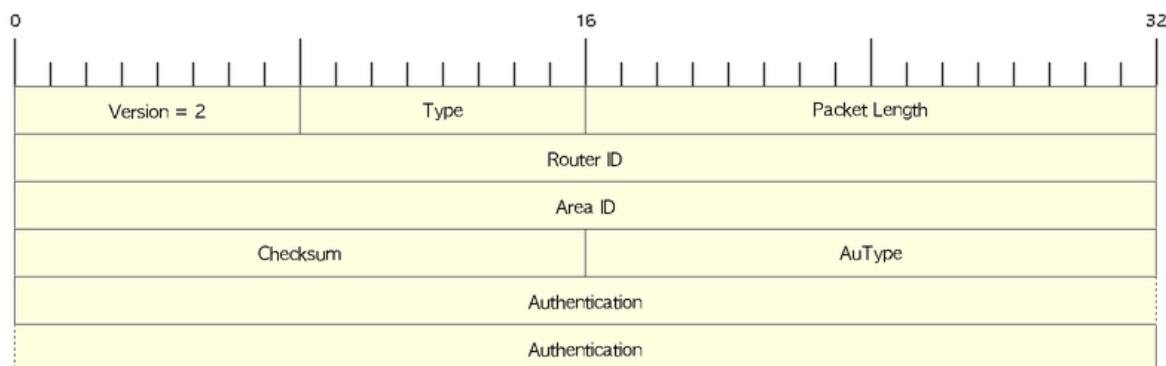


Figure 36. OSPF Header Format

Type

Hello (1), database description (2), Link-State Request (3), Link-State Update (4), or Link-State Acknowledgment (5).

Packet length

Length of the protocol packet in bytes including the OSPF header.

Router ID

The ID of the router originating the packet.

Area ID

The area that the packet is being sent into.

Checksum

standard IP checksum of the entire contents of the packet, excluding the 64-bit authentication field.

AuType

Identifies the authentication scheme to be used for the packet.

- 0: no authentication
- 1: plain-text authentication
- 2: cryptographic authentication

Authentication

64-bit field for use by the authentication scheme.

22.5.2. Hello Packet

- Sent from the primary IP address (not the secondary addresses)
- Every 10 seconds (Ethernet), 30 seconds (Non-broadcast)



OSPF neighbors will become fully adjacent if one or both of the neighbors are using unnumbered interfaces for the connection between them.

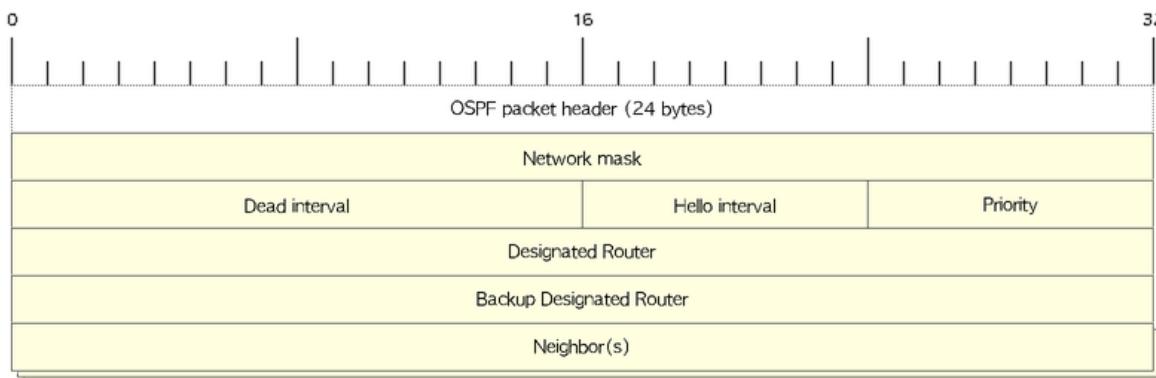


Figure 37. OSPF Hello Packet Format

Task: Configure OSPF Hello Interval

```
(config-if)# ip ospf hello-interval <seconds>
```

Task: Set the Interval During Which at Least One OSPF Hello Packet Must Be Received from a Neighbor Before the Router Declares That Neighbor Down

```
(config-if)# ip ospf dead-interval {<seconds> | minimal hello-multiplier <number>}
```

Task: Configure the OSPF demand circuit to reduce periodic OSPF hello transmission and paranoid update flooding

```
(config-router)# ip ospf demand-circuit
```



Periodic hellos are only suppressed for point-to-point and point-to-multipoint OSPF network types. This feature is negotiated as part of the neighbor adjacency establishment, thus only one OSPF router on the segment requires that the feature be enabled. If routers on the segment do not support it, it will just ignore the option in the HELLO packet, but OSPF neighbors will still be established.

22.5.3. Database Description Packet

- Uses an OSPF-defined simple error-recovery process.
 - Each DD packet, which can contain several LSA headers, has a sequence number assigned.
 - The receiver acknowledges a received DD packet by sending a DD packet with the identical sequence number back to the sender.
 - The sender uses a window size of one packet and then waits for the acknowledgment before sending the next DD packet.
- Only the master is allowed to send DD packets on its own accord as well as to set and increase their sequence numbers.
- A slave is allowed to send a DD packet only as a response to a DD packet received from master router, and must use the same sequence number. In effect, a slave is polled by the master and only responds to it.
 - If a slave has more DD than the master, he uses the M flag

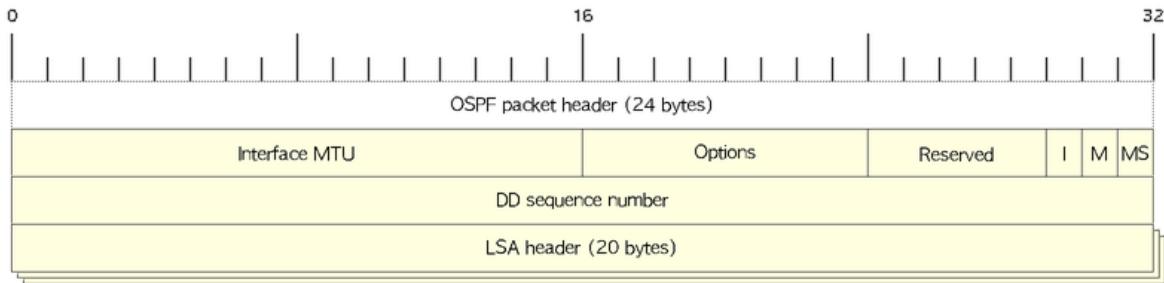


Figure 38. OSPF DD Packet Format

Interface MTU

Size of the largest IP message that can be sent on this router's interface without fragmentation

I-bit

Initial for the first in a sequence of DD messages

M-bit

More DD follow this one

MS-bit

if this message is sent by the master in the communication

Options

For optional OSPF capabilities

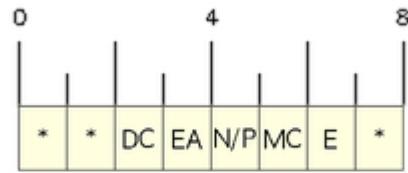


Figure 39. Options field

DC-bit

describes the router's handling of demand circuits



EA-bit

describes the router's willingness to receive and forward External-Attributes-LSAs

N/P-bit

describes the handling of Type-7 LSAs

MC-bit

describes whether IP multicast datagrams are forwarded

E-bit

describes the way AS-external-LSAs are flooded

22.5.4. Link State Request

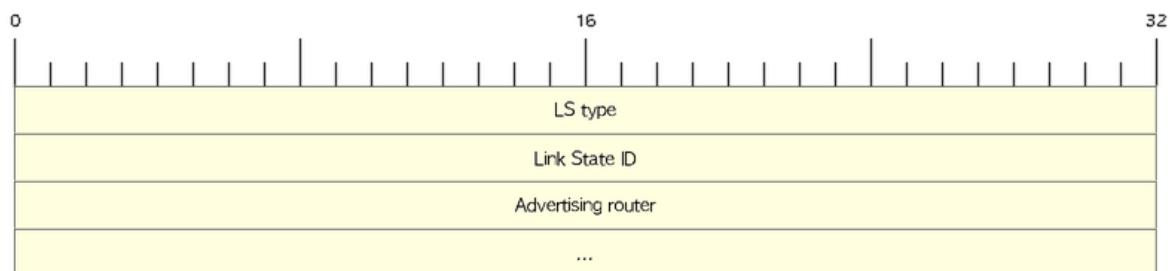


Figure 40. OSPF Link State Request Format

22.5.5. Link State Update

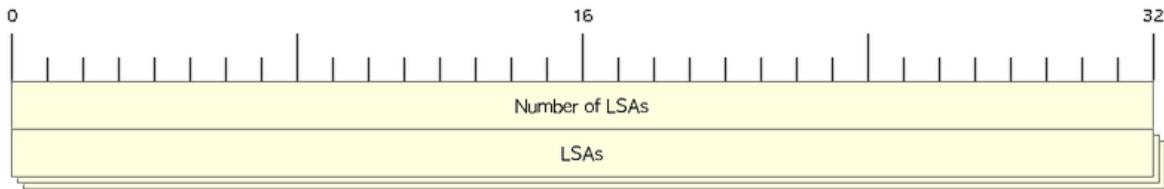


Figure 41. OSPF Link State Update Format

22.5.6. Link State Acknowledgment

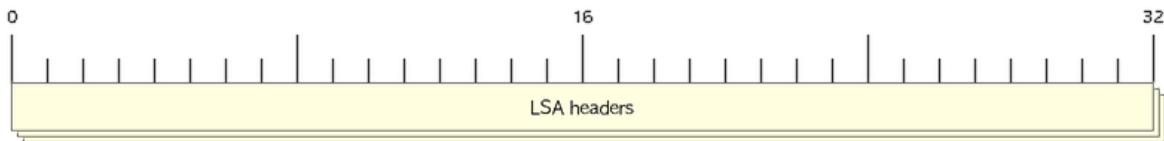


Figure 42. OSPF Link State Acknowledgment Format

LSA headers

Contains LSA headers to identify the LSAs acknowledged.

22.5.7. Link-State Packets

- only a router that has originated a particular LSA is allowed to modify it or withdraw it.
 - Other routers must process and flood this LSA within its defined flooding scope if they recognize the LSA's type and contents, but they must not ever change its contents, block it, or drop it before its maximum lifetime has expired.
- has a unique LSID (Link State Identifier)

Type 1 Router LSA

- one per router per area
- lists the RouterID, the IP Addresses and neighbors for each interface in that area
- represents Stub networks (subnet on which a router has not formed any neighbor relationships)
- flooded only within the same area
- LSID = Router ID

Type 2 Network LSA

- one per transit network
 - network over which two or more OSPF routers have become neighbors and elected a DR so that traffic can transit between them
 - except for point-to-point connection treated as a combination of p2p link and a stub IP network (to facilitate unnumbered p2p links)
- generated by DR

- describes the set of routers attached to a particular network
- describes the subnet and the router interfaces connected to the subnet
- flooded only within the area that contains the network
- LSID = DR's interface IP Address on that subnet

Type 3 Summary inter-area LSA

- Generated by ABR
- describes inter-area routes to network
 - represents networks present in one area when being advertised into another area.
 - Defines the subnets in the origin area, and cost, but no topology data.
- Flooded only within its area of origin; reoriginate on ABRs.

Type 4 Summary inter-area LSA

- Generated by ABR
- Flooded by ABR to all areas except the area containing the ASBR
- describes routes to ASBR
 - tells other routers in the area how to get to the advertising router of an external route
- Flooded all areas except the area containing the ASBR

Type 5 AS external LSA

- originated by ASBR
- describes routes to destinations external to the AS
- flooded all over except stub areas

Type 6 Group Membership LSA

- defined for MOSPF
- Not supported by Cisco

Type 7 NSSA External LSA

- Created by ASBRs inside an NSSA, instead of a type 5 LSA.
- Flooded only within its area of origin;
- converted to type 5 LSA on an ABR toward other areas.

Type 8 External Attributes LSA

- Created by ASBRs during BGP-to-OSPF redistribution to preserve BGP attributes of redistributed networks.
- Not implemented in Cisco routers

Type 9-11 Opaque LSA

- Used as generic LSAs to allow for easy future extension of OSPF;
 - for example, type 10 has been adapted for MPLS traffic engineering.

- have different flooding scope:
 - Type 9 has link-local flooding scope,
 - type 10 has area-local flooding scope,
 - type 11 has autonomous system flooding scope equivalent to the flooding scope of type 5 LSAs (not flooded into stubby areas and NSSAs).
1. OSPF'S SPF Algorithm Links Different Pieces Of Information Together.

For a router in Area 1 to reach the external route in Area 3, it has to look at the Type-5 that represents the external route. Then it has to look at the Type-4 representing the ABR on the area that the ASBR lives in. Then we have to look at the Type-3 to get to that remote ABR. Finally we look at the Type-1 and Type-2 LSAs in our area to determine how to get to our closest ABR.

Read more [here](#).

Task: Display the OSPF Database

```
# sh ip ospf database
```

22.6. Backbone

ABR

Router actively connected to multiple areas **including** Area 0

- has one LSDB for each area
- runs the SPF for each LSDB then combines the result in a single routing table
- can summarize and filter routes
- ignores type 3 LSAs learned in a nonbackbone area during SPF calculation, which prevents an ABR from choosing a route that goes into a nonbackbone area and then back into the backbone.

22.7. Stubby Areas

All stubby area types - block Type 4/5 LSA - automatically inject default routes except NSSA

22.7.1. Stubby Area

- Doesn't have an ASBR

Task: Configure a Stubby Area

```
(config-router)# area <id> stub
```

22.7.2. Totally Stubby

- Stubby areas where Type 3 are blocked

Task: Configure Totally Stubby Areas on the ABR

```
(config-router)# area <id> stub no-summary
```

22.7.3. NSSA

- Contains one or more ASBRs
- Allows creation of Type 7
- Doesn't automatically inject default routes
- The ABR with highest RID translates Type 7 to Type 5

Task: Configure NSSA

```
(config-router)# area <id> nssa
```

Task: TODO

```
(config-router)# area <id> nssa translate type7 suppress-fa
```



See INE lab for good explanation

Task: Inject Default Routes In NSSA

```
(config)# area <id> nssa default-information-originate
```

22.7.4. Totally NSSA

- NSSA where Type 3 are blocked
- but a Type 3 0/0 is created instead

Task: Configure Totally NSSA

```
(config-router)# area <id> nssa no-summary
```

22.8. OSPF Path Selection

- Intra-Area > Inter-Area > External Routes (E1/N1 > E2/N2)

22.9. Virtual Links

- purposes:
 - Areas not physically connected to area 0
 - partitioning the backbone
- transit area can not be stub

Router A

```
(config)# router ospf 10
(config-router)# area 2 virtual-link 2.2.2.2
```

Router B

```
(config)# router ospf 10
(config-router)# area 2 virtual-link 1.1.1.1
```

Task: TODO

```
(config-router)# no capability transit
```

Task: Configure Authentication on Virtual Links

```
! Null
(config-router)# area <id> virtual-link <router-id> authentication { null }

! Plaintext
(config-router)# area <id> virtual-link <router-id> authentication { authentication-
key <key-value> }

! MD5
(config-router)# area <id> virtual-link <router-id> authentication { message-digest
message-digest- key key-num md5 key-value}

! Cryptographic
(config-router)# area <id> virtual-link <router-id> key-chain <key-chain-name>
```

[What are ospf areas and virtual links](#)

22.10. Network Types

broadcast

- multicast hellos every 10 seconds
- automatic neighbor discovery
- DR/BDR election

- default for LAN ethernet, TR, FDDI
- DR doesn't change the next hop of advertised prefixes

Point-to-point

- only 2 routers
- automatic neighbor relationships
- no DR/BDR election
- multicast hellos every 10 seconds
- default for HDLC and PPP

Non-broadcast

- unicast hellos every 30 seconds
- manual configuration of neighbor
- DR/BDR election
- default on Frame Relay, X.25 and SMDS

Point-to-multipoint

- multi-access, broadcast
- hellos every 30 seconds
- automatic discovery of neighbor (MA)
- DR/BDR election
- one IP subnet
- maintain connectivity during a VC failure ???
- generates host routes (with mask /32) for each neighbor
- default for ???

Point-to-multipoint non-broadcast

- manual configuration of neighbor
- no DR/BDR election
- network proprietary to Cisco
- hellos every 30 seconds

Loopback

- if Multi-Access network type then DR/BDR election
- if non-broadcast then manual configuration of neighbors

[OSPF design guide: selecting interface network types](#)

Task: Configure OSPF Network Type

```
(config-if)# ospf network {broadcast| point-to-point| point-to-multipoint [non-broadcast] | non-broadcast | loopback }
```

22.11. Graceful Restart

- enables a router to continue to forward packets during a restart of the routing process
- must be configured on all neighbor routers
- can also work with EIGRP, BGP, IS-IS
- default since IOS 12.4(6)T
- 2 versions: RFC 3623 and Cisco NSF

[Cisco NSF](#)

22.12. SPF Throttling

22.13. Capability Vrf-Lite

Read OSG, chapter 19, VRF lite, pp. 872-876

http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_ospf/command/ospf-a1.html#wp2582896905

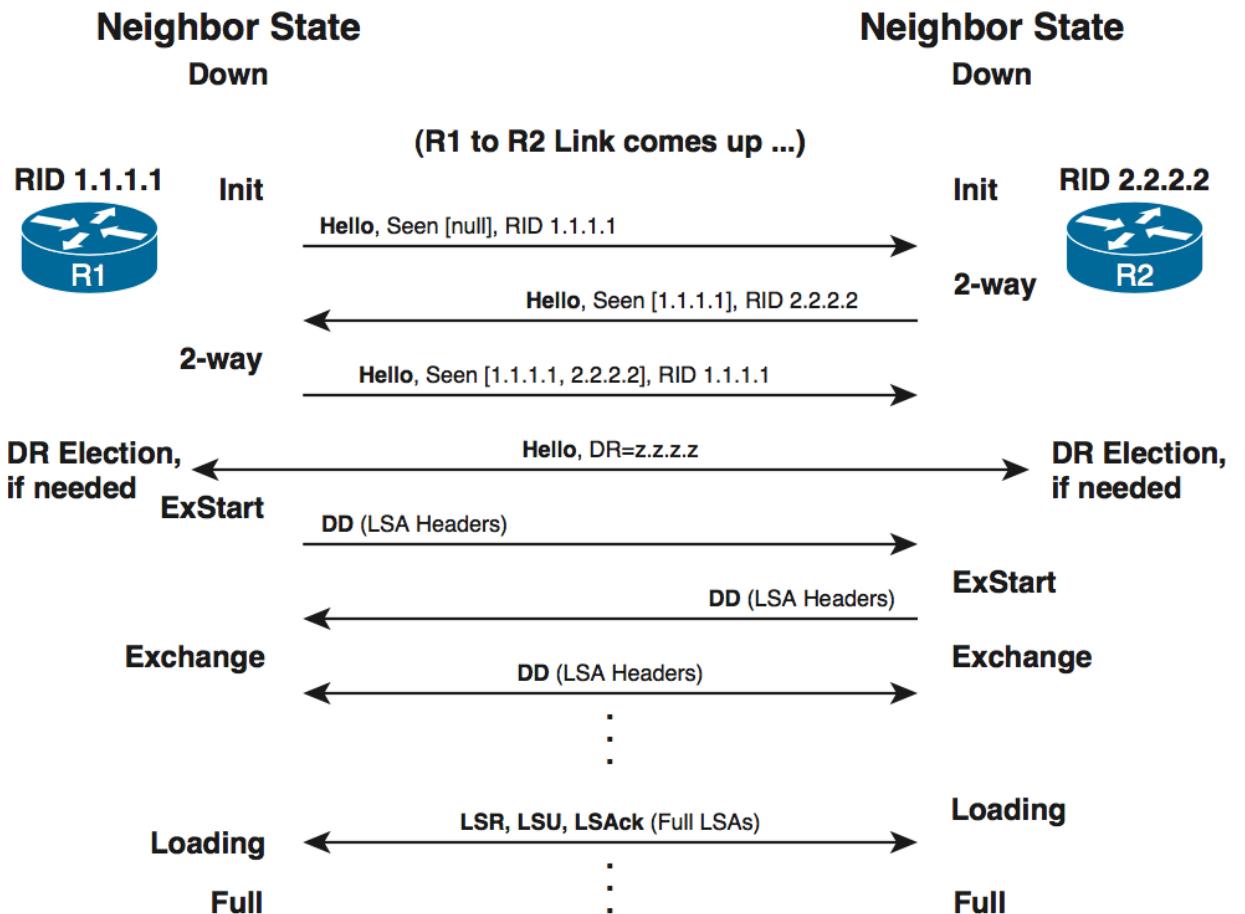
22.14. Summarization

Why the null 0 interface is added ?

- do prevent routing loops
 - packets destined for the routes that have been summarized will a longer match
 - packets destined to summary routes will be dropped

See good explanation

22.15. OSPF States



Down

- No hellos have been received from neighbors

Attempt

- Unicast hello packet has been sent to neighbor, but not yet received back
- only used for manually configured NBMA neighbors

Init

- I have received a hello packet from a neighbor, but they have not acknowledged a hello from me

2-way

- I have received a hello packet from a neighbor and he acknowledged a hello from me
- I can see my Router Id in the neighbor's hello packet
- Stop here for DROthers

Exstart

- Master & slave relationship is formed where master has higher Router-id
- Master chooses the starting sequence number of the DBD packets that are used for actual LSA exchange.

Exchange

- Local link state database is sent through DBD packets

- DBD sequence number is used for reliable acknowledgement/retransmission

Loading

- LSR packets are sent to ask for more info about a particular LSA

Full

- Neighbors are fully adjacent and databases are synchronized.

Key Point

In the beginning of the exchange, each router places the other into the ExStart state. Each of them considers itself to be the master, and sends an empty DD packet to the other router, containing a randomly chosen sequence number, and MS (Master), M (More), and I (Init) flags set to 1. After receiving the neighbor's DD packet, however, the router with the lower RID will change its role to slave, and it will respond with a DD packet with MS and I flags cleared and the sequence number set to the sequence number of master's DD packet. This accomplishes the master/slave selection, and both routers move to the Exchange state. The master will then send a DD packet with the sequence number incremented by 1, optionally containing one or more LSA headers, and the slave will respond with a DD packet reusing the same sequence number from the received packet, optionally advertising its own LSA headers. The exchange continues in the same fashion, with the master incrementing the sequence number of each subsequent DD packet, until both routers have advertised all known all LSA headers (the master will stop sending DD packets when it has advertised all LSA headers itself and the last DD response from the slave has the M flag cleared).

[ospf design guide: link-state advertisements](#)

22.16. OSPF Process

Task: Enable OSPF Process (Legacy Command)

```
(config)# router ospf <process-id>
(config-router)# network <a.b.c.d> [<w.i.l.d>] area <id>
```

- inject both the primary and secondary addresses
- If an interface is IP unnumbered, and there is a **network** statement that matches the IP address of the primary interface, inject both the primary interface and the unnumbered interface

Task: Enable OSPF Process (Interface Level)

```
(config-if)# ip ospf <process-id> area <id>
```



- inject any and all secondary subnets

Task: Prevent OSPF to Advertise Secondary Prefixes

```
(config-if)# ip ospf <process-id> area <id> secondaries none
```

22.17. OSPF Authentication

22.17.1. Classic OSPF Authentication

- Null , default: type 0
- Plain-text, simple password authentication

Task: Disable authentication

```
(config-if)# ip ospf authentication null
```

Task: Configure clear-text OSPF authentication

```
(config-router)# area <id> authentication  
(config-if)# ip ospf authentication  
(config-if)# ip ospf authentication-key <string>
```

- Message digest authentication

```
(config-router)# area <id> authentication message-digest  
(config-if)# ip ospf message-digest-key <key-id> md5 <string>
```

Key Rollover Procedure with Multiple MD5 Keys

Multiple MD5 keys with different key IDs are allowed per interface. This allows for graceful key migration where a new key can be added without disrupting the adjacencies.



- To sign sent packets, it always uses the key that was added as the last one to the interface (regardless of the key number).
- To authenticate the received packet, it uses the key ID that is indicated in the packet.
- If a neighbor is detected on an interface that uses a different key number than this router, OSPF enters a key migration phase in which it sends all packets as many times as how many keys are configured on the interface, and each packet is signed with a different key.
- The migration phase ends when all neighbors have migrated to the same key as the one used to sign sent packets by this router.
- This procedure is also called the OSPF key rollover procedure.
- Because plaintext passwords do not have key numbers, the key rollover is not available for plaintext authentication.

22.17.2. Extended Cryptographic OSPF Authentication

- Uses SHA-HMAC (Secure Hash Algorithm - Hash Message Authentication Code) as per RFC 5709
- Uses key chains
 - Each key in the key chain must have a cryptographic algorithm configured using a per-key **cryptographic-algorithm** command. Failure to do so will result in OSPF not using that key.
 - Each key in a key chain can be configured with the **send-life-time** and accept-life-time keywords to limit its usability to a particular timeframe. If multiple keys in the key chain are eligible to sign egress packets, the key with the highest key ID will be used. Be aware that this behavior differs from RIPv2 and EIGRP that select the key with the lowest key ID.
 - The key rollover procedure as used by classic OSPF is not used with key chains. There is no key migration phase of sending multiple OSPF packets signed with different valid keys.
 - To sign egress packets, use the valid key with the highest key ID in the key chain.
 - To authenticate ingress packets, try to use the key indicated in the received packet.

Task: Configure a Cryptographic Algorithm for the Key Chain

```
(config)# key-chain <name>
(config-keychain)# key <number>
(config-keychain-key)# cryptographic-algorithm ?

hmac-sha-1    HMAC-SHA-1 authentication algorithm
hmac-sha-256   HMAC-SHA-256 authentication algorithm
hmac-sha-384   HMAC-SHA-384 authentication algorithm
hmac-sha-512   HMAC-SHA-512 authentication algorithm
md5           MD5 authentication algorithm
```

Task: Configure the Extended Cryptographic OSPF Authentication

```
(config-if)# ip ospf authentication key-chain <key-chain-name>
```



Configuring the extended cryptographic authentication using the area OSPF process level command is not supported.

22.18. TTL Security Check

- Drops packets with TTL < 255 except on virtual links and sham links
 - If all OSPF routers sent their packets with TTL set to 255, receiving an OSPF packet with its TTL less than 255 would be a clear indication that the packet originated outside the network segment over which it was received. Because OSPF communication is, with the notable exception of virtual links and sham links, always based on direct router-to-router communication, receiving an OSPF packet outside a virtual link or a sham link with its TTL less than 255 is a possible indication of a malicious activity.

Task: Configure the Time-to-Live (TTL) Security Check Feature on a Specific Interface

```
(config-if)# ip ospf ttl-security [hops <count> | disable]
```

Task: Configure the Time-to-Live (TTL) Security Check Feature on All Interfaces

```
(config-router)# ip ospf ttl-security all-interfaces
```

Task: Configure TTL Security on a Virtual Link

```
(config-router)# area virtual-link ttl-security <hops>
```

Task: Configure TTL Security on a Sham Link

```
(config-router)# area virtual-link ttl-security <hops>
```

22.19. SPF

22.19.1. Spf Timers

- spf-delay: between topology change notifications and recalculation of the shortest path
- spf-holdtime : between spf calculations

Task: Configure Spf Timers

```
(config-router)# timers spf seconds <seconds>
```

22.19.2. SPF Throttling

- Defines a variable-length wait interval between two consecutive SPF runs
- Controls by 3 parameters:
 - spf-start: initial wait interval before an SPF computation, if the network has been stable for a prolonged period of time.
 - spf-hold: wait time between subsequent SPF runs, and its value doubles for each consecutive SPF run.
 - spf-max-wait: maximum time between two SPF runs (that is, doubling the spf-hold value is capped at spf-max-wait), and also defines a period during which the network must be stable for the wait interval to be set back to spf-start and the spf-hold to its preconfigured value. If the network has been stable for the last spf-hold period but not for the entire spf-max-wait since the last SPF run, the wait interval returns to the spf-start value but the subsequent wait will still be set to twice the previous spfhold value.

Task: Configure Spf Throttling

```
(config-router)# timers throttle spf <spf-start> <spf-hold> <spf-max-wait>
```

Task: Verify SPF Throttling Configuration

```
# sh ip ospf | i SPF  
  
Initial SPF schedule delay 10000 msec  
Minimum hold time between two consecutive SPFs 15000 msec  
Maximum wait time between two consecutive SPFs 100000 msec
```

22.19.3. LSA Throttling

Task: Configure LSA Throttling

```
(config-router)# timers throttle lsa all <start-interval> <hold-interval> <max-interval>
```

Task: Verify LSA Throttling Configuration

```
# sh ip ospf | i LSA

Initial LSA throttle delay 10000 msec
Minimum hold time for LSA throttle 15000 msec
Maximum wait time for LSA throttle 100000 msec
Minimum LSA arrival 1000 msec
LSA group pacing timer 240 secs
```

TODO Apart from throttling the LSA origination, a router can also be configured to ignore the same LSA upon arrival if it appears to arrive too often. This throttling of arriving LSAs is configured using the timers lsa arrival milliseconds OSPF command. If two or more same LSAs arrive less than milliseconds apart, only the first one is accepted and the remaining LSAs are dropped. In effect, the same LSA is accepted only if it arrives more than milliseconds after the previous accepted one. The default setting is 1000 milliseconds and can be seen in the show ip ospf output in Example 9-16. Obviously, the value of the minimum LSA arrival interval should be smaller than the neighbors' initial hold interval in LSA Throttling. Otherwise, a neighbor would be allowed to send an updated LSA sooner than this router would be willing to accept it.

22.19.4. Incremental SPF

Task: Configure Incremental SPF

```
(config-router)# ispf
```

Task: Verify Incremental SPF Configuration

```
# sh ip ospf | i Incremental

Incremental-SPF enabled
```

22.20. OSPF Filtering

22.20.1. Routes Filtering Not LSA Filtering

- uses **distribute-list**
- The distribute list in the inbound direction applies to results of SPF—the routes to be installed into the router's routing table.
- The distribute list in the outbound direction applies only to redistributed routes and only on an ASBR; it selects which redistributed routes shall be advertised.
- The inbound logic does not filter inbound LSAs; it instead filters the routes that SPF chooses to add to that one router's routing table.
- If the distribute list includes the incoming interface parameter, the incoming interface is checked as if it were the outgoing interface of the route.

22.20.2. ABR Type 3 LSA Filtering

- allows an ABR to filter type 3 LSAs at the point where the LSAs would normally be created.

Task: Filter Type 3 LSA on the ABR

```
(config-router)# area <id> filter-list prefix <prefix-list-name> { in | out }
```

22.20.3. Using the Area Range No-Advertise Option

Task: Summarize and Do Not Advertise Components

```
(config-router)# area <id> range <prefix /length> not-advertise [ cost cost ]
```

22.21. OSPFv2 Prefix Suppression

- RFC 6860 defines a method of hiding, or suppressing, the transit link prefixes in OSPF TODO Complete this

Task: Activate OSPFv2 Prefix Suppression for the Entire Router

```
(config-router)# prefix-suppression
```



suppress all prefixes on all its OSPF-enabled interfaces except loopbacks, secondary IP addresses, and prefixes on passive interfaces. Such prefixes are considered nontransit prefixes.

Task: Activate OSPFv2 Prefix Suppression on a Specific Interface

```
(config-if)# ip ospf prefix-suppression [disable]
```

22.22. OSPF Stub Router

- allows a router to either temporarily or permanently be prevented from becoming a transit router.
 - a transit router is simply one to which packets are forwarded, with the expectation that the transit router will forward the packet to yet another router.
 - a nontransit routers only forward packets to and from locally attached subnets.

TODO Better explanation

Task: Advertise the maximum metric value to all neighbors inside its Type-1 Router LSA

```
(config-router)# max-metric router-lsa
```

Task: Advertise the maximum metric until BGP converges

```
(config-router)# max-metric router-lsa on-startup wait-for-bgp
```

Task: Advertise the maximum metric for a specified amount of time

```
(config-router)# max-metric router-lsa on-startup announce-time <minutes>
```

22.23. OSPF Graceful Restart

22.24. OSPF Graceful Shutdown

Chapter 23. IS-IS

- link-state protocol
- doesn't run over any other network protocol
 - encapsulates messages directly in data-link frames
 - sends to the layer 2 address called SNAP (sub-network point of attachment)

23.1. NSAP Address

- variable length between 8 and 20 octets
- assigned to the entire node

IDI

Initial Domain ID

DSP

Domain Specific Part

AFI

Authority and Format ID

HO-DSP

high-order DSP

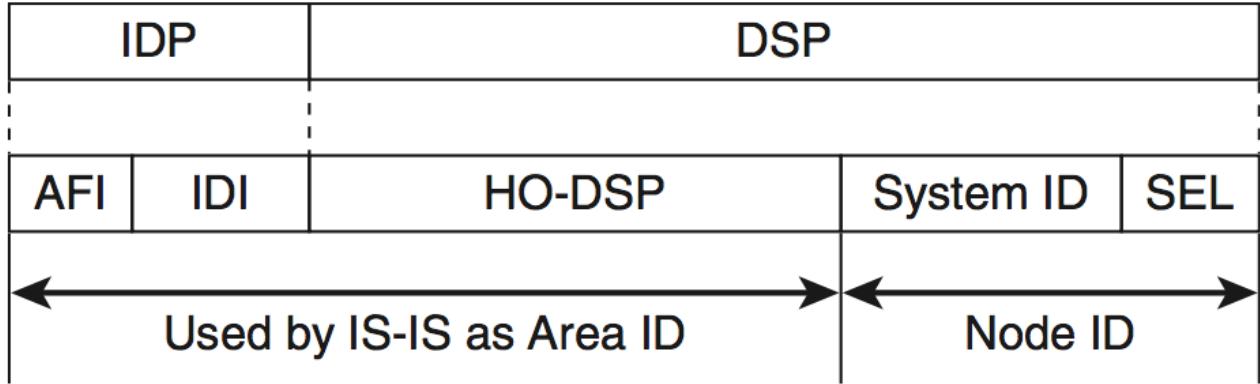
- identifies the area
- can be divided in sub-fields

System ID

- generally 6 octets
- between 1 and 8 octets -

SEL

- identifies particular services on the destination node
- if SEL = 0,
 - identifies node NET (Network Entity Title)
 - mandatory for is-is configuration



AFI Meaning	IDI Length and Contents	HO-DSP Length and Contents
39 Use of Data Country Code (ISO 3166)	2 octets; numeric country code according to ISO 3166	10 octets; area number
45 Use of international phone numbers (ITU-T E.164)	8 octets; international phone number according to E.164	4 octets; area number
47 Use of International Code Designator (ISO 6523)	2 octets; international organization code according to ISO 6523	10 octets; area number
49 Locally defined format (private addressing; free format)	Formally not present	Between 0 and 12 octets; area number

23.2. Levels Of Routing

- Level 0 : ES-ES / ES-IS on the same link
- Level 1 : IS-IS intra-area
- Level 2 : IS-IS inter-area
- Level 3 : IS-IS inter-domain (inter AS)

23.3. Adjacency

- Separate for L1 and L2
 - L1 with same area
 - L2
- possible adjacency states:
 - Down: The initial state. No IIHs have been received from the neighbor.
 - Initializing: IIHs have been received from the neighbor, but it is not certain that the neighbor is properly receiving this router's IIHs.
 - Up: IIHs have been received from the neighbor, and it is certain that the neighbor is properly receiving this router's IIHs.

23.4. Metrics

- assigned to individual interfaces
- default to 10
 - not automatically recalculated if bandwidth changes
 - manually set with **isis metric <number>**
- types of metrics
 - Default: Required to be supported by all IS-IS implementations; usually relates to the bandwidth of the link (higher value represents a slower link)
 - Delay: Relates to the transit delay on the link
 - Expense: Relates to the monetary cost of carrying data through the link
 - Error: Relates to the residual bit error rate of the link
- narrow (original) metrics
 - 6 bits for interface
 - 10 bits for complete path
- wide metrics
 - 24 bits for interface
 - 32 bits for complete path



Use same type of metrics in one area

23.5. Packets

- Hello
- Link State PDU
- Complete Sequence Numbers PDU
- Partial Sequence Numbers PDU

23.5.1. Hello

- also called IIH (IS-IS Hello)
- separate L1 and L2 hellos in bcast network
- single L1L2 hellos on point-to-point link
- sent every 10 seconds per default
- defined Hold time with **isis hello-multiplier**
- do not need to match (contrary to OSPF)
- always the one-third of the configured values on DIS
 - to detect the outage more readily

23.5.2. Link State PDU

- one single LSP
- may be fragmented by originator because of MTU
- uniquely identified by LSPID
 - System Id
 - Pseudo Node:
 - Fragment Number
- uses Sequence Number for different version of the same LSP
- has a Remaining Lifetime
 - default to 20 minutes
 - refreshes every 15 minutes

If the LSP's Remaining Lifetime decreases to 0, the router will delete the LSP's body from the link-state database, keep only its header, and advertise the empty LSP with the Remaining Lifetime set to 0. Flooding an empty LSP with the Remaining Lifetime set to 0 is called an LSP purge. Router purging an LSP will not flush the LSP from its link-state database just yet, though. The expired LSP can be purged from the link-state database after an additional time called ZeroAgeLifetime set to 60 seconds. This is done to ensure that the LSP's header is retained until the purged LSP has been safely propagated to all neighbors. Cisco routers, however, appear to hold the empty LSP header for another 20 minutes.



23.5.3. Complete Sequence Numbers PDU

- contains list of LSPID
- doesn't contain the LSP body(similar to OSPF's DBD)
- exchanged only during initialization on P2P links
- sent periodically by DIS on bcast networks
 - Receivers of CSNP packets can compare their link-state database contents to the list of LSPs in the CSNP and perform appropriate action—flood a newer or missing LSP if they have one, or request an LSP if they find it missing in their own database.
 - If the sender's link-state database contains so many LSPs that listing them all in a single CSNP packet would cause it to exceed the MTU, multiple CSNPs are sent. For this purpose, the individual LSPIDs to be advertised are first sorted as integer numbers in ascending order. Each CSNP contains information about the Start LSPID and End LSPID that is described by this CSNP. The full range of possible LSPIDs starts with the value of 0000.0000.0000.00-00 (the bold part is the System ID, the following octet is the Pseudonode ID, and the octet following the dash is the LSP Number ID), and ends with the value of FFFF.FFFF.FFFF.FF-FF. If all LSPs can be listed in a single CSNP, the Start and End LSPIDs will use these respective values. If it is necessary to send more CSNPs, the first CSNP will have the 0000.0000.0000.00-00 as the Start LSPID, and the End LSPID will be set to the LSPID of the

last entry in this CSNP. In the following CSNPs, the Start and End LSPIDs will be set to the respective LSPIDs of the first and last entry, sorted in ascending order. The last CSNP will have the value of FFFF.FFFF.FFFF.FF-FF as the End LSPID. This sorting of LSPIDs into ascending number and CSNPs sequentially listing all LSPIDs from the allowable range are the reasons for calling these PDUs Sequence Numbers PDUs.

23.5.4. Partial Sequence Numbers PDU

- used to request an LSP or acknowledge its successful arrival.

23.6. Network Types

- Broadcast (multi-access)
- P2P

23.6.1. Point-to-Point Links

- On point-to-point interfaces, IS-IS expects to detect a single neighbor, bring up an adjacency, and then synchronize link-state databases.
- three-way handshake:
 - configured with **isis three-way-handshake {cisco | ietf}**
 - introduces Extended Local Circuit ID (4 octets)
 - contains adjacency state TLV with
 - Adjacency Three Way State: This is the state of adjacency as seen by the sending router.
 - Extended Local Circuit ID: This is the ID of the sending router's interface.
 - Neighbor System ID: This value is set to the ID of the neighboring router whose IIHs have been successfully received.
 - Neighbor Extended Local Circuit ID: This value is set to the Extended Local Circuit ID field value from the neighbor's IIH packets.
 - After the adjacency is declared as Up, routers will attempt to synchronize their link-state databases.
 - Both routers will mark all their LSPs for flooding over the point-to-point link; plus they send CSNP packets to each other.
 - Because the IS-IS standard assumes that the actual transmission of LSPs marked for flooding is driven by a periodically scheduled process, it is possible that the CSNP packets are exchanged before the LSP transmission takes place. If a router learns from the received CSNP that its neighbor already has an LSP that is scheduled to be sent, the router will unmark the LSP, removing it from the set of LSPs to be flooded. This way, only the LSPs missing from the neighbor's database will be sent to it. In addition, if a router learns from the received CSNP that the neighbor has LSPs that are newer or unknown, it will request them using a PSNP packet. Note that neither of these is necessary, as both routers nonetheless initially set up all their LSPs to be flooded across the link, without the aid of CSNP or PSNP packets. The initial sending of CSNPs to compare the link-state

databases and PSNPs to request missing or updated entries increases the resiliency of the synchronization process but is not strictly necessary: Without these packets, routers will simply exchange the full link-state database.

23.6.2. Broadcast Networks

- Routers must create adjacencies, synchronize their databases, and keep them synchronized
- on Ethernet networks,
 - encapsulates IEEE 802.2 LLC frames to DSAP and SSAP set to 0xFE
 - sends L1 packets to mcast 0180.c200.0014
 - sends L2 packets to mcast 0180.c200.0015
- detects neighbors with IIH

Task: Configure the IS-IS Priority

```
(config-if)# isis priority <0-255>
```



Priority 0 doesn't exclude the router from the election

DIS

- helps routers on bcast segment to synchronize with periodic flooding of CSNPs
- represents the bcast segment as the pseudo-node
- doesn't elect/need backup DIS
- elected with each IIH based on Highest
 - Interface priority
 - SNPA
 - System ID: when SNPA are not comparable
 - Frame Relay DLCI vs ATM VPI/VCI

23.7. Areas

- it is possible to configure up to three different NSAP addresses on an IS-IS router in a single IS-IS instance, provided that the System ID in all NSAP addresses is identical and the NSAP addresses differ only in their Area ID.
 - A router with multiple NSAP addresses will nonetheless maintain only a single link-state database, causing all configured areas to merge together. This behavior is useful when splitting, joining, or renumbering areas.
 - For example, when you are renumbering an area, all routers are first added a second NSAP address with the new Area ID and then the old NSAP address is removed—without causing any adjacencies between routers to flap.
 - Similarly, when you are joining two areas, routers in an annexed area are given the new

NSAP with the same Area ID as the area into which they are being joined, and afterward, the old NSAP is removed. Splitting an area again uses a similar approach—first add the new NSAP address to all routers, and afterward, remove the former NSAP address.

- L1 routers advertise directly connected networks
- L2 routers advertise directly connected networks, + all other L1 networks in its own area

23.8. Authentication

- authenticates IIH independently of LSP, CSNP and PSNP packets
- L1 routers must have the same L1 area password
- L2 routers must have the same L2 domain password
- If security is a major concern, different passwords for L1 IIH, L2 IIH, L1 non-IIH, and L2 non-IIH packets can be configured.

Task: Configure IIH Authentication

```
(config-if)# isis authentication mode {text | md5} [level-1|level-2]
(config-if)# isis auth key-chain name [level-1|level-2]
```

Task: Configure LSP, CSNP and PSNP Authentication

```
(config-router)# authentication mode {text | md5} [level-1|level-2]
(config-router)# authentication key-chain name [level-1|level-2]
```



In the previous commands , if the **level-1** or **level-2** keyword is omitted from a command where it is currently indicated, the corresponding authentication type will be activated for both levels.

23.9. IPv6 Support

- Supports out-of-the box

TODO:

Chapter 24. BGP

Configuration guides > IP Routing > [BGP](#)

- Exterior gateway protocol
- Creates loop-free inter-domain routing between AS.
- Path vector algorithm = (distance vector + AS-path loop detection)
- TCP 179
- AD: external 20 , internal and local 200
- RFC 1771

24.1. BGP Message Format

- Minimum size: 19 bytes
- Maximum size: 4096 bytes

24.1.1. BGP Header

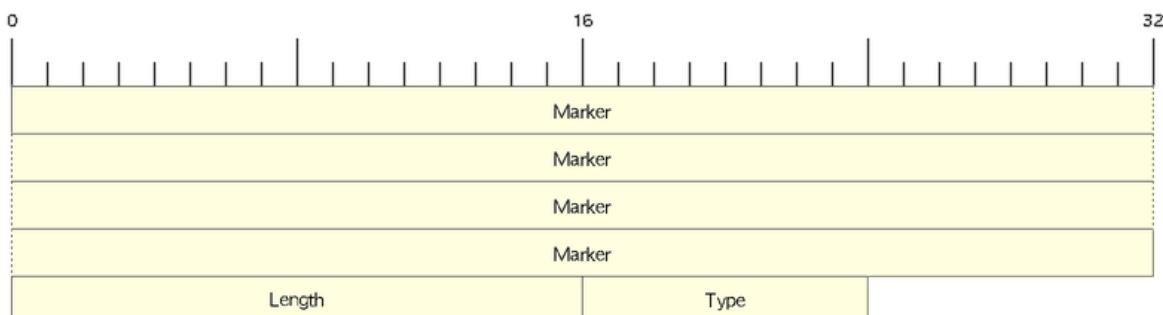


Figure 43. BGP Header Format

Marker

- 16 bytes
- set to all 1s for OPEN message or if OPEN message without authentication
- computed by the authentication process

Length

- 2 bytes
- Total length in bytes of the message including the header

Type

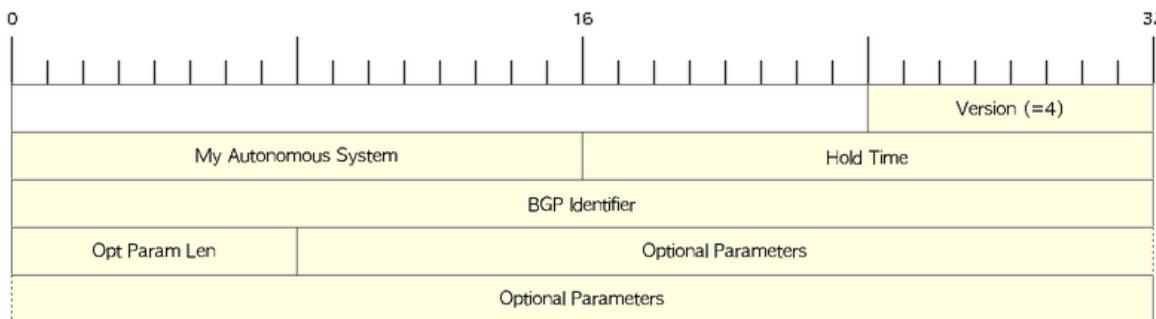
- 1 byte
- Indicates message type (1: Open, 2: Update, 3: Notification, 4: Keepalive)

24.1.2. OPEN

- Initiates the session
- Contains BGP version , local AS number, BGP router Id



If 2 BGP peers attempt to establish a connection at the same time, the session originated from the device with the higher BGP router-id is maintained. (BGP connection collision detection in RFC 4271)



Version

1 octet

My autonomous system

Hold time

- maximum interval in seconds between successive Keepalive or Update messages.
- A receiver compares the value of the Hold Time and the value of its configured hold time and accepts the smaller value or rejects the connection.
- Can be set to zero to indicates that the connection is always up
- if not set to zero, the minimum recommended hold time is 3 seconds

BGP identifier

- Router ID
- determined by these rules in order of preference at boot or BGP process restart:
 - manually configured Router id
 - highest IP address of an up/up loopback
 - highest IP address of an up/up non-loopack

Optional parameters length

- total length in octects of the following Optional Parameters field

Optional Parameters

- Variable length field containing a triplet <Type: 1 octet,Length: 1 octet,Value>

Task: Configure the BGP Router Id

```
(config-router)# bgp router-id <ip-address>
```

24.1.3. KEEPALIVE

- Every 60 seconds
- Hold-time: 180 seconds

Task: Set the BGP Network Timers

```
(config-router)# timers bgp <keepalive-seconds> <holdtime-seconds>
```

Task: Set the BGP Network Timers for a Specific Neighbor

```
(config-router)# neighbor {<ip-address> | <peer-group-name>} timers <keepalive-seconds> <holdtime-seconds>
```

24.1.4. UPDATE

- Advertises a single feasible route to a peer and/or withdraws multiple unfeasible routes

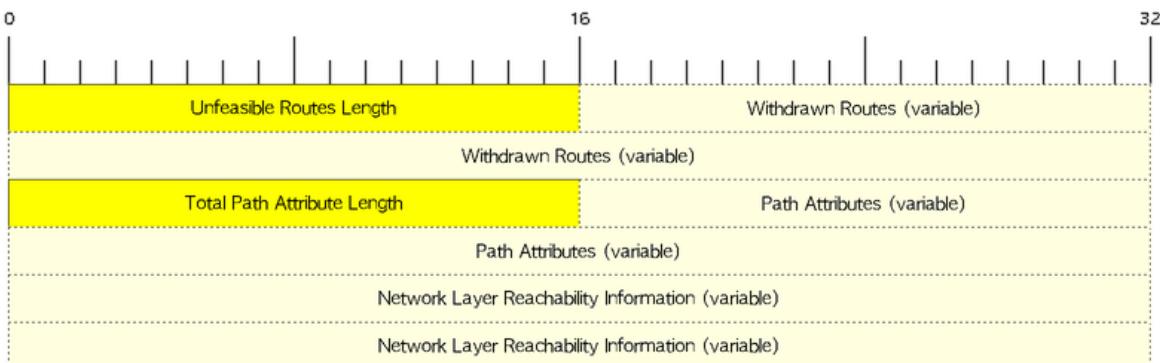


Figure 44. Header Format

Unfeasible Routes Length

- 2-octet field
- total length of the following Withdrawn Routes field, in octets.

Withdrawn Routes

- variable-length
- lists routes to be withdrawn from service.
- Each route in the list is described with a (Length, Prefix) tuple in which the Length is the length of the prefix and the Prefix is the IP address prefix of the withdrawn route.

Total Path Attribute Length

- 2-octet
- total length of the following Path Attribute field, in octets.

Path Attributes

- variable-length
- lists the attributes associated with the NLRI in the following field. Each path attribute is a variable-length triple of (Attribute Type, Attribute Length, Attribute Value). The Attribute Type part of the triple is a 2-octet field consisting of four flag bits, four unused bits, and an Attribute Type code (see [Attribute Type Code](#)).



Figure 45. Attribute Type Part Of the Path Attributes Field

Flag bits (1/0)

- O: Optional / Well-known
- T: Transitive / Non-transitive
- P: Partial / Complete
- E: Extended length / Regular length (2-bytes/ 1-bytes)
- U: Unused

Table 14. Attribute Type Code

Code	Attribute	Category
1	ORIGIN	Well-known mandatory
2	AS_PATH	Well-known mandatory
3	NEXT_HOP	Well-known mandatory
4	MULTI_EXIT_DISC	Optional nontransitive
5	LOCAL_REF	Optional transitive
6	ATOMIC_AGGREGATE	Well-known discretionary
7	AGGREGATOR	Optional transitive
8	COMMUNITY	Optional transitive
9	ORIGINATOR_ID	Optional nontransitive
10	CLUSTER_LIST	Optional nontransitive



tasks for Internet, no-export, no-advertise, local-as

24.1.5. NOTIFICATION

- go out in response to error, fatal condition
- torn down or reset the BGP peer session

24.1.6. BGP FSM States

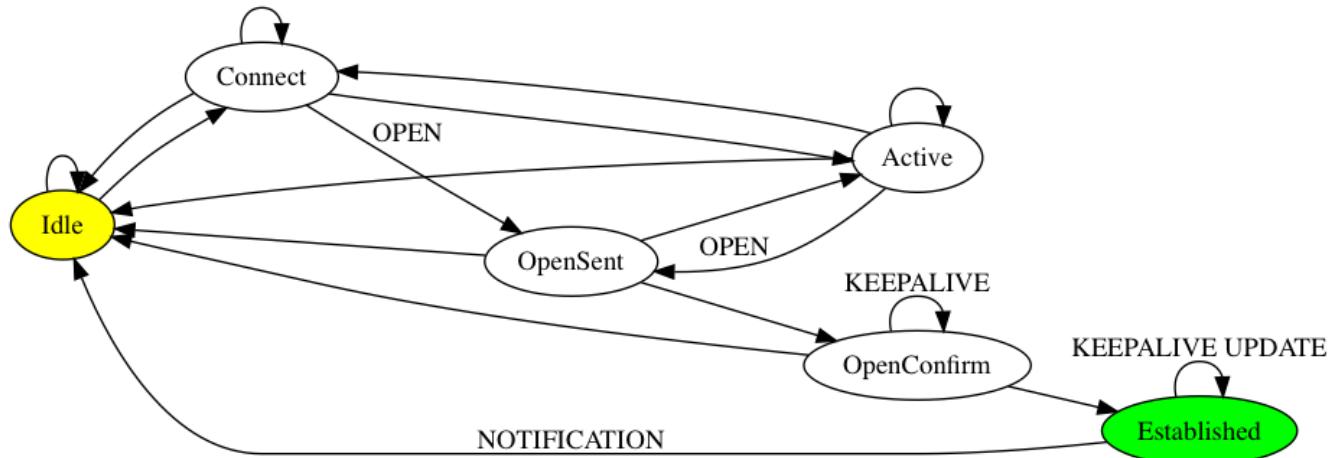


Figure 46. BGP Neighbor Negotiation Finite State Machines

Idle

initial BGP state after enabling BGP process or resetting device.

Connect

waits for a TCP connection with the remote peer. If successful, sends OPEN message. If not, resets the ConnectRetry timer and transitions to Active state.

Active

attempts to initiate a TCP connection with the remote peer. If successful, sends OPEN message. If not, resets ConnectRetry timer and transitions back to Connect state

OpenSent

TCP connection up and OPEN message sent, transition to OpenReceive state and wait for initial keepalive to move into OpenConfirm state. If TCP session disconnect, terminate BGP session, reset ConnectRetry timer, move back to Active State.

OpenConfirm

OPEN messages sent and received. Wait for KEEPALIVE

Established

KEEPALIVE received, neighbor parameters match. the BGP peer session is fully established. UPDATE messages containing routing information will now be sent.

- If peer stuck in **Active** state, potential problems can include:
 - no IP connectivity
 - incorrect **neighbor** statement

- access-list filtering TCP port 179

TODO: To display transitions from idle to established with debug ip bgp

```
R1(config)# router bgp 123
R1(config-router)# no neigh 172.16.16.6 shutdown
*Mar 4 21:02:16.958: BGP: 172.16.16.6 went from
*Mar 4 21:02:16.958: BGP: 172.16.16.6 , delay 15571ms
*Mar 4 21:02:29.378: BGP: 172.16.16.6
*Mar 4 21:02:29.382: BGP: 172.16.16.6 rcv message type 1, length (excl. header) 26
*Mar 4 21:02:29.382: BGP: 172.16.16.6 rcv OPEN, version 4, holdtime 180 seconds *Mar 4
21:02:29.382: BGP: 172.16.16.6 went from
*Mar 4 21:02:29.382: BGP: 172.16.16.6 , version 4, ,
holdtime 180 seconds
*Mar 4 21:02:29.382: BGP: 172.16.16.6 w/ OPTION parameter len: 16 BGP: 172.16.16.6
*Mar 4 21:02:29.382: BGP: 172.16.16.6 went from OpenSent to OpenConfirm
*Mar 4 21:02:29.382: BGP: 172.16.16.6 send message type 1, length (incl. header) 45
*Mar 4 21:02:29.394: BGP: 172.16.16.6 went from
```

24.2. Autonomous Systems

- AS: set of routers under a single technical administration
- AS can be:
 - stub : only one exit
 - multihomed: multiple connections with the one or multiple providers
 - transit: allows traffic with origin and destination outside the AS
 - non-transit:

24.2.1. ASN Format

- 2-byte (RFC 4271)
 - 0 - 65535
 - reserved: 0, 65535
 - public use: 1 - 64495
 - documentation: 64496-64511 (RFC 5398)
 - private use: 64512 - 65534
- 4-byte (RFC 5396)
 - Asplain: decimal value notation for 2-byte and 4-byte ASNs
 - Asdot: decimal value notation for 2-byte and dot notation for 4-byte ASN
 - Documentation: 65536-65551 (RFC 5398)
- AS 23456: reserved for gradual transition from 2-byte to 4-byte (RFC 4893)

Task: Modify the Default Output and Regex Match Format for 4-Byte ASN

```
(config-router)# bgp asnotation dot
```

24.3. BGP Peers

- Manually configured and not automatically discovered
- Formed over a TCP connection
- Exchanges PA(Path Attributes) and NLRI (IP/prefix) with the same PA
- Starts with full BGP routing table then incremental updates
- Keeps table version number
- iBGP peers
 - same AS
 - must be fully meshed within AS
- eBGP peers
 - different AS
 - by default, one hop away but you can change that with **ebgp-multihop**

Task: Configure Neighbor

```
(config-router)# neighbor <ip-address> remote-as <asn>
```

Task: Enable the Neighbor to Exchange Prefixes for the Ipv4 Unicast Address Family with the Local Device

```
(config-router)# address-family ipv4 [unicast | multicast | vrf <name>]  
!TODO check the mode  
(config-router)# neighbor <ip-address> activate
```

Task: Display Info About the TCP and BGP Connection to Neighbors

```
# sh ip bgp neighbors <ip-address>
```

24.4. BGP Peer Groups

- Group of peers with the same update policies (outbound route maps, distribute lists, filter lists, update source ,)
- Benefits:
 - simplify configuration
 - make configuration updates more efficient
- Restrictions for eBGP peers:

Task: Create a BGP Peer Group

```
(config-router)# neighbor <peer-group-name> peer-group
```

Task: Assign a Neighbor to a Peer Group

```
(config-router)# neighbor <ip-address> peer-group <name>
```

Task: Add a Text Description with a Specified Peer Group

```
(config-router)# neighbor <peer-group-name> description <text>
```

Task: Disable a BGP Peer or Peer Group

```
(config-router)# neighbor <ip-address> shutdown
```

24.5. BGP Session Reset

- Whenever the routing policy changes due to a configuration change
- Can be hard reset, soft reset or dynamic inbound soft reset

Task: Clear and Reset BGP Neighbor Sessions

```
# clear ip bgp *
```

Task: Enable Logging Of BGP Neighbor Resets

```
(config-router)# bgp log-neighbor-changes
```

Task: Clear BGP Update Group Membership and Recalculate BGP Update Groups

```
# clear ip bgp update-group [ <index-group> | <ip-address> ]
```

24.5.1. Hard Reset

- Tears down the peering sessions including the TCP connections
- Deletes prefixes learned from the peers.
- Pros: no memory overhead

24.5.2. Soft Reset

- Stores prefix information
- Do not tear down existing peering sessions

- Can be configured for inbound or outbound sessions

Task: Configure a BGP Speaker to Perform Inbound Soft Reconfiguration for Peers That Do Not Support the Route Refresh Capability.

```
(config-router)# bgp soft-reconfig-backup
```

Task: Start Storing Updates for Each Neighbor That Do Not Support Route Refresh

```
(config-router)# neighbor <ip-address|peer-group-name> soft-reconfiguration [inbound]
```



- All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates.
- Memory requirements can increase.

24.5.3. Dynamic Inbound Soft Reset

- Do not store update information locally
- Relies on dynamic exchanges with supporting peers
- The peer supports the capability if **show ip bgp neighbors** displays *Received route refresh capability from peer*.
- Use **bgp soft-reconfig-backup** to store updates for peers who do not support the refresh route capability

24.5.4. Routing Policy Change Management

TODO: add this part under bgp reset

24.6. BGP Route Aggregation

- 2 methods
 - basic route redistribution: creates an aggregate route, then redistributes the routes in BGP
 - conditional aggregation: creates an aggregate route, then advertises or not certain routes based on route maps, AS-SET, or summary information
- **bgp suppress-inactive** stops BGP to advertise inactive routes (not installed into the RIB) to any peer.

24.6.1. BGP Route Aggregation Generating AS_SET Information

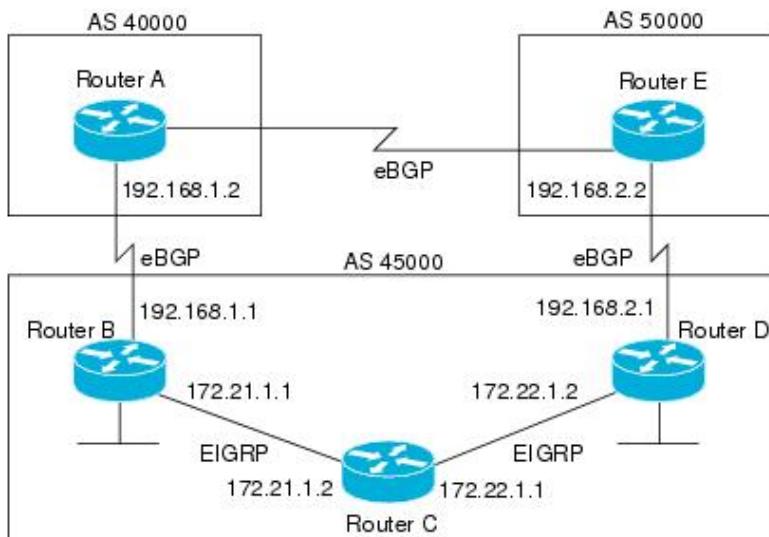
#TODO: improve this part

AS_SET information can be generated when BGP routes are aggregated using the aggregate-address

command. The path advertised for such a route is an AS_SET consisting of all the elements, including the communities, contained in all the paths that are being summarized. If the AS_PATHs to be aggregated are identical, only the AS_PATH is advertised. The ATOMIC-AGGREGATE attribute, set by default for the aggregate-address command, is not added to the AS_SET.

24.7. BGP Backdoor Routes

- Use **network backdoor** to cause BGP to prefer EIGRP



Task: Indicate a Network Reachable Through a Backdoor Route

```
(config-router)# network <ip-address> backdoor
```

24.8. Best Path Selection Algorithm

1. Reachable Next Hop (Well-Known Mandatory)
2. Highest Weight
3. Highest Local Pref
4. Locally Originated Paths (Network, Distribute, Aggregate-Summary) Over Externally Originated Paths
5. Shortest AS Path
6. Lowest Origin Type (Internal Over External Over Incomplete)
7. Lowest MED
8. EBGP Paths Over IBGP Paths
9. Lowest IGP Cost
10. Oldest Path
11. Lowest BGP Router Id



“We Love Oranges AS Oranges Mean Pure Refreshment”. W Weight (Highest) L Local_Pref (Highest) O Originate (local originate) AS As_Path (shortest) O Origin Code (IGP < EGP < Incomplete) M MED (lowest) P Paths (External Paths preferred Over Internal) R Router ID (lowest)



wise lip lovers apply oral medication every night

24.9. Community Attributes

- No-advertise: prevents advertisements to any BGP peer
- No-export: prevents advertisements to any eBGP peer
- local-as: prevents advertisements outside the AS, or in confederation scenarios, outside the sub-AS
- Internet: advertises routes to any peer

24.10. BGP Routing Process

Task: Configure a BGP Routing Process

```
(config)# router bgp <asn>
```

Task: Specify a Network As Local to the BGP Routing Table

```
(config-router)# network <prefix> [mask <a.b.c.d>] [route-map <name>]
```

Task: Disable the IPv4 Unicast Address Family for the BGP Routing Process

```
no bgp default ipv4-unicast
```

Task: Add a Text Description with a Specified Neighbor

```
(config-router)# neighbor <ip-address> description <text>
```

- Apply a route map to incoming or outgoing routes

```
(config-router)# neighbor <ip-address|peer-group-name> route-map <name> [in | out]
```

24.10.1. Aggregating Route Prefixes Using BGP

Task: Redistribute Static Routes Into the BGP Routing Table

```
(config-router)# redistribute static
```

Task: Create an Aggregate Entry In a BGP Routing Table

```
(config-router)# aggregate-address <prefix> <mask> [as-set]
```

Task: Create an Aggregate Route and Suppress Advertisements Of More-Specific Routes to All Peers

```
(config-router)# aggregate-address <prefix> <mask> [summary-only]
```

Task: Create an Aggregate Route but Suppress Advertisement Of Specified Routes

```
(config-router)# aggregate-address <prefix> <mask> [suppress-map <map-name>]
```

Task: Selectively Advertises Routes Previously Suppressed by the Aggregate-Address Command

```
(config-router)# neighbor <ip-address | peer-group-name> unsuppress-map <map-name>
```

- Conditionally advertise BGP routes

The routes or prefixes that will be conditionally advertised are defined in two route maps: an advertise map and either an exist map or nonexist map. The route map associated with the exist map or nonexist map specifies the prefix that the BGP speaker will track. The route map associated with the advertise map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

- If a prefix is found to be present in the exist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If a prefix is found not to be present in the nonexist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised must exist in the BGP routing table in order for conditional advertisement to occur. These routes are referenced from an access list or an IP prefix list.

Task: Advertise Selectively Some BGP Routes to Neighbor

```
(config-router)# neighbor <ip-address> advertise-map <name-1> { exist-map <name> |  
non-exist-map <name>}
```

Task: Inject More Specific Prefixes Into a BGP Routing Table Over Less Specific Prefixes

```
(config-router)# bgp inject-map <name> exist-map <name> [copy-attributes]
```

24.11. BGP Routes

Task: Advertise a Default Route to BGP Peers

```
(config-router)# neighbor <ip-address> default-originate [route-map <name>]
```

Task: Suppress Inactive Route Advertisement Using BGP

- Suppress inactive route advertisement

```
(config-router-af)# bgp suppress-inactive
```

24.12. Peer Session Template

Task: Create a Peer Session Template

```
(config-router)# template peer-session <name>
```

Task: Inherit the Configuration Of Another Peer Session Template

```
(config-router-stmp)# inherit peer-session <template-name>
```

Task: Send a Peer Session Template to a Neighbor So That the Neighbor Can Inherit the Configuration

```
(config-router)# neighbor <ip-address> inherit peer-session <template-name>
```

24.13. Peer Policy Template

Task: Create a Peer Policy Template

```
(config-router)# template peer-policy <name>
```

Task: Configure the Maximum Number Of Prefixes That a Neighbor Will Accept from This Peer

```
(config-router-ptmp)# maximum-prefix <limit> [<threshold>] [restart <interval> | warning-only]
```

- A peer policy template can directly or indirectly inherit up to 8 peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

24.14. BGP Routing Table

Task: Display the Entries In the BGP Routing Table

```
# sh ip bgp [prefix] [mask]
```

- Verify that the VRF instance has been created

```
# show ip vrf
```

- Display information about all the BGP paths in the database

```
# show ip bgp paths
```

- Display the status of all BGP connections

```
# show ip bgp summary
```

- Display IPv4 multicast database-related information

```
show ip bgp ipv4 multicast <command>
```

- Display injected paths

```
# show ip bgp injected-paths
```

```
BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes:i - IGP, e - EGP, ? - incomplete
      Network          Next Hop          Metric LocPrf Weight Path
*> 172.16.0.0        10.0.0.2            0 ? 
*> 172.17.0.0/16    10.0.0.2            0 ?
```

- Display update replication stats for BGP update groups

```
# show ip bgp replication [<index-group> | <ip-address>] [summary]
```

- Display BGP routes that are not installed in the RIB

```
# show ip bgp rib-failure
```

Network	Next Hop	RIB-failure	RIB-NH Matches
10.1.15.0/24	10.1.35.5	Higher admin distance	n/a
10.1.16.0/24	10.1.15.1	Higher admin distance	n/a

TODO: Add section on RIB-failure



The RIB-failure output is an informational message to let us know that although the BGP route is valid, it is not being installed in the routing table. This usually occurs when there is an identical match to a BGP route via an IGP route with a lower administrative distance. In the output below, we can see that the external EIGRP route with a distance of 170 prevents the iBGP route from being installed, which would normally have a distance of 200:



RIB-failure by itself is not necessarily bad, but there are certain cases in which this disconnect between the BGP table and the routing table can cause traffic loops. By default, BGP routes that have RIB-failure can be advertised to other neighbors, because the command `no bgp suppress-inactive` is the default option under the routing process. To stop RIB-failure routes from being advertised, issue the `bgp suppress-inactive` command under the process.

- Display locally configured peer session template

```
show ip bgp template peer-session
```

24.15. Troubleshoot

Task: Display Info About the Processing Of BGP Update Groups.

```
# debug ip bgp groups
```

24.16. Todos

- Concept: bgp route aggregation generating AS_SET information
- Multiprotocol bgp concepts
- Multiprotocol bgp extensions for IP multicast concepts
- AFI bgp address family identifier model : ipv4, ipv6,clns, vpng4

24.17. BGP PIC

- BGP Prefix-Independent Convergence for IP and MPLS-VPN feature

- creates and stores a backup/alternate path in the RIB,FIB, and CEF so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover.

How BGP Converges Under Normal Circumstances

Under normal circumstances, BGP can take several seconds to a few minutes to converge after a network change. At a high level, BGP goes through the following process:

- BGP learns of failures through either IGP or BFD events or interface events.
- BGP withdraws the routes from the RIB, and the RIB withdraws the routes from the FIB and dFIB. This process clears the data path for the affected prefixes.
- BGP sends withdraw messages to its neighbors.
- BGP calculates the next best path to the affected prefixes.
- BGP inserts the next best path for affected prefixes into the RIB, and the RIB installs them in the FIB and dFIB.

This process takes a few seconds or a few minutes to complete, depending on the latency of the network, the convergence time across the network, and the local load on the devices. The data plane converges only after the control plane converges.



When BGP PIC is enabled, CEF recursion is disabled when next-hop is learned via /32 mask or next-hop is directly connected

Read more [details](#)

24.18. BGP TTL Security Check

TTL Security Check is a security feature that protects BGP peers from multi-hop attacks. This feature is based on the Generalized TTL Security Mechanism (GTSM, RFC 3682), and is currently available for BGP. Work is currently in progress to implement this feature for other routing protocols such as OSPF and EIGRP.

TTL Security Check allows the configuration of a minimum acceptable TTL value for the packets exchanged between two eBGP peers. When enabled, both peering routers transmit all their traffic to each other with a TTL of 255. In addition, routers establish a peering session only if the other eBGP peer sends packets with a TTL equal to or greater than the TTL value configured for the peering session. All packets received with TTL values less than the predefined value are silently discarded.

Task: Enable TTL security check between BGP peers

```
(config-router)# neighbor <a.b.c.d> ttl-security hops <count>
```

TODO Add section on bgp network statement

Unlike the network statement used in IGP protocols configuration, the BGP version of the command is different. The basic command syntax is simple: `network <subnet> mask <netmask>`. It does not define a group of interfaces to enable the protocol, it only specifies the prefix in the IGP table (RIB) to be imported into BGP LocRIB. The term LocRIB stands for Local RIB and is another name for BGP table, which is separate from the routing table (RIB). For the prefix to be imported, it must exactly match the specification; it should have the same subnet number and network mask. For example, if you have interface Loopback0 with the IP address 150.1.1.1/24, the command would be `network 150.1.1.0 mask 255.255.255.0`, not `network 150.1.1.1 mask 255.255.255.255`. The second statement will not match any route in the IGP and therefore will not import any prefix. Notice that you may omit the mask specification if it matches the default mask for the IPv4 address class (such as 255.255.255.0 for class C)

TODO add section on next-hop

When peering with another AS, the common question is how to deal with the external next-hop. One way is to use the `next-hop-self` parameter when peering via iBGP or advertising the external link subnet into IGP. Another way is to advertise the link subnet into BGP and thus propagate it to all iBGP peers. All BGP routers will install it into their RIBs and perform a recursive lookup to find the actual next hop for every BGP prefix learned from the external AS.

24.19. BGP Auto-Summary

TODO rewrite this section

BGP auto-summarization is the legacy feature that automatically summarizes network prefixes to their classful boundaries when the prefixes are advertised into BGP. The automatic summarization starts working when you enable it using the command `auto-summary` under BGP process configuration. It only applies in the following two cases:

A network command is configured with a classful subnet, such as `network 54.0.0.0` or `network 155.1.0.0` or `network 192.168.1.0`. In this case, the classful aggregate is installed into the BGP table if there is a prefix in the IGP table that is a subnet to the classful network. For example, if you advertise `network 150.1.0.0`, it would work if any of the prefixes (150.1.2.0/24 or 150.1.3.0/24, etc.) are in the IGP table. This is contrary to the regular exact match requirement imposed by the BGP network statements. Prefixes are advertised into BGP using route redistribution. All redistributed networks are subject to auto-summarization; that is, only the major classful subnets are installed in the BGP table.

Chapter 25. Redistribution

- Redistribution occurs from the routing table not the routing database
- When redistributing protocol X into Y, take ...
 - routes in the RIB via protocol X
 - connected interfaces running protocol X
- choose
 - routes with lower AD

25.1. Administrative Distance

Route source	Distance
Connected route	0
Static route	1
summary EIGRP	5
eBGP	20
internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
ODR	160
External EIGRP	170
iBGP	200
Unknown	255

25.2. Spot Issues

- Loops cannot occur with one single point of redistribution
- Loops may occur with multiple points of redistribution

25.3. Heuristics

- identify each domain and associate a tag
- assign tags to each domain

```
route-map ospf2eigrp permit  
    set tag 123
```

- deny tag on re-entry
 - always block routes to be re-enter the domain
 - optionally: block routes as per scenario requirement

```
! block own routes
route-map ospf2eigrp deny 10
  match tag 456
! block some routes if requested
route-map deny 20
  match tag 78
```

- all tags to pass through transit domains without re-tagging them

```
! identify the transit tags without tagging
route-map ospf2eigrp permit 60
  match tag 234
```

- Use BGP community instead of tags for BGP

```
route-map ospf2bgp permit 70
  set community 110
```

```
ip community-list 1 permit 10
  match community 1
  set tag 110
```

25.4. Connected Routes

Task: Redistribute Connected Routes

```
redistribute connected
```



- Override the implicit redistribution of interfaces running the protocol X

25.5. Static Routes

Task: Redistribute Static Route

```
(config-router)# redistribute static route-map <name> metric <value>
(config-router)# default-metric <hops>
```

25.6. RIP

- doesn't differentiate between internal and external routes
- no default seed metric
 - recommendation: use 1 as default-metric

Task: Prevent Loss Of Packet When BGP Routes Are Redistributed In RIP

```
(config)# router rip  
(config-router)# input-queue 1024
```

25.7. EIGRP

```
redistribute <protocol> metric <bandwidth> <delay> <load> <reliability> <MTU>
```

- internal routes AD < external routes
- uses router-id for loop prevention
- no default seed metric unless EIGRP to EIGRP
 - default-metric <bandwidth> <delay> <load> <reliability> <MTU>
 - default-metric 10000 100 1 255 1500



Duplicate router-ids will prevent EIGRP to install routes

25.8. OSPF Redistribution

- differentiates between internal and external routes but same AD= 110
- Router-id for flooding loop prevention
- Use **subnets** keyword
- default metric is 1 for BGP and 20 for other IGP
- default metric-type E2/N2
- OSPF path selection TODO: improve this part
 - E1 > E2 > N1 > N2
 - E1 & N1 vs E2 & N2 metrics

```
router ospf 1  
 redistribute rip  
 redistribute eigrp  
 default-metric 10
```

Task: Assign Different AD to Internal and External

25.9. BGP Redistribution

25.9.1. IGP to BGP

- denies OSPF external routes by default

Task: Redistribute OSPF Into BGP

```
redistribute ospf <pid> match internal external
```

25.9.2. BGP to IGP

- iBGP routes denied by default, eBGP routes win

Part III : VPN Technologies

Chapter 26. VRF

Virtual Routing and Forwarding

- create multiple virtual, isolated networks, where each technically has its own separate RIB and FIB .

steps

```
conf t
ip routing
ip vrf <case-sensitive-name>
  rd <route-distinguisher>
  route-target {export|import|both} <rt-ext-community>
  import map <route-map>
  export map <route-map>
interface <type slot/subslot/port>
  ip vrf forwarding <vrf-name>
```

26.1. VRF definition

- old command for ipv4 only **ip vrf <name>**
- new command for ipv4 and ipv6 address family

Task: Create a VRF

```
(config)# vrf definition <name>
(config-vrf)# description <same useful information>
```

26.2. Route Distinguisher

- 64-bit prepended to every route in the respective VRF routing table
- create unique VPNv4 prefix in case two VPNs contain the same prefixes
- the first 2 bytes represent the RD type
- the last 6 bytes can be
 - Common format: ASN(2bytes):Site(4bytes) or ASN(4bytes):Site(2bytes)
 - Alternative format: a.b.c.d:NN

26.3. Route Target

- RT is Path attribute of the NLRI
- 1 or + for each RD/prefix
- useful in overlapping VPNs or central service VPN offered by SP

- Allows for granular control of traffic

task: import or export route target communities for the specified VRF

```
(config)# route-target {import|export|both} <ext-community>
```

26.4. VRF Interface

- can be physical (ethernet) or logical (SVI)
- belongs to only one VRF
- packets received are routed and forwarded using the associated VRF table

Task: Assign an interface to a VRF

```
(config-if)# ip vrf forwarding <vrf-name>
```



this command will erase all existing IP addresses configured on the interface to avoid potential address duplication in the new routing table.

26.5. VRF Static Route

Task: Create a VRF-bound static route

```
(config)# ip route vrf <name> <prefix> <mask> [<interface>] [next-hop]
```



with multi-access interfaces, specify the next-hop associated with the interface subnet because Cisco IOS will install a CEF entry in the source VRF using the information provided and will not attempt to resolve the next-hop recursively. Remember that this trick only works with the non-recursive static routes that use directly connected interfaces.

26.6. VRF lite

- extend VRFs beyond a single router by properly mapping the VRFs to the links connecting two routers.
- simplest way of creating non-overlapping VPNs
- poor scalability because each VPN needs a dedicated inter-router links
 - Example: for two routers and 100 VPNs, you must provision 100 connections between the two routers, one for every VPN.
- The connection could be either a separate interface or some Layer 2 virtualization technique, such as Frame-Relay PVC or Ethernet VLAN.

26.7. Multi-VRF

TODO Maybe this belongs to the mpls l3vpns section

Chapter 27. MPLS

- Protocol number: 137 ???
- packet-forwarding technology which uses labels in order to make data forwarding decisions
- LSP Label Switching Path
- FEC forwarding equivalence class: a group of IP packets which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment)

27.1. MPLS Label Stack

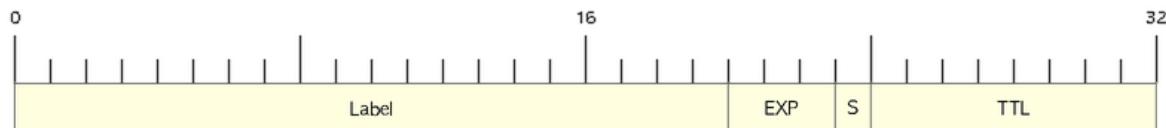
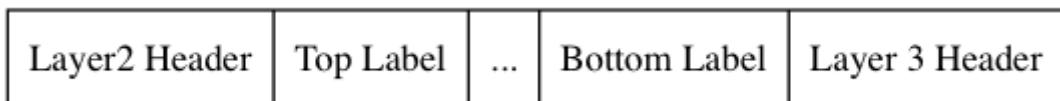


Figure 47. MPLS Header Format

Label

Locally significant to the router

- 0 : IPv4 Explicit NULL Label. indicates that the label stack must be popped, and the packet forwarding must be based on the IPv4 header.
- 1 : Router Alert Label. When a received packet contains this label value at the top of the label stack, it is delivered to a local software module for processing. The actual packet forwarding is determined by the label beneath it in the stack. However, if the packet is forwarded further, the Router Alert Label should be pushed back onto the label stack before forwarding. The use of this label is analogous to the use of the Router Alert Option in IP packets (for example, ping with record route option)
- 2 : IPv6 Explicit NULL Label. indicates that the label stack must be popped, and the packet forwarding must be based on the IPv6 header
- 3 : Implicit NULL Label. never actually appears in the encapsulation. indicates that the LSR pops the top label from the stack and forwards the rest of the packet (labeled or unlabeled) through the outgoing interface (as per the entry in Lfib). Although this value might never appear in the encapsulation, it needs to be specified in the Label Distribution Protocol, so a value is reserved
- 4-15: Reserved

EXP

Experimental, class of service

S

Bottom-of-Stack flag

TTL

Time to live

27.2. Label Distribution

- protocol : LDP (default RFC 3036) or TDP (cisco)

27.3. MPLS ping and traceroute

MPLS LSP ping uses MPLS echo request and reply packets to validate an LSP. You can use MPLS LSP ping to validate IPv4 LDP, AToM, and IPv4 RSVP FECs by using appropriate keywords and arguments with the ping mpls command. The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address. The 127.x.y.z/8 address prevents the IP packet from being IP switched to its destination if the LSP is broken.

An MPLS echo reply is sent in response to an MPLS echo request. The reply is sent as an IP packet and it is forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address obtained from the router generating the echo reply. The destination address is the source address of the router that originated the MPLS echo request packet. The MPLS echo reply destination port is set to the echo request source port.

MPLS LSP traceroute uses MPLS echo request and reply packets to validate an LSP. You can use MPLS LSP traceroute to validate IPv4 LDP and IPv4 RSVP FECs by using appropriate keywords and arguments with the trace mpls command. The MPLS LSP Traceroute feature uses TTL settings to force expiration of the TTL along an LSP. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4) to discover the downstream mapping of each successive hop. The success of the LSP traceroute depends on the transit router processing the MPLS echo request when it receives a labeled packet with a TTL = 1. On Cisco routers, when the TTL expires, the packet is sent to the Route Processor (RP) for processing. The transit router returns an MPLS echo reply containing information about the transit hop in response to the TTL-expired MPLS packet. The MPLS echo reply destination port is set to the echo request source port.

Task: Select an LDP IPv4 prefix FEC for validation

```
# ping mpls ipv4 <destination-address/destination-mask-length> [repeat <count>] [exp <bits> ] [verbose]
```

Task: Select an LDP IPv4 prefix FEC for validation

```
# trace mpls ipv4 <destination-address/destination-mask-length>
```

Further Reading <http://goo.gl/V1Z2kN> Good explanation on INE blog

27.4. L3VPNs

- mpls vpn
 - CE : no mpls-aware
 - PE : mpls and vpn aware
 - P : no vpn aware

TODO Place this section at an appropriate place

- Establish an LSP between PEs: IGP + LDP , free BGP core
- Exchange routes with customer: PE-CE IGP or BGP
- Exchange customer routes between PEs: iBGP + MPLS VPN label
- Label switch between PEs: Data follows the IGP + LDP transport label

Read [Route leaking](#)

```
show mpls forwarding-table
```

check <http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/command/mp-s2.html#wp4232274342>

27.5. IPv6 over MPLS: 6PE and 6VPE

- enables the service providers running an MPLS/IPv4 infrastructure to offer IPv6 services without any major changes in the infrastructure
- benefits
 - Minimal operational cost and risk : No impact on existing IPv4 and MPLS services.
 - Only PE routers upgrade : A 6PE and 6VPE router can be an existing PE router or a new one dedicated to IPv6 traffic.
 - No impact on IPv6 CE routers : The ISP can connect to any CE router running Static, IGP or EGP.
 - Production services ready : An ISP can delegate IPv6 prefixes.
 - IPv6 introduction into an existing MPLS service : 6PE and 6VPE routers can be added at any time.

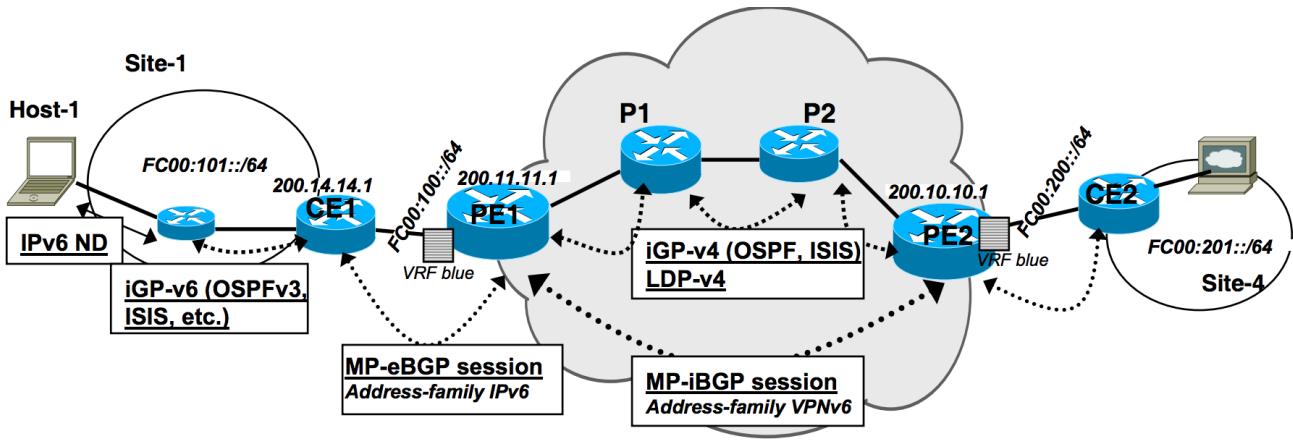


Figure 48. protocols leveraged with 6vpe

More at <https://gixtools.net/wp-content/uploads/2011/05/Cisco-IPv6-Provider-Edge-Router-over-MPLS-Cisco-6PE.pdf>

Chapter 28. LDP

Configuration Guides > MPLS > Label Distribution Protocol

- [RFC 5036](#)
- enables peer label switch routers (LSRs) to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network.
- UDP 646 (TCP 711 for TDP)
- Hellos multicast to 224.0.0.2
- Updates unicast TCP to LDP ID
- supercedes Cisco TDP

28.1. LDP process

Task: Enable LDP at Interface Level

```
(config-if)# mpls ip
```

Task: Specify the label protocol

```
(config)# mpls label protocol {ldp|tdp}
```

Task: Enable Ldp at IGP Process

```
(config-router)#mpls ldp autoconfig
```

Task: Verify Ldp Is Enabled

```
# sh mpls interfaces
```

28.2. Discovery Of Adjacent LDP Peers

- neighbor discovery UDP port 646
 - basic neighbor discovery: multicast hellos to directly connected neighbors
 - extended neighbor discovery: unicast hellos to non-directly connected neighbors

Task: Verify that the interface is up and is sending Discovery Hello messages.

```
# sh mpls ldp discovery [all| vrf <vpn-name>] [detail]  
  
Local LDP Identifier:  
    172.16.12.1:0  
Discovery Sources:  
Interfaces:  
    Ethernet3/0 (ldp): xmit
```

- timers: hello interval (5 seconds) and holdtime (15 seconds)

```
(config)# mpls ldp discovery hello interval <sec>  
(config)# mpls ldp discovery hello holdtime <sec>
```

TODO: test this command

Task: Establish LDP session between devices that are not directly connected

```
(config)# mpls ldp neighbor targeted
```



This command can be used if you want to create LDP session that persists even when the direct link goes down and there is alternative paths.

Task: Respond to request for LDP targeted hellos

```
(config)# mpls ldp neighbor targeted accept
```

28.3. LDP Sessions

in 2 steps:

- Transport connection: if the 2 peers have never established a tcp session, create a new session with a client (active device, highest ip address) using a random port and the server (lowest ip addr) listening on the TCP 646 port.

```
R1#show tcp brief (state)  
TCB      Local_Address   Foreign_Address  
498D80D8  192.1.1.1.646  192.1.5.5.21288  ESTAB
```

- Session establishment:
 - negotiates ldp protocol version, label exchange method, timers
 - if incompatibility, sends error negotiation messages and restart the negotiation with initial backoff value (15 seconds) and maximum backoff value (120 seconds)

```
R1#show mpls ldp parameters
```

```
Protocol version: 1
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
Downstream on Demand Path Vector Limit: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

- the label exchange methods:
 - Unsolicited downstream distribution mode
 - solicited downstream distribution mode

Task: Display the status of LDP session

```
R1#sh mpls ldp neighbor
```

```
Peer LDP Ident: 192.1.5.5:0; Local LDP Ident 192.1.1.1:0
TCP connection: 192.1.5.5.21288 - 192.1.1.1.646
State: Oper; Msgs sent/rcvd: 10/11; Downstream
Up time: 00:04:24
LDP discovery sources:
    FastEthernet0/0, Src IP addr: 172.16.15.5
Addresses bound to peer LDP Ident:
    172.16.15.5 192.1.5.5
```

- with keepalives (60 seconds)

```
(config)# mpls ldp holdtime <sec>
%Previously established sessions may not use the new holdtime.
```

- Keepalive timer is reset every time LDP packets or keepalives are received
- Keepalives are automatically adjusted to 1/3 of the configured holdtime
- Reset the tcp session for new timers to take effect

28.4. LDP label binding, label spaces and identifiers

- LDP label binding: association between a destination prefix and a label
- LDP supports two types of label spaces:
 - Interface-specific: An interface-specific label space uses interface resources for labels.
 - For example, label-controlled ATM (LC-ATM) interfaces use virtual path

identifiers/virtual circuit identifiers (VPIS/VCIs) for labels.

- Depending on its configuration, an LDP platform may support zero, one, or more interface-specific label spaces.
- Platform-wide: An LDP platform supports a single platform-wide label space for use by interfaces that can share the same labels.
 - For Cisco platforms, all interface types, except LC-ATM, use the platform-wide label space.

LDP uses a 6-byte quantity called an LDP Identifier (or LDP ID) to name label spaces. The LDP ID is made up of the following components:

- The first four bytes, called the LDP router ID, identify the LSR that owns the label space.
- The last two bytes, called the local label space ID, identify the label space within the LSR. For the platform-wide label space, the last two bytes of the LDP ID are always both 0.

The LDP ID takes the following form: <LDP router ID> : <local label space ID> The following are examples of LDP IDs: 172.16.0.0:0 , 192.168.0.0:3

The router determines the LDP router ID as follows, if the **mpls ldp router-id** command is not executed,

1. The router examines the IP addresses of all operational interfaces.
2. If these IP addresses include loopback interface addresses, the router selects the largest loopback address as the LDP router ID.
3. Otherwise, the router selects the largest IP address pertaining to an operational interface as the LDP router ID.

The normal (default) method for determining the LDP router ID may result in a router ID that is not usable in certain situations. For example, the router might select an IP address as the LDP router ID that the routing protocol cannot advertise to a neighboring router. The **mpls ldp router-id** command allows you to specify the IP address of an interface as the LDP router ID. Make sure the specified interface is operational so that its IP address can be used as the LDP router ID. When you issue the **mpls ldp router-id** command without the force keyword, the router select selects the IP address of the specified interface (provided that the interface is operational) the next time it is necessary to select an LDP router ID, which is typically the next time the interface is shut down or the address is configured. When you issue the **mpls ldp router-id** command with the force keyword, the effect of the **mpls ldp router-id** command depends on the current state of the specified interface:

Task: Define Router-Id (Recommended)

```
(config)# mpls ldp router-id <interface-type number> [force]
```



- If the interface is up (operational) and if its IP address is not currently the LDP router ID, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.
- If the interface is down (not operational) when the **mpls ldp router-id** force command is issued, when the interface transitions to up, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

Task: Verify LDP Sessions

```
# sh mpls ldp neighbor
```

Task: Troubleshoot LDP Adjacencies

```
# debug mpls ldp transport events
```

Task: Establish a TCP connection using the physical interface IP address

```
(config-if)# mpls ldp discovery transport-address interface.
```

28.5. LDP Session protection

- provides faster LDP convergence when a link recovers following an outage.
- protects an LDP session between directly connected neighbors or an LDP session established for a traffic engineering (TE) tunnel.
- uses LDP Targeted Hellos to protect LDP sessions

Task: Enables MPLS LDP session protection

```
(config)# mpls ldp session protection [vrf <vpn-name>] [for <acl>] [duration {infinite | <seconds>}]
```

28.6. LDP Authentication

- MD5 with same password

Task: Specify authentication between two LDP peers

```
(config)# mpls ldp neighbor [vrf <vpn-name>] ip-address [password [0-7] <password-string> ]
```

Task: Make the use of passwords mandatory between LDP peers

```
(config)# mpls ldp password required
```

28.7. LDP MD5 Global Configuration

- enables LDP MD5 globally instead of on a per-peer basis.
- can set up password requirements for a set of LDP neighbors to prevent unauthorized peers from establishing LDP sessions and to block spoofed TCP messages.
- enhancements
 - You can specify peers for which MD5 protection is required. This can prevent the establishment of LDP sessions with unexpected peers.
 - You can configure passwords for groups of peers. This increases the scalability of LDP password configuration management.
 - The established LDP session with a peer is not automatically torn down when the password for that peer is changed. The new password is used the next time an LDP session is established with the peer.
 - You can control when the new password is used. You can configure the new password on the peer before forcing the use of the new password.
 - If the neighboring nodes support graceful restart, then LDP sessions are gracefully restarted. The LDP MD5 password configuration is checkpointed to the standby Route Processors (RPs). The LDP MD5 password is used by the device when the new active RP attempts to establish LDP sessions with neighbors after the switchover.

TODO more

28.8. LDP Auto-configuration

- enables you to globally enable LDP on every interface associated with an IGP instance.
- supported on OSPF and IS-IS
- provides a means to block LDP from being enabled on interfaces

Task: Enable the MPLS LDP Autoconfiguration feature on OSPF interfaces

```
(config-router)# mpls ldp autoconfig [area <area-id>]
```



If no area is specified, the command applies to all interfaces associated with the OSPF process.

Task: Enables the MPLS LDP Autoconfiguration feature on IS-IS interfaces

```
(config-router)# mpls ldp autoconfig [level 1 | level 2]
```

Task: Disable LDP autoconfiguration on a specified interface

```
(config-if)# no mpls ldp igrp autoconfiguration
```

28.9. LDP outbound label filtering

- By default, LDP will generate and advertise labels for every prefix found in the local routing table.
- Use a standard access-list to select the prefixes eligible for label generation.

Task: Stop the generation of labels for every prefix found in the local routing table

```
(config)# no mpls ldp advertise-labels
```

Task: Select prefixes for LDP label generation

```
(config)# mpls ldp advertise-labels for <acl>
```

28.10. LDP Inbound Label Binding Filtering

Task: control the label bindings a label switch router accepts from its peer LSRs.

```
(config)# mpls ldp neighbor [vrf <vpn-name>] <ip-address> labels accept <acl>
```

Task: Verify that MPLS LDP Inbound Label Bindings are Filtered

```
# show mpls ldp neighbor [vrf <vpn-name>] [<address> | interface] [detail]  
# show ip access-list [<acl>]  
# show mpls ldp bindings
```

28.11. LDP Graceful restart

TODO

Chapter 29. GRE

- Generic Routing Encapsulation
- Tunnelling protocol developed by Cisco
- IP protocol 47
- [RFC 2784](#)
- [RFC 2345](#)
- [RFC 1234](#)

29.1. Tunneling

- Tunneling uses encapsulates data packets with from one protocol inside a different protocol at the same OSI layer and transports the data packets unchanged across a foreign network (which may not support the passenger protocol).
- Unlike encapsulation, tunneling allows a lower-layer protocol, or same-layer protocol, to be carried through the tunnel.
- **Passenger protocol** : The protocol that you are encapsulating. Examples: AppleTalk, IP, IPX.
- **Carrier protocol** : The protocol that does the encapsulating. Examples: GRE, IP-in-IP, L2TP,MPLS, STUN,DLSw+.
- **Transport protocol** : The protocol used to carry the encapsulated protocol. The main transport protocol is IP.

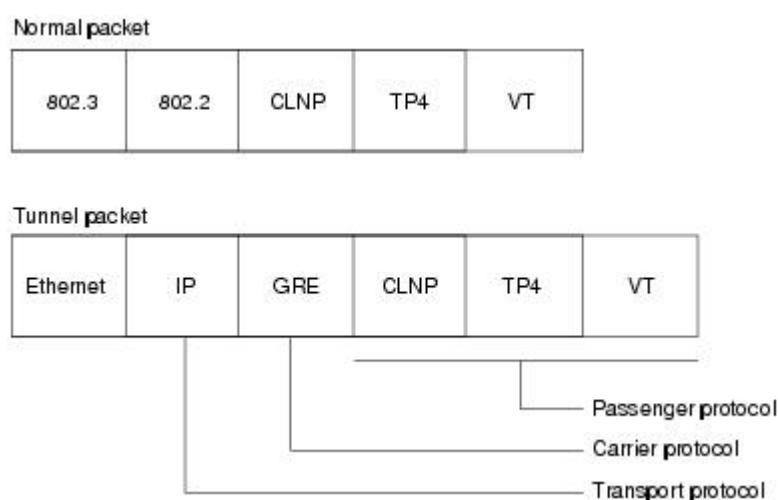
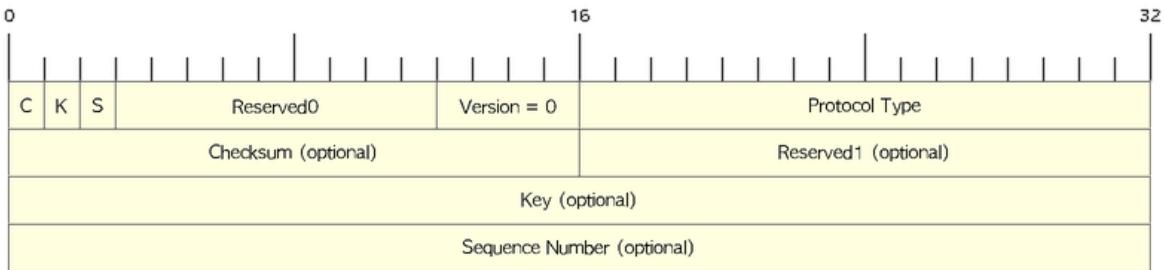


Figure 49. IP Tunneling Terminology and Concepts

29.2. GRE Header



C

- Checksum present
- Set to 1 if the checksum and the Reserved1 field are present

K

- Key bit
- Set to 1 if

Reserved0

- if any of bits 1-5 are non-zero, a receiver must discard the packet unless receiver implements RFC1701

Protocol Type

- Ether protocol type (e.g. IPv4 → 0x800)



GRE adds at least 24 bytes to the original IP (20 bytes for new IP header + 4 bytes for GRE header)

29.3. GRE Keepalive

The GRE tunnel keepalive mechanism gives the ability for one side to originate and receive keepalive packets to and from a remote router even if the remote router does not support GRE keepalives. For GRE keepalives, the sender pre-builds the keepalive response packet inside the original keepalive request packet so that the remote end only needs to do standard GRE decapsulation of the outer GRE IP header and then forward the inner IP GRE packet. GRE tunnel keepalives timers on each side are independent and do not have to match. The problem with the configuration of keepalives only on one side of the tunnel is that only the router that has keepalives configured marks its tunnel interface as down if the keepalive timer expires. The GRE tunnel interface on the other side, where keepalives are not configured, remains up even if the other side of the tunnel is down. The tunnel can become a black-hole for packets directed into the tunnel from the side that did not have keepalives configured.

Specifies the number of times that the device will continue to send keepalive packets without response before bringing the tunnel interface protocol down.

GRE keepalive packets may be configured either on only one side of the tunnel or on both. If GRE keepalive is configured on both sides of the tunnel, the period and retries arguments can be

different at each side of the link.

This command is supported only on GRE point-to-point tunnels.

Task: Configure GRE Keepalives

```
(config-if)# keepalive [ period [retries]]
```

29.4. GRE Tunnel

To build a tunnel, a tunnel interface must be defined on each of two routers and the tunnel interfaces must reference each other. At each router, the tunnel interface must be configured with a L3 address. The tunnel endpoints, tunnel source, and tunnel destination must be defined, and the type of tunnel must be selected.

Optional steps can be performed to customize the tunnel.

Remember to configure the router at each end of the tunnel. If only one side of a tunnel is configured, the tunnel interface may still come up and stay up (unless keepalive is configured), but packets going into the tunnel will be dropped.

Task: Summary Steps

```
interface tunnel <number>
  bandwidth <kbps>
  keepalive [<period> [<retries>]]
  tunnel source {<ip-address> | <interface-type interface-number>}
  tunnel destination { <hostname> | <ip-address>}
  tunnel key <key-number>
  tunnel mode [gre {ip | multipoint} | dvmrp | ipip | mpls | nos]
  ip mtu <bytes>
  ip tcp mss <mss-value>
  tunnel path-mtu-discovery [age-timer {<aging-mins>}| infinite]
```

- The router should have a route to the destination address, but not through the tunnel interface.
- Tunnel Key ID must match on both sides of the tunnel. It is used as a weak form of authentication



29.4.1. Configuration Example

Note that Ethernet interface 0/1 is the tunnel source for Router A and the tunnel destination for Router B. Fast Ethernet interface 0/1 is the tunnel source for Router B and the tunnel destination for Router A.

Router A

```

interface Tunnel0
 ip address 10.1.1.2 255.255.255.0
 tunnel source Ethernet0/1
 tunnel destination 192.168.3.2
 tunnel mode gre ip
!
interface Ethernet0/1
 ip address 192.168.4.2 255.255.255.0

```

Router B

```

interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 tunnel source FastEthernet0/1
 tunnel destination 192.168.4.2
 tunnel mode gre ip
!
interface FastEthernet0/1
 ip address 192.168.3.2 255.255.255.0

```

29.5. GRE backup interface

In this example, R4 and R5 use the backup interface feature along with duplicate routing information to perform both traffic engineering and redundancy. With the backup interface configured on R4's and R5's point-to-point GRE Tunnel100 interface, R4 and R5 wait for the line protocol of Tunnel100 interface to go DOWN before GRE interface Tunnel45 is activated. The following rules and restrictions apply when implementing the backup interface functionality:

The primary/active interface being backed up must be a point-to-point interface type, because its state can be better determined.

The secondary/standby interface acting as backup can be any interface except sub-interface, because the state of the main interface determines the state of sub-interfaces in general

```

(config-if)# backup interface <intf-id>
(config-if)# backup delay <after-down> <before-up>

```

```
# sh backup
```

29.6. Troubleshooting

Three reasons for a GRE tunnel to shut down:

- There is no route to the tunnel destination address.
- The interface that anchors the tunnel source is down.
- The route to the tunnel destination address is through the tunnel itself. “%TUN-5-RECURDOWN:Tunnel0“

With the above three reasons for tunnel shut down are problems local to the router at the tunnel endpoints and do not cover problems in the intervening network.

Also if the two routers tunnel modes do not match, the tunnel interface can still stay in an up/ip state but the routers cannot forward packets because of the mismatch encapsulation.

29.6.1. "%TUN-5-RECURDOWN" Error Message and Flapping EIGRP/OSPF/BGP Neighbors Over a GRE Tunnel

<http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/22327-gre-flap.html>

29.7. Questions

1. What is the minimum amount of additional header that GRE adds to a packet?
 - a. 16 bytes
 - b. 20 bytes
 - c. 24 bytes
 - d. 36 bytes
 - e. 48 bytes
2. Which of the following are valid options in a GRE header (select all that apply)?
 - a. GRE Header Length
 - b. Checksum Present
 - c. Key Present
 - d. External Encryption
 - e. Protocol
3. What is the purpose of a GRE tunnel interface?
 - a. It is always the tunnel source interface.
 - b. It is always the tunnel destination interface.
 - c. It is where the protocol that travels through the tunnel is configured.
 - d. It is the interface that maps to the physical tunnel port.
 - e. It is not used today

http://ptgmedia.pearsoncmg.com/9781587201509/samplechapter/158720150X_CH14.pdf

Chapter 30. DMVPN

- DMVPN = mGRE + NHRP + IPSec

Start with mGRE configuration

```
interface tunnel 0
    ip address 141.11.10.1 255.255.255.0
    tunnel source e0
    tunnel mode gre multipoint
```

30.1. Phases

30.1.1. Phase 1

- Hub and spoke functionality
- for simplified and smaller configuration
- zero touch provisioning for adding spokes to the VPN
- mGRE on hub, GRE on the spokes
- supports dynamically addressed CPEs

30.1.2. Phase 2

- Spoke-to-spoke functionality (by changing the routing protocol behavior self-hop ?)
 - mGRE on hub and spokes
- on demand spoke-to-spoke tunnels avoids dual encrypts/decrypts
- Smaller spoke CPE can participate in the virtual mesh
- restrictions:
 - no summarization allowed on the hub
 - no default routing allowed on the hub
 - spokes must always maintain next-hop reachability

30.1.3. Phase 3

- Architecture and scaling
- uses ip nhrp redirect on the hub
 - sends NHRP Traffic Indication messages
- uses ip nhrp shortcut on the spokes to accept and install the shortcut route
 - accepts NHRP Traffic Indication messages
 - sends NHRP Resolution Request message for the original packet's destination

- install the destination network discovered through the Resolution Reply into the RIB with the responding spoke router as a next hop.

Task: Verify NHRP Configuration

```
# sh ip nhrp

141.11.10.2/32 via 141.11.10.2
  Tunnel1234 created 00:02:21, expire 00:57:38
  Type: dynamic, Flags: unique registered
  NBMA address: 10.11.10.2
141.11.10.3/32 via 141.11.10.3
  Tunnel1234 created 00:02:09, expire 00:57:50
  Type: dynamic, Flags: unique registered
  NBMA address: 10.11.10.3
```

30.2. DMVPN with IPsec using pre-shared key

The feature works according to the following rules.

- Each spoke has a permanent IPSec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server.
- When a spoke needs to send a packet to a destination (private) subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke.
- After the originating spoke learns the peer address of the target spoke, it can initiate a dynamic IPSec tunnel to the target spoke.
- The spoke-to-spoke tunnel is built over the multipoint GRE (mGRE) interface.
- The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets are able to bypass the hub and use the spoke-to-spoke tunnel.
- If an IP multicast stream originates from a spoke location, a rendezvous point (RP) must be deployed at the hub site in order for other spoke site clients to receive the stream
- mGRE Tunnel Interface allows a single GRE interface to support multiple IPSec tunnels and simplifies the size and complexity of the configuration.

TODO: Work this long configuration sample. TODO: Understand the purpose of the individual commands

Hub Router

```
crypto isakmp policy 100
hash md5
authentication pre-share

!--- Add dynamic pre-shared keys for all the remote VPN
!--- routers.
crypto isakmp key cciein8weeks address 0.0.0.0 0.0.0.0
!--- Create the Phase 2 policy for actual data encryption.
```

```

crypto ipsec transform-set strong esp-3des esp-md5-hmac
!
!--- Create an IPSec profile to be applied dynamically to the
!--- GRE over IPSec tunnels.

crypto ipsec profile cciein8weeks
set security-association lifetime seconds 120
set transform-set strong

!--- Create a GRE tunnel template which will be applied to
!--- all the dynamically created GRE tunnels.

interface Tunnel0
ip address 192.168.1.1 255.255.255.0
no ip redirects
ip mtu 1440
ip nhrp authentication cciein8weeks
ip nhrp map multicast dynamic
ip nhrp network-id 1
no ip split-horizon eigrp 90
no ip next-hop-self eigrp 90
tunnel source FastEthernet0/0
tunnel mode gre multipoint

tunnel key 0
tunnel protection ipsec profile cciein8weeks

!--- This is the outbound interface.

interface FastEthernet0/0
ip address 209.168.202.225 255.255.255.0
duplex auto
speed auto

!--- This is the inbound interface.

interface FastEthernet0/1
ip address 1.1.1.1 255.255.255.0
duplex auto
speed auto
!
!--- Enable a routing protocol to send and receive
!--- dynamic updates about the private networks.

router eigrp 10
network 1.1.1.0 0.0.0.255

```

```
network 192.168.1.0
no auto-summary
```

Spoke 1 (DMVPN Phase II)

```
crypto isakmp policy 10
hash md5
authentication pre-share
```

```
!--- Add dynamic pre-shared keys for all the remote VPN
!--- routers and the hub router.
```

```
crypto isakmp key cciein8weeks address 0.0.0.0 0.0.0.0
!
!--- Create the Phase 2 policy for actual data encryption.
crypto ipsec transform-set strong esp-3des esp-md5-hmac
```

```
!--- Create an IPSec profile to be applied dynamically to
!--- the GRE over IPSec tunnels.
```

```
crypto ipsec profile cciein8weeks
set security-association lifetime seconds 120
set transform-set strong
```

```
!--- Create a GRE tunnel template to be applied to
!--- all the dynamically created GRE tunnels.
```

```
interface Tunnel0
ip address 192.168.1.2 255.255.255.0
no ip redirects
ip mtu 1440
ip nhrp authentication cciein8weeks
ip nhrp map multicast dynamic
ip nhrp map 192.168.1.1 209.168.202.225
ip nhrp map multicast 209.168.202.225
ip nhrp network-id 1
ip nhrp nhs 192.168.1.1
tunnel source FastEthernet0/0
tunnel mode gre multipoint <- facilitates spoke to spoke communication
tunnel key 0
tunnel protection ipsec profile cciein8weeks
!
!--- This is the outbound interface.
interface FastEthernet0/0
ip address 209.168.202.131 255.255.255.0
duplex auto
speed auto
!
!--- This is the inbound interface.
interface FastEthernet0/1
ip address 2.2.2.2 255.255.255.0
```

```

duplex auto
speed auto

!--- Enable a routing protocol to send and receive
!--- dynamic updates about the private networks.

router eigrp 10
network 2.2.2.0 0.0.0.255
network 192.168.1.0
no auto-summary

```

Spoke 2

```

crypto isakmp policy 10
hash md5
authentication pre-share

!--- Add dynamic pre-shared keys for all the remote VPN
!--- routers and the hub router.

crypto isakmp key cciein8weeks address 0.0.0.0 0.0.0.0
!--- Create the Phase 2 policy for actual data encryption.
crypto ipsec transform-set strong esp-3des esp-md5-hmac

!--- Create an IPSec profile to be applied dynamically to
!--- the GRE over IPSec tunnels.

crypto ipsec profile cciein8weeks
set security-association lifetime seconds 120
set transform-set strong
!--- Create a GRE tunnel template to be applied to
!--- all the dynamically created GRE tunnels.

interface Tunnel0
ip address 192.168.1.3 255.255.255.0
no ip redirects
ip mtu 1440
ip nhrp authentication cciein8weeks
ip nhrp map multicast dynamic
ip nhrp map 192.168.1.1 209.168.202.225
ip nhrp map multicast 209.168.202.225
ip nhrp network-id 1
ip nhrp nhs 192.168.1.1
tunnel source FastEthernet0/0
tunnel mode gre multipoint
tunnel key 0
tunnel protection ipsec profile cciein8weeks
!

!--- This is the outbound interface.
interface FastEthernet0/0

```

```

ip address 209.168.202.130 255.255.255.0
duplex auto
speed auto
!
!--- This is the inbound interface.
interface FastEthernet0/1
ip address 3.3.3.3 255.255.255.0
duplex auto
speed auto
!
!--- Enable a routing protocol to send and receive
!--- dynamic updates about the private networks.

router eigrp 10
network 3.3.3.0 0.0.0.255
network 192.168.1.0
no auto-summary

```

30.3. QoS profile

The Per-Tunnel QoS for DMVPN feature introduces per-tunnel quality of service (QoS) support for Dynamic Multipoint VPN (DMVPN) and increases per-tunnel QoS performance for Internet Protocol Security (IPsec) tunnel interfaces. This feature allows you to apply a QoS policy on a DMVPN hub on a tunnel instance (per-endpoint or per-spoke basis) in the egress direction for DMVPN hub-to-spoke tunnels. The QoS policy on a DMVPN hub on a tunnel instance allows you to shape the tunnel traffic to individual spokes (parent policy) and to differentiate individual data flows going through the tunnel for policing (child policy).

The QoS policy that is used by the hub for a particular endpoint or spoke is selected by the Next Hop Resolution Protocol (NHRP) group in which the spoke is configured. Even though many spokes may be configured in the same NHRP group, the tunnel traffic of each spoke is measured individually for shaping and policing.

The following example shows how to map NHRP groups to a QoS policy on the hub. The example shows a hierarchical QoS policy (parent: group1_parent/group2_parent; child: group1/group2) that will be used for configuring per-tunnel QoS for DMVPN feature. The example also shows how to map the NHRP group spoke_group1 to the QoS policy group1_parent and map the NHRP group spoke_group2 to the QoS policy group2_parent on the hub:

```
class-map match-all group1_Routing
match ip precedence 6
class-map match-all group2_Routing
match ip precedence 6
class-map match-all group2_voice
match access-group 100
class-map match-all group1_voice
match access-group 100

policy-map group1
class group1_voice
    priority 1000
class group1_Routing
    bandwidth percent 20
policy-map group1_parent
class class-default
    shape average 3000000
service-policy group1
policy-map group2
class group2_voice
    priority percent 20
class group2_Routing
    bandwidth percent 10
policy-map group2_parent
class class-default
    shape average 2000000
service-policy group2

interface tunnel 1
ip address 209.165.200.225 255.255.255.224
no ip redirects
ip mtu 1400
ip nhrp authentication testing
ip nhrp map multicast dynamic
ip nhrp map group spoke_group1 service-policy output group1_parent
ip nhrp map group spoke_group2 service-policy output group2_parent
ip nhrp network-id 172176366
ip nhrp holdtime 300
ip nhrp registration no-unique
tunnel source fastethernet 2/1/1
tunnel mode gre multipoint
tunnel protection ipsec profile DMVPN

interface fastethernet 2/1/1
ip address 209.165.200.226 255.255.255.224
```

30.4. QoS Pre-classify

Configure qos pre-classify in VPN designs where both QoS and IPsec occur on the same system and QoS needs to match on parameters in the cleartext packet other than the DSCP/ToS byte.

Further Reading

Chapter 31. IPSEC

31.1. IPSEC with pre-shared key

31.1.1. IPv4 site to IPv4 site

- IPsec between two sites such as branch and a headquarter is known as site to site or LAN to LAN tunnel.
- can be configured with or without GRE.
- IKE has two modes of operation, aggressive or main mode. Main mode hides IKE/IPSec peer identities.

Router B

```
crypto isakmp policy 2
authentication pre-share

crypto isakmp key cciein8weeks address 172.16.1.1
!
!
!--- Configuration for IPsec policies.
!--- Enables the crypto transform configuration mode,
!--- where you can specify the transform sets that are used
!--- during an IPsec negotiation.

crypto ipsec transform-set Router-IPSEC esp-des esp-sha-hmac
!
!--- Indicates that IKE is used to establish
!--- the IPsec Security Association for protecting the
!--- traffic specified by this crypto map entry.

crypto map cciein8weeks 1 ipsec-isakmp
description Tunnel to172.16.1.1

!--- Sets the IP address of the remote end.

set peer 172.16.1.1

!--- Configures IPsec to use the transform-set
!--- "Router-IPSEC" defined earlier in this configuration.
set transform-set Router-IPSEC

!--- Specifies the interesting traffic to be encrypted.

match address 100
!
!--- Configures the interface to use the
!--- crypto map " cciein8weeks" for IPsec.

interface FastEthernet0
ip address 172.17.1.1 255.255.255.0
duplex auto
speed auto
crypto map cciein8weeks
```

IPSec(validate_transform_proposal): proxy identities not supported ISAKMP: IPSec policy invalidated proposal ISAKMP (0:2): SA not acceptable!

Above messages are indicative of the fact that access lists (as referenced in match address command inside a crypto map) for IPsec interesting traffic do not match between peers.

Further Reading <http://goo.gl/PmO6l4>

31.1.2. IPv6 in IPv4 tunnels

Generic routing encapsulation (GRE) tunnels sometimes are combined with IPsec, because IPsec does not support IPv6 multicast packets. This function prevents dynamic routing protocols from running successfully over an IPsec VPN network. Because GRE tunnels do support IPv6 multicast, a dynamic routing protocol can be run over a GRE tunnel. Once a dynamic routing protocol is configured over a GRE tunnel, you can encrypt the GRE IPv6 multicast packets using IPsec.

IPsec can encrypt GRE packets using a crypto map or tunnel protection. Both methods specify that IPsec encryption is performed after GRE encapsulation is configured. When a crypto map is used, encryption is applied to the outbound physical interfaces for the GRE tunnel packets. When tunnel protection is used, encryption is configured on the GRE tunnel interface.

“%CRYPTO-4-IKMP_BAD_MESSAGE: IKE” message from 150.150.150.1 failed its sanity check or is malformed appears if the pre-shared keys on the peers do not match. In order to fix this issue, check the pre-shared keys on both sides.

Further Reading <http://goo.gl/pqy0E8> <http://goo.gl/RhXJDZ>

31.1.3. VTI

Virtual Tunneling Interface

The use of IPsec VTIs both greatly simplifies the configuration process when you need to provide protection for remote access and provides a simpler alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation and crypto maps with IPsec. A major benefit associated with IPsec VTIs is that the configuration does not require a static mapping of IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel. The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration compared to the more complex process of using access control lists (ACLs) with the crypto map in native IPsec configurations. VTIs function like any other real interface so that you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface. When IPsec VTIs are used, you can separate the application of features such as NAT, ACLs, and QoS and apply them to clear-text or encrypted text, or both. When crypto maps are used, there is no simple way to apply encryption features to the IPsec tunnel.

There are two types of VTI interfaces: - Static VTIs (SVTIs) - Dynamic VTIs (DVTIs)

SVTIs configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites. The advantage of using SVTIs as opposed to crypto map configurations is

that users can enable dynamic routing protocols on the tunnel interface without the extra 4 bytes required for GRE headers, thus reducing the bandwidth for sending encrypted data. Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels. Dynamic VTIs can be used for both the server and remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs. Dynamic VTIs function like any other real interface so that you can apply QoS, firewall, other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

Further Reading <http://goo.gl/0yHakK>

31.2. GET VPN

Group Encrypted Transport

The IOS GETVPN is a tunnel-less (i.e. no overlay) VPN technology that provides end-to-end security for network traffic in a native mode and maintaining the fully meshed topology. It uses the core network's ability to route and replicate the packets between various sites within the enterprise. Cisco IOS GETVPN preserves the original source and destination IP addresses information in the header of the encrypted packet for optimal routing. Hence, it is largely suited for an enterprise running over a private Multiprotocol Label Switching (MPLS)/IP-based core network. It is also better suited to encrypt multicast traffic. Cisco IOS GET VPN uses Group Domain of Interpretation (GDOI) as the keying protocol and IPSec for encryption. A GETVPN deployment has primarily three components, Key Server (KS), Group Member (GM), and Group Domain of Interpretation (GDOI) protocol. GMs do encrypt/decrypt the traffic and KS distribute the encryption key to all the group members. The KS decides on one single data encryption key for a given lifetime. Since all GMs use the same key, any GM can decrypt the traffic encrypted by any other GM. GDOI protocol is used between the GM and KS for group key and group SA management. Minimum one KS is required for a GETVPN deployment. Unlike traditional IPSec encryption solutions, GET VPN uses the concept of group SA. All members in the GETVPN group can communicate with each other using a common encryption policy and a shared SA and therefore no need to negotiate IPSec between GMs on a peer to peer basis; thereby reducing the resource load on the GM routers.

31.2.1. Group Member

The group member registers with the key server to get the IPSec SA that is necessary to encrypt data traffic within the group. The group member provides the group ID to the key server to get the respective policy and keys for this group. These keys are refreshed periodically by KS, and before the current IPSec SAs expire, so that there is no loss of traffic.

31.2.2. Key Server

Key server is responsible for maintaining security policies, authenticating the GMs and providing the session key for encrypting traffic. KS authenticates the individual GMs at the time of registration. Only after successful registration the GMs can participate in group SA. A group member can register at any time and receive the most current policy and keys. When a GM registers with the key server, the key server verifies the group id number of the GM. If this id number is a valid and the GM has provided valid Internet Key Exchange (IKE) credentials, the key server sends the SA policy and the Keys to the group member.

There are two types of keys that the GM will receive from the KS: - Key Encryption Key (KEK), for securing control plane - Traffic Encryption Key (TEK), for securing data plane The TEK becomes part of the IPSec SA with which the group members within the same group encrypt the data. KEK is used to secure rekey messages (i.e. control plane) between the key server and the group members.

The Key Server sends out rekey messages either because of an impending IPSec SA expiration or because the security policy has changed on the key server. Keys can be distributed during rekey using either multicast or unicast transport. Multicast method is more scalable as keys need not be transmitted to each group member individually. Unlike in unicast, KS will not receive acknowledgement from GM about the success of the rekey reception in multicast rekey method. In unicast rekey method, KS will delete a GM from its database if three consecutive rekeys are not acknowledged by that particular GM.

Further Reading <http://goo.gl/mxG401>

Chapter 32. L2 VPN

- simplest solution where a client wants to manage his own routing protocols, IP management, and QoS mechanisms; this means that the service provider only focuses on providing high-throughput Layer 2 connections.

32.1. Pseudo-Wire

- Ethernet Pseudo-Wire
- allows Ethernet frames to traverse an operational MPLS cloud.
 - This implies that spanning tree will run on the links, and devices connected through these connections will use the same subnet.
 - This service has its own field name, Emulated Service, and operates over pseudowire.
 - In addition to this, it is necessary to have an operational packet label switched network (MPLS network).

Two modes: raw mode and tagged mode.

tagged mode

- uses the same 802.1Q tag at each end of the link.
- uses pseudowire type 0x0004.
- uses service-delimiting VLAN tag
 - Every frame sent on the PW must have a different VLAN for each customer.
 - If a frame as received by a PE from the attachment circuit is missing a service-delimiting VLAN tag, the PE must prepend the frame with a dummy VLAN tag before sending the frame on the PW.

Raw Mode

- the service-delimiting tag might or might not be added in the frame and has no significance to the end points.
- pseudowire type 0x0005.
- service-delimiting tags are never through the attachment circuit by the PE; it is mandatory that they are stripped from the frame before the frame is transmitted.

32.2. L2TPv3

- provides several enhancements to L2TP to tunnel any Layer 2 payload over L2TP, by defining how the L2TP protocol tunnels Layer 2 payloads over an IP core network by using Layer 2 VPNs.
- RFC 3931, 4719
- uses protocol ID 115
- A L2TPv3 tunnel is a control connection between two PE routers.

- One L2TPv3 tunnel can have multiple data connections, and each data connection is termed as an L2TPv3 session.
- The control connection is used to establish, maintain, and release sessions.
- Each session is identified by a session ID which is unique across the entire router.
- L2TPv3 carries frames inside IP packets.
- Pre-requisites on Cisco routers
 - cef or dcef
 - loopback address reachable from the remote PE device at the other end of the L2TPv3 control channel

Further Reading <http://goo.gl/V9egil>

32.3. ATOM

- Any Transport over MPLS (AToM) transports Layer 2 packets over a MPLS backbone.
- AToM uses a directed LDP session between edge routers for setting up and maintaining connections. Forwarding occurs through the use of two levels of labels, switching between the edge routers.
- The external label (tunnel label) routes the packet over the MPLS backbone to the egress Provider Edge (PE) at the ingress PE. The VC label is a de-multiplexing label that determines the connection at the tunnel endpoint (the particular egress interface on the egress PE as well as the virtual path identifier [VPI]/virtual channel identifier [VCI] value for an ATM Adaptation Layer 5 [AAL5] protocol data unit [PDU], the data-link connection identifier [DLCI] value for a Frame Relay PDU, or the virtual LAN [VLAN] identifier for an Ethernet frame). EoMPLS carries frames inside MPLS packets.

Because the control word is included by default, so it may be necessary to explicitly disable this command in static pseudowire configurations. You can use **mpls control-word** command is used in static pseudowire configurations, the command must be configured the same way on both ends of the connection to work correctly. Otherwise, the provider edge routers cannot exchange control messages to negotiate inclusion or exclusion of the control word.

```
!First we will create the xconnect configuration on routers.
R2(config)# int f0/0
R2(config-if)# xconnect 4.4.4.4 204 encapsulation mpls
R2(config-if-xconn)# end
```

```
!Now we do the matching configuration on the other device.
R4(config)# int f0/0
R4(config-if)# xconnect 2.2.2.2 204 encapsulation mpls
R4(config-if-xconn)# end
```

The xconnect command used on the F0/0 interface on both routers is used to create a bridged connection with the destination specified. The command is broken down with an xconnect keyword

followed by the peering router address and the unique virtual circuit ID (VCID). The VCID must match on both ends of the connection. The encapsulation can be L2TPv2, L2TPv3, or MPLS. Never lose sight of the fact that there must be a unique address per router for each xconnect.

Task: Verify the status of a pseudo-wire

```
R2# show xconnect all
```

Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State

UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware

XC ST Segment 1 S1 Segment 2 S2

```
-----+-----+-----+-----+  
DN ac Fa0/0(Ethernet) AD mpls 4.4.4.4:204 DN
```

An MPLS Layer 2 pseudowire has two segments. Segment 1 (S1) is for the customer-facing port. Segment 2 (S2) relates to the core configuration.

To specify the path that traffic uses a MPLS Traffic engineering (TE) tunnel or destination IP address and Domain Name Server (DNS) name, use the preferred-path command in pseudowire configuration mode.

To disable tunnel selection, use the no form of this command.

```
preferred-path {interface tunnel tunnel-number | peer {ip-address | host-name}} [disable-fallback]  
no preferred-path {interface tunnel tunnel-number | peer {ip-address | host-name}} [disable-fallback]
```

Further Reading <http://goo.gl/6RsX89>

32.4. VPLS

TODO clean the notes

- Virtual Private LAN Service
- offers Layer 2 Ethernet services.
- enables geographically separate LAN segments to be interconnected as a single bridged domain over an MPLS network.
 - VPLS over GRE enables VPLS across an IP network.
 - The provider edge (PE) routers for VPLS over GRE must support VPLS and additional GRE encapsulation and decapsulation.

An instance of VPLS must be configured on each PE router.

- provides multipoint Ethernet service as compared to Ethernet over MPLS (EoMPLS) that is point to point.
- emulates a virtual IEEE Ethernet bridge network.
- uses flooding to communicate MAC address reachability information.

- VPLS can carry single VLAN within each instance.
- supports MAC address aging and replicates broadcast and multicast traffic.
- A point to point Ethernet Virtual Circuit (EVC) connecting a pair of physical UNIs is also known as Ethernet Wire Service (EWS) or Ethernet Private Line (EPL).
 - EPL provides VLAN transparency and control protocol tunneling are supplied by the implementation of 802.1Q-in-Q tag-stacking technology.
- Ethernet Virtual Private Line (EVPL) and EPL are also considered E-Line services.

Unlike Layer 3 VPN, there is no routing interaction between customer and service provider networks.

- Multipoint-to-multipoint configuration
- Forwarding of frames based on learned MAC addresses
- Uses virtual forwarding instance (VFI, like VLAN) for customer separation

VPLS Components:

- User-facing PE (U-PE): The U-PE is the device to which the functions needed to take forwarding or switching decisions at the ingress of the provider network.
- Network PE (N-PE): The N-PE is the device to which the signaling and control functions are allocated when a VPLS-PE is distributed across more than one box.
- Virtual switching instance (VSI): Virtual switching instance that serves one single VPLS. A VSI performs standard LAN (that is, Ethernet) bridging functions, including forwarding done by a VSI based on MAC addresses and VLAN tags.
- Pseudowire (PW): PWE3 is a mechanism that emulates the essential attributes of a telecommunications service (such as a T1 leased line or Frame Relay) over a PSN.
- Attachment circuit (AC): The physical or virtual circuit attaching (AC) a CE to a PE. An attachment circuit may be, for example, a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, or an MPLS LSP. One or multiple ACs can belong to same VFI.
- VC (virtual circuit): Martini-based data encapsulation, tunnel label is used to reach remote PE, VC label is used to identify VFI. One or multiple VCs can belong to same VFI Virtual Forwarding Instance (VFI):
- VFI creates L2 multipoint bridging among all ACs and VCs. It's an L2 broadcast domain such as VLAN.
- Multiple VFIs can exist on the same PE box to separate user traffic such as VLANs.
- Signaling

Signaling uses LDP to establish and tear down PWs. Using LDP as the signaling VPLS control plane does not have inherent support of auto-discovery. Therefore, LDP-VPLS relies on manual configuration to identify all PE routers. MPLS in the core, normal LDP sessions per hop to exchange tunnel label or IGP label. Targeted or directed LDP session between PEs to exchange VC label. Tunnel label is used to forward packet from PE to PE VC label and is used to identify L2VPN circuit.

Further Reading <http://goo.gl/KwPVFS>

32.5. OTV

- Overlay Transport Virtualization
- OTV is a “MAC address in or over IP” technique for supporting Layer 2 VPNs to extend LANs over any transport. The transport can be Layer 2 based, Layer 3 based, IP switched, label switched, or anything else as long as it can carry IP packets. By using the principles of MAC routing, OTV provides an overlay that enables Layer 2 connectivity between separate Layer 2 domains while keeping these domains independent and preserving the fault-isolation, resiliency, and load-balancing benefits of an IP-based interconnection.

The core principles on which OTV operates are the use of a control protocol to advertise MAC address reachability information (instead of using data plane learning) and packet switching of IP encapsulated Layer 2 traffic (instead of using circuit switching) for data forwarding. These features are a significant departure from the core mechanics of traditional Layer 2 VPNs. In traditional Layer 2 VPNs, a static mesh of circuits is maintained among all devices in the VPN to enable flooding of traffic and source-based learning of MAC addresses. This full mesh of circuits is an unrestricted flood domain on which all traffic is forwarded. Maintaining this full mesh of circuits severely limits the scalability of existing Layer 2 VPN approaches. At the same time, the lack of a control plane limits the extensibility of current Layer 2 VPN solutions to properly address the requirements for extending LANs across data centers.

OTV uses a control protocol to map MAC address destinations to IP next hops that are reachable through the network core. OTV can be thought of as MAC routing in which the destination is a MAC address, the next hop is an IP address, and traffic is encapsulated in IP so it can simply be carried to its MAC routing next hop over the core IP network. Thus a flow between source and destination host MAC addresses is translated in the overlay into an IP flow between the source and destination IP addresses of the relevant edge devices. This process is called encapsulation rather than tunneling as the encapsulation is imposed dynamically and tunnels are not maintained. Since traffic is IP forwarded, OTV is as efficient as the core IP network and will deliver optimal traffic load balancing, multicast traffic replication, and fast failover just like the core would. OTV also supports detection of multi-homing.

A technology typically deployed at the customer edge (CE), unlike VPLS, OTV is configured on each CE router or switch. OTV provides Layer 2 LAN extension over Layer 3-, Layer 2-, or MPLS-based networks. One of the significant benefits or advantages of OTV is the fault-domain isolation feature; thus spanning-tree root does not change. With each CE having its own root, there is no intervention or planning required by the provider. OTV supports automatic detection of multihoming and ARP optimization.

OTV entities/roles and their description

edge device

This is a device which performs all OTV functions. The OTV Edge device is connected to Layer 2 segments and IP transport network.

join interfaces

These are Layer 3 interfaces on the OTV Edge device which connects to the IP transport network

internal interface

These are Layer 2 interfaces on the OTV Edge device. These can be "trunk" or "access" ports.

overlay interface

This is a multicast-enabled multi-access network over which all OTV encapsulated Layer 2 frames are carried.

site VLAN

OTV Edge devices need to elect an Authoritative Edge Device (AED) per VLAN so that only one device forwards traffic for that VLAN. For this election, the OTV Edge devices use Site VLAN for communication on the local site.

authoritative edge device

The authoritative edge device is responsible for all MAC address reachability updates for a VLAN.

Further Reading <http://goo.gl/XioB96>

Part IV : Infrastructure Security

Chapter 33. AAA

33.1. Local AAA Server

The Local AAA Server feature allows you to configure your router so that user authentication and authorization attributes currently available on AAA servers are available locally on the router. The attributes can be added to existing framework, such as the local user database or subscriber profile. The local AAA server provides access to the complete dictionary of Cisco IOS supported attributes.

You can configure your router so that AAA authentication and authorization attributes currently available on AAA servers are made available on existing Cisco IOS devices. The attributes can be added to existing framework, such as the local user database or subscriber profile. For example, an attribute list can now be added to an existing username, providing the ability for the local user database to act as a local AAA server. For situations in which the local username list is relatively small, this flexibility allows you to provide complete user authentication or authorization locally within the Cisco IOS software without having a AAA server. This ability can allow you to maintain your user database locally or provide a failover local mechanism without having to sacrifice policy options when defining local users. A subscriber profile allows domain-based clients to have policy applied at the end-user service level. This flexibility allows common policy to be set for all users under a domain in one place and applied there whether or not user authorization is done locally.

Further Reading <http://goo.gl/aaTqf5>

33.2. PPP Security

TODO

- implement device access control
 - lines
 - password encryption
 - management plane protection

Chapter 34. CoPP

Chapter 35. CPP

Control Plane Protection - security feature that extends the policing functionality provided by the software-based Control Plane Policing (CoPP) feature. The CoPP feature controls the rate in which control plane traffic is sent to the Route Processor in Cisco IOS software-based devices. Control Plane Protection extends this policing functionality by dividing the Control Plane into three control plane sub-interfaces and allowing the enforcement of separate rate-limiting policies. In addition, CPP incorporates port-filtering and queue-thresholding. Port-filtering is a mechanism for the early dropping of packets that are directed to closed or non-listened IOS TCP/UDP ports. Queue-thresholding is a mechanism that limits the number of packets per protocol held in the control-plane input queue, preventing the input queue from being overwhelmed by any single protocol traffic.

- The three control plane sub-interfaces implemented by Control Plane Protection are:
 - Control-plane host subinterface—This interface handles all control-plane IP packets that are destined to any of the IP addresses configured on the router interfaces. Examples of traffic falling in this category include tunnel termination traffic, management traffic or routing protocols such as SSH, SNMP, BGP, OSPF, and EIGRP. All host traffic terminates on and is processed by the router.
 - Control-plane transit subinterface—This subinterface receives all IP packets that are software switched by the route processor. This means packets that are not directly destined to the router itself but rather traffic traversing through the router and that require process switching.
 - Control-plane CEF-exception subinterface—This control-plane subinterface receives all IP packets that are either redirected as a result of a configured input feature in the CEF packet forwarding path for process switching or directly enqueued in the control plane input queue by the interface driver (i.e. ARP, L2 Keepalives and all non-IP host traffic).
- In addition, CPP enhances the protection of the control-plane host subinterface by implementing Port-filtering and Queue-thresholding. Port-filtering is a feature that can only be applied to the control-plane host subinterface and that automatically drops packets directed toward closed or non-listened UDP/TCP ports on the router. Queue-thresholding is another feature that can only be applied to the control-plane host subinterface and that limits the number of unprocessed packets per protocol, preventing the input queue from being overwhelmed by any single protocol traffic.
- At a very high level the sequence of events with Control Plane Protection is as follows:
 - step 1 A packet enters the router configured with CoPP on an ingress interface.
 - step 2 The interface performs the basic input port and QoS services.
 - step 3 The packet gets forwarded to the router processor.
 - step 4 The router processor makes a routing decision, determining whether or not the packet is destined to the control plane.
 - step 5 Packets destined for the control plane are processed by Aggregate CoPP, and are dropped or forward to the Control Plane Path according to the policies for each traffic class. Packets that have other destinations are forwarded normally.

- step 6 Packets sent to the Control Plane Path are intercepted by the Control Plane Protection traffic classifier, which classifies the packets into the corresponding control-plane subinterfaces.
- step 7 Packets received by each control-plane subinterface are dropped or forward to the Control Plane global input queue according to the configured policies.
- step 8 In addition, packets sent to the control-plane host subinterface can be dropped or forwarded according to the Port-filter and Queue-thresholding policies before they are sent to the global input queue.

Similar to CoPP, CPP helps protect the RP of Cisco IOS software-based routers by filtering unwanted traffic and by rate-limiting the traffic expected by the control plane. This shields the control plane from traffic that might be part of DoS or other attacks, helping maintain network stability even during attack conditions.

CPP ability to divide the control plane traffic and rate-limit each traffic type individually, gives you greater traffic control for attack mitigation. Port-filtering and Queue-thresholding also provide for a more advanced attack protection. On one hand, Port-filtering shields the RP from packets directed to closed or non-listened TCP/UDP ports, mitigating attacks attempting to spoof legitimate traffic permitted by CoPP. On the other hand, Queue-thresholding limits protocol queue usage mitigating attacks designed to overwhelm the input queue with the flooding of a single protocol.

CPP is recommended on all software-based IOS platforms, where hardware-based CoPP is not available. CPP is particularly useful on routers facing the Internet or other external networks.

Chapter 36. Management Plane Protection

- restrict the interfaces on which network management packets are allowed to enter a device.
- disabled by default
- After MPP is enabled, no interfaces except designated management interfaces will accept network management traffic destined to the device.
- Restricting management packets to designated interfaces provides greater control over management of a device, providing more security for that device. Other benefits include improved performance for data packets on nonmanagement interfaces, support for network scalability, need for fewer access control lists (ACLs) to restrict access to a device, and management packet floods on switching and routing interfaces are prevented from reaching the CPU.
- management protocols: beep, (t)ftp, http(s), ssh v1 and v2, telnet, snmp v1,v2 and v3
- pre-requisites: cef
- interfaces not supported:
 - Out-of-band management interfaces (also called dedicated management interfaces)
 - Loopback and virtual interfaces not associated to physical interfaces
 - Fallback and standby management interfaces
 - Hardware-switched and distributed platforms
- restrictions
 - Secure Copy (SCP) is supported under the Secure Shell (SSH) Protocol and not directly configurable in the command-line interface (CLI).
 - Uninformed management stations lose access to the router through nondesignated management interfaces when the Management Plane Protection feature is enabled.

Task:

```
conf t  
control-plane  
management-interface <type slot/port> allows <protocols>
```

Task: Displays information about the management interface such as type of interface, protocols enabled on the interface, and number of packets dropped and processed.

```
# show management-interface [<interface> | protocol <protocol-name>]
```

Chapter 37. Access Control List

37.1. References

sec-data-acl

37.2. Reflexive ACL

- source control

Chapter 38. IP Source Guard

TODO Improve

IP Source Guard will inspect traffic from hosts to be certain they're not spoofing their IP address or, optionally, MAC address. As mentioned, it knows the MAC and IP information from the DHCP snooping database. Enabling the feature is simple:

```
SW1(config-if)#ip verify source ! check IP addresses  
SW1(config-if)#ip verify source port-security ! check IPs & MACs
```

If you don't want to rely strictly on the DHCP snooping database, you can make manual entries:

```
SW1(config)#ip source binding mac-address vlan vlan ip-address interface interface
```

Of important note, source guard can be used in a non-DHCP environment using the above command to populate the reference table, however, you must still enable dhcp snooping globally and on the specific VLAN in order for source guard to function!



GNS3 doesn't support this feature

Chapter 39. Dynamic ARP Inspection

Dynamic ARP inspection blocks "a variety of ARP-based attacks", according to the documentation. Drop ARPs that shouldn't be coming based on the information seen in the DHCP snooping table.

```
SW1(config-if)#ip arp inspection trust ! need to trust other switches  
SW1(config)#ip arp inspection vlan 100 ! turn the feature on globally for the VLAN
```

- port-security

39.1. Router Security

- ipv4 access list
- ipv6 traffic filter
- unicast reverse path forwarding
- ipv6 first hop security
 - ra guard
 - dhcp guard
 - binding table
 - device tracking
 - ND inspection/snooping
 - source guard
 - PACL

Chapter 40. IEEE 802.1X

40.1. Definition

- Port-based authentication
- until the client is authenticated, allows only EAPoL (Extensible Authentication Protocol over LAN), CDP and STP traffic
- supplicant: client workstation running 802.1x compliant software
- authenticator: edge switch or wireless access point
- authentication server: performs the actual authentication (Radius with EAP)

40.2. Port Security

TODO Redo this part

```
DOT1X-SP-5-SECURITY_VIOLATION: Security violation on interface GigabitEthernet4/8,  
New MAC address 0080.ad00.c2e4 is seen on the interface in Single host mode  
%PM-SP-4-ERR_DISABLE: security-violation error detected on Gi4/8, putting Gi4/8 in  
err-disable state
```

This message indicates that the port on the specified interface is configured in single-host mode. Any new host that is detected on the interface is treated as a security violation. The port has been error disabled. Ensure that only one host is connected to the port. If you need to connect to an IP phone and a host behind it, configure Multidomain Authentication Mode on that switchport.

The Multidomain authentication (MDA) mode allows an IP phone and a single host behind the IP phone to authenticate independently, with 802.1X, MAC authentication bypass (MAB), or (for the host only) web-based authentication. In this application, Multidomain refers to two domains — data and voice — and only two MAC addresses are allowed per port. The switch can place the host in the data VLAN and the IP phone in the voice VLAN, though they appear to be on the same switch port. The data VLAN assignment can be obtained from the vendor-specific attributes (VSAs) received from the AAA server within authentication.

Part V : Infrastructure Services

Chapter 41. System Management

Chapter 42. Switch Administration

42.1. Interface Characteristics

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swint.html

42.2. System Clock

- Can be set manually or dynamic with NTP
- Keeps track internally based on UTC

Task: Configure the Time Zone

```
(config)# clock timezone <zone> <hours-offset> [<minutes-offset>]
```



Use the minutes-offset variable when the local time zone is a percentage of an hour different from UTC. For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5, where the 3 means 3 hours and .5 means 50 percent. In this case, the necessary command is **clock timezone AST -3 30**.

Task: Reset the Time to UTC

```
(config)# no clock timezone
```

Task: Set the System Clock Manually

```
(config)# clock set <hh:mm:ss month day year>
```

Task: Display the Time and Date Configuration

```
# sh clock [detail]
```

Task: Configure Daylight Saving Time

```
(config)# clock summer-time <zone> recurring [week day month hh : mm : week day month hh : mm [offset]]
```

Example

```
(config)# clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October 2:00
```

42.3. System Name and Prompt

Task: Configure a System Name

```
(config)# hostname <name>
```

42.4. MOTD Login Banner

- MOTD and login not configured by default

Task: Configure a Message Of the Day Login Banner

```
(config)# banner motd <delimiting-character> <message> <delimiting-character>
```

42.5. Login Banner

- displayed on all connected terminals
- appears after the MOTD banner and before the login prompt

Task: Configure a Login Banner

```
(config)# banner login <delimiting-character> <message> <delimiting-character>
```

42.6. MAC Address Table

- lists the destination MAC address with the associated VLANs , port number, and the type (static or dynamic)
- dynamic address are discarded after the **aging time** (300 seconds by default)

Task: Display Address Table Entries for the Specified MAC Address

```
# sh mac address-table address <MAC>
```

Task: Display Only Dynamic MAC Addresses

```
# sh mac address-table dynamic
```

Task: Display the Number Of Addresses Present

```
# sh mac address-table count
```

Task: Display the MAC Address Table Information for the Specified VLAN

```
# sh mac address-table vlan
```

Task: Display the MAC Address Table Information for the Specified Interface

```
# sh mac address-table interface
```

42.6.1. Aging Time

- Default: 300 seconds

Task: Set the Length Of Time That a Dynamic Entry Remains In the MAC Address After the Entry Is Used or Updated

```
# mac address-table aging-time [0 | 10-1000000] [vlan <1-4094>]
```

Task: Displays the Aging Time

```
# sh mac address-table aging-time [<vlan_id>]
```

Task: Remove Dynamic Address Entries

```
# clear mac address-table dynamic [<mac-address>]
```

42.6.2. MAC Address Change Notification Traps

- send SNMP trap when the switch learns or removes dynamic and secure MAC addresses.
- do not send trap for self addresses, multicast addresses or static addresses
- can set a trap-interval time to bundle the notification traps to reduce network traffic

Task: Send MAC Address Change Notification Traps to an NMS Host

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }  
<community-string> mac-notification  
(config)# snmp-server enable traps mac-notification change  
(config)# mac address-table notification change [ interval <seconds> ] [ history-size  
<i0-1-500> ]  
(config)# interface <interface-id>  
(config-if)# snmp trap mac-notification change {added | removed }
```

Task: Verify the MAC Address Table Notification Change Configuration

```
# sh mac address-table notification change [interface]
```

42.6.3. MAC Address Move Notification Traps

- send a SNMP notification whenever a MAC address moves from one port to another within the same VLAN

Task: Send MAC Address Move Notification Traps to an NMS Host

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }  
<community-string> mac-notification  
(config)# snmp-server enable traps mac-notification move  
(config)# mac address-table notification mac-move
```

Task: Verify the MAC Address Table Notification Move Configuration

```
# sh mac address-table notification mac-move
```

42.6.4. MAC Threshold Notification Traps

- Send an SNMP notification when a MAC Address table threshold limit is reached or exceeded.

Task: Configure MAC Threshold Notifcation Traps

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }  
<community-string> mac-notification  
(config)# snmp-server enable traps mac-notification threshold  
(config)# mac address-table notification threshold ! to enable the feature  
(config)# mac address-table notification threshold [limit <percentage>] | [ interval  
<seconds> ]
```

Task: Verify the MAC Address Table Notification Threshold Configuration

```
# sh mac address-table notification threshold
```

42.6.5. Static Address

- manually entered in the address table and must be manually removed
- can be unicast or mcast
- doesn't age and is retained when the switch restarts
- must be associated with a VLAN and a interface
 - A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned
 - if the VLAN is in a private-primary or private-secondary, configure the same static address in all associated VLANs.

Task: Add a Static Address to the MAC Address Table

```
(config)# mac address-table static <MAC> vlan <vlan-id> interface <interface-id>
```

Task: Display Only Static MAC Addresses

```
# sh mac address-table static
```

42.6.6. Unicast MAC Address Filtering

- Drops packets with specific source or destination MAC addresses
- disabled by default
- mcast, bcast and router MAC addresses are not supported

Task: Enable Unicast MAC Address Filtering

```
(config)# mac address-table static <MAC> vlan <vlan-id> drop
```

42.6.7. MAC Address Learning

- enabled by default on all VLANs
- can be disabled with the following restrictions:
 - If the VLAN has a configured SVI, the switch then floods all IP packets in the Layer 2 domain.
 - If you disable MAC address learning on a VLAN with more than two ports, every packet entering the switch is flooded in that VLAN domain.
 - You cannot disable MAC address learning on a VLAN that is used internally by the switch. If the VLAN ID that you enter is an internal VLAN, the switch generates an error message and rejects the command. To view internal VLANs in use, enter the show vlan internal usage privileged EXEC command.
 - If you disable MAC address learning on a VLAN configured as a private-VLAN primary VLAN, MAC addresses are still learned on the secondary VLAN that belongs to the private VLAN and are then replicated on the primary VLAN. If you disable MAC address learning on the secondary VLAN, but not the primary VLAN of a private VLAN, MAC address learning occurs on the primary VLAN and is replicated on the secondary VLAN.
 - You cannot disable MAC address learning on an RSPAN VLAN. The configuration is not allowed.
 - If you disable MAC address learning on a VLAN that includes a secure port, MAC address learning is not disabled on that port. If you disable port security, the configured MAC address learning state is enabled.



Task: Disable MAC Address Learning on an interface

```
(config)# no mac-address-table learning interface <interface-type slot/port>
```

Task: Disable MAC Address Learning on an range of VLANs

```
(config)# no mac-address-table learning {vlan <vlan-id> [,<vlan-id> | -<vlan-id>]}
```

Task: Display the MAC Address Learning

```
sh mac address-table learning [vlan <vlan-id>]
```

Task: Reenable MAC Address Learning

```
(config)# default mac address-table learning vlan <vlan-id>
```

42.7. Error disable

- port disabled due to error condition
 - no traffic sent or received
 - port LED orange or amber
- eliminates the possibility that this port can cause other ports on the module or the entire module to fail. Such a failure can occur when a bad port monopolizes buffers or port error messages monopolize interprocess communications on the card, which can ultimately cause serious network issues.

Task: Show which local ports are involved in the errdisabled state.

```
# show interfaces status err-disabled
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Gi4/1		err-disabled	100	full	1000	1000BaseSX

42.7.1. Err-Disable detection

- error-disable detection enabled by default

Task: Disable error disable detection

```
(config-if)# no err-disable detect cause
```

Task: Shows the reason for the errdisable status.

```
# show errdisable detect
```

Reasons for the interface to go into errdisable

- Duplex mismatch
- Port channel misconfiguration

- BPDU guard violation
- UDLD condition
- Late-collision detection
- Link-flap detection
- PAgP flap
- Security violation
- L2TP guard
- DHCP snooping rate-limit
- Incorrect GBIC / Small Form-Factor Pluggable module or cable
- ARP inspection
- Inline power

42.7.2. Link-flap detection

- if the interface goes up and down more than five times in 10 seconds.
- common cause: Layer 1 issue such as a bad cable, duplex mismatch, or bad GBIC card.
- console messages or syslog server that state the reason for the port shutdown.

```
%PM-4-ERR_DISABLE: link-flap error detected on Gi4/1, putting Gi4/1 in err-disable state
```

Task: View the flap values:

ErrDisable Reason	Flaps	Time (sec)
pagp-flap	3	30
dtp-flap	3	30
link-flap	5	10

42.7.3. Loopback error

- occurs when the keepalive packet is looped back to the port that sent the keepalive.

```
%PM-4-ERR_DISABLE: loopback error detected on Gi4/1, putting Gi4/1 in err-disable state
```

42.7.4. L2TP guard

When the Layer 2 PDUs enter the tunnel or access port on the inbound edge switch, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 802.1Q tunneling is enabled, packets are also double-tagged. The outer tag is the customer metro tag and the inner tag is the customer VLAN tag. The core switches ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge switches on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel or access ports in the same metro VLAN. Therefore, the Layer 2 PDUs are kept intact and delivered across the service-provider infrastructure to the other side of the customer network.

```
(config)#interface gigabitethernet 0/7  
(config-if)# l2protocol-tunnel {cdp | vtp | stp}
```

The interface goes to errdisabled state. * If an encapsulated PDU (with the proprietary destination MAC address) is received from a tunnel port or access port with Layer 2 tunneling enabled, the tunnel port is shut down to prevent loops. * The port also shuts down when a configured shutdown threshold for the protocol is reached.

You can manually reenable the port (by issuing a shutdown, no shutdown command sequence) or if errdisable recovery is enabled, the operation is retried after a specified time interval.

The interface can be recovered from errdisabled state by reenabling the port using **errdisable recovery cause l2ptguard**.

This command is used to configure the recovery mechanism from a Layer 2 maximum rate error so that the interface can be brought out of the disabled state and allowed to try again. You can also set the time interval. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.

42.7.5. Incorrect GBIC / Small Form-Factor Pluggable module or cable

- Ports go into errdisabled state with the **%PHY-4-SFP_NOT_SUPPORTED** error message when you connect Catalyst 3560 and Catalyst 3750 Switches using an SFP Interconnect Cable.
- The Cisco Catalyst 3560 SFP Interconnect Cable (CAB-SFP-50CM=) provides for a low-cost, point-to-point, Gigabit Ethernet connection between Catalyst 3560 Series Switches. The 50-centimeter cable is an alternative to using SFP transceivers when interconnecting Catalyst 3560 Series Switches through their SFP ports over a short distance. All Cisco Catalyst 3560 Series Switches support the SFP Interconnect Cable. When a Catalyst 3560 Switch is connected to a Catalyst 3750 or any other type of Catalyst switch model, you cannot use the CAB-SFP-50CM= cable.
- You can connect both switches using a copper cable with SFP (GLC-T) on both devices instead of a CAB-SFP-50CM= cable.

42.7.6. Err-Disable recovery

Steps

- Determine the cause with **sh interfaces status err-disabled**
- Fix the root cause
- Reenable the Errdisabled ports manually with **shutdown, no shutdown** command sequence or automatically after a specified amount of time with **errdisable recovery** command.

Task: Shows the time period after which the interfaces are enabled for errdisable conditions.

```
#show errdisable recovery
```

ErrDisable Reason	Timer Status
udld	Enabled
bpduguard	Enabled
security-violation	Enabled
channel-misconfig	Enabled
pagp-flap	Enabled
dtp-flap	Enabled
link-flap	Enabled
l2ptguard	Enabled
psecure-violation	Enabled
gbic-invalid	Enabled
dhcp-rate-limit	Enabled
mac-limit	Enabled
unicast-flood	Enabled
arp-inspection	Enabled

42.8. L2 MTU

- default size: 1500 bytes
- max size: 1998 bytes, 9198 bytes for jumbo frames

Task: Change the MTU size for all Fast Ethernet interfaces on the switch.

```
(config)# system mtu <bytes>
```

Task: Change the MTU size for all Gigabit and 10-Gigabit Ethernet interfaces on the switch.

```
(config)# system mtu jumbo <bytes>
```

Task: Change the system MTU for routed ports.

```
(config)# system mtu routing <bytes>
```



The system routing MTU is the maximum MTU for routed packets and is also the maximum MTU that the switch advertises in routing updates for protocols such as OSPF.

42.9. Switch Internal Processing

Switches forward frames when necessary, and do not forward when there is no need to do so, thus reducing overhead.

- Main actions
 - Learn MAC addresses by examining the source MAC address of each received frame
 - Decide when to forward a frame or when to filter (not forward) a frame, based on the destination MAC address
 - Create a loop-free environment with other bridges by using the Spanning Tree Protocol

Store-and-forward

The switch fully receives all bits in the frame (store) before forwarding the frame (forward). This allows the switch to check the FCS before forwarding the frame, thus ensuring that errored frames are not forwarded.

Cut-through

The switch performs the address table lookup as soon as the Destination Address field in the header is received. The first bits in the frame can be sent out the outbound port before the final bits in the incoming frame are received. This does not allow the switch to discard frames that fail the FCS check, but the forwarding action is faster, resulting in lower latency.

Fragment-free

This performs like cut-through switching, but the switch waits for 64 bytes to be received before forwarding the first bytes of the outgoing frame. According to Ethernet specifications, collisions should be detected during the first 64 bytes of the frame, so frames that are in error because of a collision will not be forwarded.

42.10. Switching and Bridging Logic

Type of Address	Switch Action
Known unicast	Forwards frame out the single interface associated with the destination address
Unknown unicast	Floods frame out all interfaces, except the interface on which the frame was received
Broadcast	Floods frame identically to unknown unicasts
Multicast	Floods frame identically to unknown unicasts, unless multicast optimizations are configured

42.11. Telnet

- allows a user at one site to establish a TCP connection to a login server at another site, then passes the keystrokes from one system to the other.

- offers 3 main services:
 - Network virtual terminal connection
 - Option negotiation
 - Symmetric connection
- Cisco implementation of Telnet supports the following Telnet options:
 - Remote echo
 - Binary transmission
 - Suppress go ahead
 - Timing mark
 - Terminal type
 - Send location
 - Terminal speed
 - Remote flow control
 - X display location

Task: Specify the IP Address Of an Interface As the Source Address for Telnet Connections

```
(config)# ip telnet source-interface <interface>
```



- If the specified interface is not up, IOS selects the address of the interface closest to the destination as the source address.

Task: Set the TCP Window to Zero When the Telnet Connection Is Idle

```
(config)# service telnet-zeroidle
```



- Normally, data sent to noncurrent Telnet connections is accepted and discarded. When **service telnet-zero-idle** is enabled, if a session is suspended (that is, some other connection is made active or the EXEC is sitting in command mode), the TCP window is set to zero. This action prevents the remote host from sending any more data until the connection is resumed.
- Use this command when it is important that all messages sent by the host be seen by the users and the users are likely to use multiple sessions.
- Do not use this command if your host will eventually time out and log out a TCP user whose window is zero.

Task: Hide Host Address While Attempting to Establish a Telnet Session

```
(config)# ip telnet hidden addresses
```

Task: Mark Outgoing Traffic with IP Precedence

```
(config)# ip telnet tos <0-7>
```

Task:

```
(config)# ip telnet quiet
```

Task: Display Error Message When Telnet Connection Fails to a Specific Host

```
(config)# busy-message <hostname> d message d
```

Task: Sets Line to Send a RETURN (CR) As a CR Followed by a NULL Instead Of a CR Followed by a LINE FEED (LF).

```
(config)# ip telnet transparent
```

42.12. SSH

Example

```
!--- Step 1: Configure a hostname and domain name
(config)# hostname router
(config)# ip domain-name nyc.cisco.com

!--- Step 2: Generate an RSA key pair, automatically enabling SSH.
(config)# cry key generate rsa

!--- Step 3: Configure time-out and number of authentication retries.
(config)# ip ssh time-out 60
(config)# ip ssh authentication-retries 2

!--- Step 4: Configure VTYs to only accept SSH.
(config)# line vty 0 4
(config-line)# transport input ssh

!--- Step 5: Allow SSH connections only originated from the management network.
(config)# access-list 111 remark ACL for SSH
(config)# access-list 111 permit tcp 172.26.0.0 0.0.255.255 any eq 22
(config)# access-list 111 deny ip any any log-input
(config)# line vty 0 4
(config-line)# access-class 111 in
```

42.13. SCP

42.14. [T]FTP

- UDP 69

passive-mode vs active mode

42.14.1. FTP Client

TODO

42.14.2. FTP Server

TODO

42.15. SNMP

[Configuration guides](#) › [Network Management](#) › [configuring SNMP support](#)

- Application-layer protocol between SNMP managers and agents
 - SNMP manager running NMS software: UDP port 162 open for traps/informs messages
 - SNMP managed devices running SNMP agent: UDP 161 open for GET/SET messages

42.15.1. Version

v1

original specs, weak authentication with community string, Uses SMIV1, uses MIB-I originally.

v2

Uses SMIV2, removed requirement for communities, added GetBulk and Inform messages, began with MIB-II originally.

v2c

64-bit counters, getBulkRequest, informsRequest,Pseudo-release (RFC 1905) that allowed SNMPv1-style communities with SNMPv2;

v3

authentication, encryption, Mostly identical to SNMPv2, but adds significantly better security, although it supports communities for backward compatibility. Uses MIB-II.

42.15.2. MIB

MIB: dictionaries of OID

OID: hierarchical identifiers in numerical format that represent MIB variables. (e.g. 1.3.6.1.2.1 "Interfaces" 1.3.6.1.4.1.9 "Enterprises - Cisco")

42.15.3. Packet Format

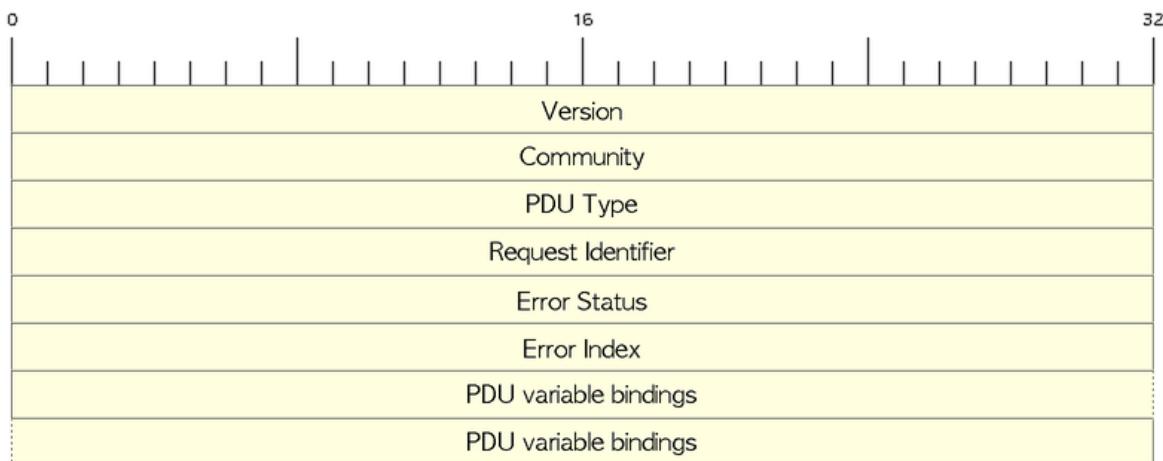


Figure 50. SNMP Header Format

SNMP PDU Types

1. SetRequest
2. GetRequest
3. GetNextRequest
4. GetBulkRequest
5. Trap
6. InformsRequest
7. Response

42.15.4. Basic System Information

Task: Configure the Location Information

```
(config)# snmp-server location <homesweethome>
```

Task: Configure the Contact Information

```
(config)# snmp-server contact <no-where-to-be-found>
```

Task: Configure the System Serial Number

```
(config)# snmp-server chassis-id <system-serial-number>
```

42.15.5. Views

Task: Create a View

```
(config)# snmp-server view <name> <oid-tree> {included | excluded}
```

42.15.6. Communities

Task: Configure a Community String

```
(config)# snmp-server community <string> [view <name>] [ro|rw] [<acl>]
```

Task: Display Community String

```
# sh snmp community
```

42.15.7. Traps

Task: Send Traps to NMS

```
(config)# snmp-server host <ip-address>
          [traps | informs]
          [version {1| 2c | 3 [auth | noauth | priv]}]
          community-string [udp-port port-number] [notification-type]
```

42.15.8. SNMP V3

Task: Configure SNMP V3 Group

```
(config)# snmp-server group [<groupname> {v1 | v2c | v3 [auth | noauth | priv]}]
          [read <readview>]
          [write <writeview>]
          [notify <notifyview>]
          [access <acl>]
```

Task: Display SNMP V3 Group Settings

```
# sh snmp group
```

Task: Configure SNMP V3 User

```
(config)# snmp-server engineID {local <engine-id> | remote <ip-address> [udp-port
<number> ] [vrf <vrf-name> ] <engine-id-string> }
(config)# snmp-server user <username> <groupname> [remote <ip-address> [udp-port
<number> ]] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} <auth-password> ]} [access
<acl>]
```

Task: Display SNMP User Information

```
# sh snmp user <user>
```

Task: Display SNMP EngineID

```
# sh snmp engineID
```

42.15.9. SNMP Manager

- control and monitor the activities of network hosts using SNMP.
- Network Management System (NMS)
 - can be dedicated device used for network management, or the applications used on such a device.
 - can be CLI or GUI (CiscoWorks2000)

Task: Configure the SNMP Manager Process

```
(config)# snmp-server manager
```

Task: Configure the SNMP Manager Session Time-Out

```
(config)# snmp-server manager session-timeout <seconds>
```

Task: Display the Status Of the SNMP Sessions

```
# sh snmp sessions brief
```

Task: Display the Current Set Of Pending SNMP Requests

```
# sh snmp pending
```

42.15.10. SNMP Shutdown Mechanism

Task: Enable the SNMP Shutdown Mechanism

```
(config)# snmp-server system-shutdown
```

Task: Define the Maximum SNMP Agent Packet Size

```
(config)# snmp-server packetsize <bytes>
```

Task: Specify the TFTP Servers Used for Saving and Loading Configuration Files

```
(config)# snmp-server tftp-server-list <acl>
```

Task: Disable SNMP Agent

```
(config)# no snmp-server
```

42.16. RMON

TODO clean this section

Remote Monitoring, or RMON, is an event-notification extension of the SNMP capability on a Cisco router or switch. RMON enables you to configure thresholds for alerting based on SNMP objects, so that you can monitor device performance and take appropriate action to any deviations from the normal range of performance indications. RMON is divided into two classes: alarms and events. An event is a numbered, user-configured threshold for a particular SNMP object. You configure events to track, for example, CPU utilization or errors on a particular interface, or anything else you can do with an SNMP object.

You set the rising and falling thresholds for these events, and then tell RMON which RMON alarm to trigger when those rising or falling thresholds are crossed. For example, you might want to have the router watch CPU utilization and trigger an SNMP trap or log an event when the CPU utilization rises faster than, say, 20 percent per minute. Or you might configure it to trigger an alarm when the CPU utilization rises to some absolute level, such as 80 percent. Both types of thresholds (relative, or delta," and absolute) are supported. Then, you can configure a different alarm notification as the CPU utilization falls, again at some delta or to an absolute level you specify.

The alarm that corresponds to each event is also configurable in terms of what it does (logs the event or sends a trap). If you configure an RMON alarm to send a trap, you also need to supply the SNMP community string for the SNMP server.

Event and alarm numbering are locally significant. Alarm numbering provides a pointer to the corresponding event. That is, the configured events each point to specific alarm numbers, which you must also define.

Example

```
rmon event 1 log trap public description Fa0.0RisingErrors owner config
rmon event 2 log trap public description Fa0.0FallingErrors owner config
rmon event 3 log trap public description Se0.0RisingErrors owner config
rmon event 4 log trap public description Se0.0FallingErrors owner config
rmon alarm 11 ifInErrors.1 60 delta rising-threshold 10 1 falling-threshold 5 2 owner config
rmon alarm 20 ifInErrors.2 60 absolute rising-threshold 20 3 falling-threshold 10 4 owner config
```

Task: Monitor RMon Activity

```
# sh rmon activity  
# sh rmon event
```

42.17. Syslog

- RFC 5424
- Clear-text protocol that provides event notifications without requiring difficult, time-intensive configuration or opening attack vectors.

Steps

- Install a Syslog server on a workstation with a fixed IP address.
- Configure the logging process to send events to the Syslog server's IP address using the **logging host** command.
- Configure any options, such as which severity levels (0–7) you want to send to the Syslog server using the logging trap command.

42.18. NTP

[Configuration guides](#) › [Network Management](#) › [Basic System Management](#) › [Setting Time and Calendar Services](#)

- Version 3 [RFC 1305](#)
- UDP port 123
- IOS does not support stratum 1 service, cannot be linked to stratum 0 atomic clock
- Accuracy < milliseconds with 1 NTP packet per minute
- Stratum
 - 0 : atomic clock
 - 8 : default value

42.18.1. NTP Associations

- Polled-based : better accuracy and reliability
 - Client mode : **ntp server**
 - Symmetric active mode : **ntp peer**
- Broadcast-based : less manual configuration on LAN
 - Server : **ntp broadcast**
 - Client : **ntp broadcast client**

Task: Verify NTP Status

```
# show ntp status
```

Task: Verify NTP Associations

```
# show ntp associations
```

Task: Troubleshoot NTP Associations

```
# debug ntp refclock
```

Polled-Based Associations

```
(config)# ntp server <ip-address> [normal-sync] [version <number>] [key <id>] [prefer]  
(config)# ntp peer <ip-address> [normal-sync] [version <number>] [key <id>] [prefer]
```

Broadcast-Based Associations

```
(config-if)# ntp broadcast version <number>  
(config-if)# ntp broadcast client  
(config-if)# ntp broadcastdelay <microseconds>
```

42.18.2. NTP Access Groups

Task: Grant/Deny Access Privileges with Ipv4 or Ipv6 Access-Lists

```
(config)# ntp access-group [ipv4 | ipv6] <options> <access-list-id> [kod]
```

- *options* per increasing order of restrictions are:
 - **peer**: synchronize itself to systems whose address passes access list criteria
 - **serve**: allows time requests and NTP control queries but no synchronization
 - **serve-only**: allows only time requests
 - **query-only**: allows only NTP control queries
- **kod** sends the kiss-of-death packet to any host that tries to send a packet that is not compliant with the access-group policy.



42.18.3. NTP Authentication

- Use cryptographic checksum keys
- Encryption/decryption are CPU-intensive and may degrade accuracy

```
(config)# ntp authenticate  
(config)# ntp authentication-key <number> md5 <key>  
(config)# ntp trusted-key <key-number> [-<end-key-number>]  
(config)# ntp server <ip-address> key <id>
```

42.18.4. Source IP Address

- Default to NTP packet outgoing interface

Task: Change the Source IP Address for All Destinations

```
(config)# ntp source interface
```

Task: Change the Source for a Specific Association

```
(config)# ntp {server|peer} source
```

42.18.5. Authoritative Server

Task:

```
(config)# ntp master
```

42.18.6. Panic Threshold

Task: Reject Time Updates Greater Than the Panic Threshold Of 1000 Seconds

```
(config)# ntp panic update
```

42.18.7. Orphan Mode

- When a subnet lost communications with clock servers
- Orphan parent simulate a UTC source for orphan children

```
(config)# ntp server <a.b.c.d>  
(config)# ntp peer <e.f.g.h>  
(config)# ntp orphan <stratum>
```

42.18.8. External Reference Clock

```
# line aux <number>  
# ntp refclock trimble pps none stratum <number>
```

42.18.9. Software Clock

```
(config)# clock timezone <zone> <hours-offset> [<minutes-offset>]
(config)# summer-time <zone> recurring [<week day month hh:mm> [<offset>]]
(config)# summer-time <zone> date [<date month year hh:mm> [<offset>]]
# clock set <hh:mm:ss date month year>
# show clock
```

42.18.10. Hardware-Clock

- different from software-clock

```
# calendar set <hh:mm:ss date month year>
(config)# clock calendar-valid
# clock read-calendar
# clock update-calendar
# show calendar
# show clock [detail]
# show ntp associations [details]
# show ntp status
```

42.18.11. Time Ranges

Task: Configure Time Ranges

```
(config)# time-range <name>
(config-time-range)# absolute [start <hh:mm date month year>] [end <hh:mm date month
year>]
(config-time-range)# periodic <day-of-week> <hh:mm> to [<day-of-the-week>] <hh:mm>
```

Task: Verify Time Range

```
# show time-range
```

42.18.12. Vulnerability

- DoS for version \leq 4.2.4p7
- No workaround, disable NTP on the device

42.18.13. Example

- Step 1 Enable timestamp information for debug messages.
- Step 2 Enable timestamp information for log messages.
- Step 3 Define the network-wide time zone.

- Step 4 Enable summertime adjustments.
- Step 5 Restrict which devices can communicate with this device as an NTP server.
- Step 6 Restrict which devices can communicate with this device as an NTP peer.
- Step 7 Define the source IP address to be used for NTP packets.
- Step 8 Enable NTP authentication.
- Step 9 Define the NTP servers.
- Step 10 Define the NTP peers.
- Step 11 Enable NTP to update the device hardware clock

42.19. HTTP

Cisco IOS routers and switches support web access for administration, through both HTTP and HTTPS. Enabling HTTP access requires the **ip http server** global configuration command. HTTP access defaults to TCP port 80. You can change the port used for HTTP by configuring the **ip http port** command. You can restrict HTTP access to a router using the **ip http access-class** command, which applies an extended access list to connection requests. You can also specify a unique username and password for HTTP access using the **ip http client username** and **ip http client password** commands. If you choose, you can also configure HTTP access to use a variety of other access-control methods, including AAA, using **ip http authentication [aaa | local | enable | tacacs]**.

You can also configure a Cisco IOS router or switch for Secure Sockets Layer (SSL) access. By default, HTTPS uses TCP port 443, and the port is configurable in much the same way as it is with HTTP access. Enabling HTTPS access requires the **ip http secure-server** command. When you configure HTTPS access in most IOS Release 12.4 versions, the router or switch automatically disables HTTP access, if it has been configured. However, you should disable it manually if the router does not do it for you.

HTTPS router access also gives you the option of specifying the cipher suite of your choice. This is the combination of encryption methods that the router will enable for HTTPS access. By default, all methods are enabled

Task: Display HTTP Secure Server Status

```
# sh ip http server secure status

HTTP secure server status          : Enabled
HTTP secure server port            : 443
HTTP secure server ciphersuite    : 3des-edc-cbc-sha des-cbc-sha rc4-128-md5
rc4-128-sha
HTTP secure server client authentication : Disabled
HTTP secure server trustpoint      :
HTTP secure server active session modules : ALL
```

42.20. RTP/RTCP

Real-time Transport Control Protocol (RTCP) - sister protocol of RTP - provides out-of-band control information for an RTP flow * It is used periodically to transmit control packets to participants in a streaming multimedia session.

- gathers statistics on a media connection and information such as bytes sent, packets sent, lost packets, jitter, feedback and round trip delay. An application may use this information to increase the quality of service, perhaps by limiting flow or using a different codec.
- type of RTCP packets: Sender report packet, Receiver report packet, Source Description RTCP Packet, Goodbye RTCP Packet and Application Specific RTCP packets.
- RTCP itself does not provide any flow encryption or authentication means. SRTCP protocol can be used for that purpose.

TODO question 1

<http://ccm.net/contents/288-rtp-rtcp-protocols>

Chapter 43. QoS

Blueprint Topics

- 6.2.a Implement and troubleshoot end-to-end QoS
- 6.2.a [i] CoS and DSCP mapping
- 6.2.b Implement, optimize and troubleshoot QoS using MQC
- 6.2.b [i] Classification
- 6.2.b [ii] Network based application recognition [NBAR]
- 6.2.b [iii] Marking using IP precedence, DSCP, CoS, ECN
- 6.2.b [iv] Policing, shaping
- 6.2.b [v] Congestion management [queuing]
- 6.2.b [vi] HQoS, sub-rate ethernet link
- 6.2.b [vii] Congestion avoidance [WRED]
- 6.2.c Describe layer 2 QoS
- 6.2.c [i] Queuing, scheduling
- 6.2.c [ii] Classification, marking

43.1. Classification and Marking

43.1.1. Fields That Can Be Marked for QoS Purposes

- Layer 3: IPP, IP DSCP, DS field, ToS byte
- Layer 2: Ethernet CoS, Frame Relay DE, ATM CLP, MPLS EXP

IPP

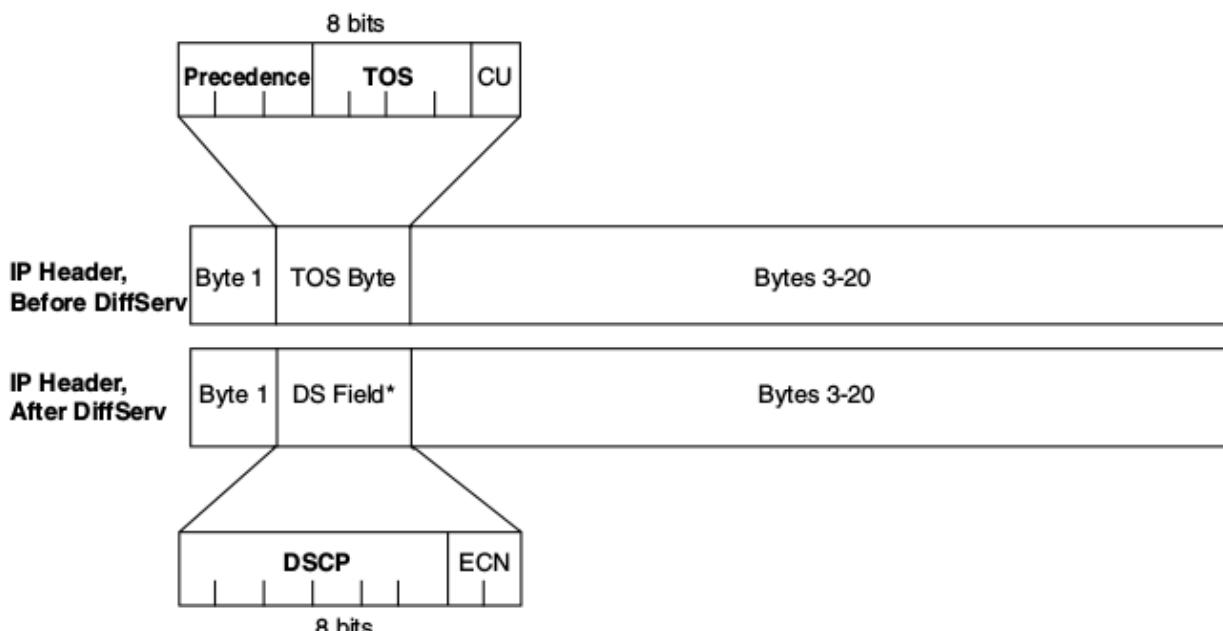


Table 15. IPP and Class Selector DSCP Values

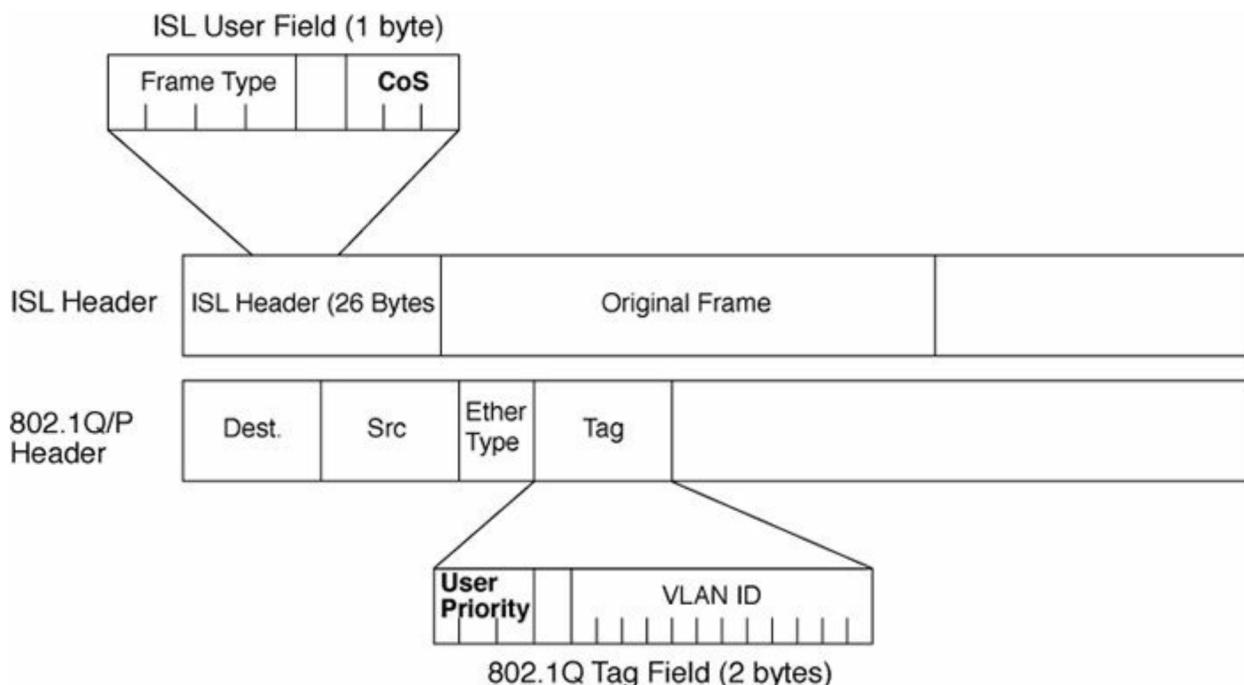
DSCP CS name	IPP names
CS0/Default	routine

DSCP CS name	IPP names
CS1	priority
CS2	immediate
CS3	flash
CS4	flash override
CS5	critical
CS6	internetwork control
CS7	network control

DSCP

- PHB: Per-Hop Behavior
- Expedited Forwarding (EF):
 - Queue EF packets so that they get scheduled quickly, to give them low latency.
 - Police the EF packets so that they do not consume all bandwidth on the link or starve other queues.
 - Binary value: 101110 Decimal value: 46
 - packets given queue priority
- Assured Forwarding (AF)
 - defines 4 queues with 3 drop priorities
 - $AF_{x_1}y_1 > AF_{x_2}y_2$ if $x_1 > x_2$ or ($x_1 = x_2$ and $y_1 < y_2$)
 - $AF_{xy} \rightarrow 8x + 2y = \text{decimal value}$

Ethernet LAN Class Of Service



WAN

- Frame Relay DE (Discard Eligibility), 1-bit
- ATM CLP (Cell Loss Priority), 1-bit
- MPLS EXP (Experimental), 3-bit

43.1.2. NBAR

- Network-Based Application Recognition
- Deep-packet classification from Layer 4 through Layer 7:
 - Statically assigned TCP and UDP port numbers.
 - Non-TCP and non-UDP IP protocols.
 - Dynamically assigned TCP and UDP port numbers.
 - requires stateful inspection: inspect a protocol across multiple packets during packet classification.
 - Support classification or classification based on deep-packet inspection.
 - URL, Hostname, or MIME type
- uses MQC match protocol

Table 16. NBAR Match Protocol

Command	Purpose
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
match protocol citrix	Configures NBAR to match Citrix traffic.
match protocol fasttrack	Configures NBAR to match FastTrack peer-to-peer traffic.
match protocol gnutella	Configures NBAR to match Gnutella peer-to-peer traffic.
match protocol http	Configures NBAR to match HTTP traffic by URL, host,MIME type, or fields in HTTP packet headers.
match protocol rtp	Configures NBAR to match Real-Time Transport Protocol (RTP) traffic.
match qos-group	Identifies a specific QoS group value as a match criterion.
match source-address mac	Uses the source MAC address as a match criterion.
match start	Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3).
match tag	Specifies tag type as a match criterion.

Restrictions

- NBAR cannot classify ipx
- NBAR cannot classify multicast traffic

Task: Configure DSCP-Based Layer 3 Custom Applications

```
(config)# ip nbar custom <name> transport {tcp| udp } id <id>
(config-custom)# dscp {ef | af }
```

Task: Display the Current Protocol-to-Port Mappings In Use by NBAR.

```
# sh ip nbar port-mapping [<protocol-name>]
```

```
port-map bgp      udp 179
port-map bgp      tcp 179
port-map cuseeme  udp 7648 7649
port-map cuseeme  tcp 7648 7649
port-map dhcp     udp 67 68
port-map dhcp     tcp 67 68
```

43.1.3. CB Marking

- requires CEF
- Mark as close to the ingress edge of the network as possible, but not so close to the edge that the marking is made by an untrusted device.

For any class inside the policy map for which there is no set command, packets in that class are not marked.

Table 17. Recommended Values for Traffic Marking

Traffic Type	IP Precedence	IP DSCP	Class of Service
Voice payload	5	EF	5
Video payload	4	AF41	4
Voice and video signaling	3	AF31 3	High priority data
2	AF21 AF22 AF23	2	Medium priority data
1	AF11 AF12 AF13	1	All other traffic

Task: Set the DSCP Value In the ToS Byte

```
(config-pmap-c)#set ip dscp {<0-63> | AF<xy> | CS<x> | EF | default}
```

43.1.4. QoS Pre-Classification

- enabled on VPN endpoint routers permit the router to make egress QoS decisions based on the original traffic, before encapsulation, rather than just the encapsulating tunnel header.
- works by keeping the original, unencrypted traffic in memory until the egress QoS actions are taken.
- enables in tunnel interface configuration mode, virtual-template configuration mode, or crypto

map configuration mode

Task: Enable QoS Pre-Classification

```
(config-if)# qos pre-classification
```

43.1.5. AutoQoS

- macro that helps automate class-based QoS configuration.
- creates and applies QoS configurations based on Cisco best-practice recommendations.
- provides the following benefits:
 - Simpler QoS deployment.
 - Less operator error, because most steps are automated.
 - Cheaper QoS deployment because less staff time is involved in analyzing network traffic and determining QoS configuration.
 - Faster QoS deployment because there are dramatically fewer commands to issue.
 - Companies can implement QoS without needing an in-depth knowledge of QoS concepts

Task: Display the Interface AutoQoS Configuration

```
> sh auto qos
```

AutoQoS for VOIP

- for voice and video applications
- enables on individual interfaces, but creates both interface and global configuration
- uses CDP on access ports to detect presence or absence of softphone
- trusts COS or DSCP values on trunk or uplink ports

AutoQoS on Switches

- no need to enable QoS globally.
 - After it is enabled for any interface, the command starts a macro that:
- Globally enables QoS.
- Creates COS-to-DCSP mappings and DSCP-to-COS mappings.
 - As the traffic enters the switch, the frame header containing the COS value is removed.
 - The switch uses the COS value in the frame header to assign a DSCP value to the packet.
 - If the packet exits a trunk port, the internal DSCP value is mapped back to a COS value.
- Enables priority or expedite ingress and egress queues.
- Creates mappings of COS values to ingress and egress queues and thresholds.
- Creates mappings of DSCP values to ingress and egress queues and thresholds.

- Creates class maps and policy maps to identify, prioritize, and police voice traffic.
- Applies those policy maps to the interface.



For best results, enable AutoQoS before configuring any other QoS on the switch. You can then go back and modify the default configuration if needed to fit your specific requirements.

Task: Enable AutoQoS on an Access Port

```
(config-if)# auto qos voip {cisco- phone | cisco-softphone}
```

Task: Enable AutoQoS on Uplink Port

```
(config-if)# auto qos voip trust
```

AutoQoS on Routers

Task: Enable AutoQoS on Router Port

```
(config-if)# auto qos voip [trust]
```

- Make sure that the interface bandwidth is configured before giving this command.
 - If you change it later, the QoS configuration will not change. When you issue the **auto qos voip** command on an individual data circuit, the configuration it creates differs depending on the bandwidth of the circuit itself.
 - Compression and fragmentation are enabled on links of 768 kbps bandwidth and lower. They are not enabled on links faster than 768 kbps.
 - The router additionally configures traffic shaping and applies an AutoQoS service policy regardless of the bandwidth.
- When you issue the command on a serial interface with a bandwidth of 768 kbps or less, the router changes the interface encapsulation to PPP. It creates a PPP Multilink interface and enables Link Fragmentation and Interleave (LFI) on the interface. Serial interfaces with a configured bandwidth greater than 768 kbps keep their configured encapsulation, and the router merely applies an AutoQoS service policy to the interface.
- If you use the **trust** keyword in the command, the router creates class maps that group traffic based on its DSCP values. It associates those class maps with a created policy map and assigns it to the interface. You would use this keyword when QoS markings are assigned by a trusted device.
- If you do not use the **trust** keyword, the router creates access lists that match voice and video data and call control ports. It associates those access lists with class maps with a created policy map that marks the traffic appropriately. Any traffic not matching those access lists is marked with DSCP 0. You would use this command if the traffic either arrives at the router unmarked or arrives marked by an untrusted device.



AutoQoS for Enterprise

- supported on Cisco routers.
- The main difference between it and AutoQoS VoIP is that it automates the QoS configuration for VoIP plus other network applications, and is meant to be used for WAN links.
- can be used for Frame Relay and ATM subinterfaces only if they are point-to-point links.
- detects the types and amounts of network traffic with NBAR and then creates policies based on that.

Task: Enable Traffic Discovery

```
(config-if)# auto discovery qos [trust]
```



- Make sure that CEF is enabled, that the interface bandwidth is configured, and that no QoS configuration is on the interface before giving the command.
- Use the **trust** keyword if the traffic arrives at the router already marked, and if you trust those markings, because the AutoQoS policies will use those markings during the configuration stage.

Task: Generate the AutoQoS Configuration for Enterprise

```
(config-if)# auto qos
```

Task: Show Auto Discovery Qos

```
# sh auto discovery qos
```

Table 18. AutoQoS for Enterprise Classes and DSCP values

Class	DSCP/PHB value	Traffic Types
Routing	CS6	EIGRP OSPF
VoIP	EF(46)	RTP Voice Media
Interactive video	AF41	RTP Video Media
Streaming video	CS4	Real Audio Netshow
Control	CS3	RTCP H323 SIP
Transactional	AF21	SAP Citrix Telnet SSH
Bulk	AF11	FTP SMTP POP3 Exchange
Scavenger	CS1	Peer-to-peer applications
Management	CS2	SNMP Syslog DHCP DNS
Best effort	All others	All others

43.2. Congestion Management and Avoidance

congestion when hardware queue is full

43.2.1. Queues

hardware queues

software queues

43.2.2. CBWFQ

43.2.3. LLQ

43.3. Shaping, Policing and Link Fragmentation

TODO

43.4. MQC

- Define a traffic class with **class map**
- Define PHB actions(marking, queuing,) with **policy map**
- Attach the traffic policy to an interface with **service-policy**
- restrictions: maximum of 256 classes in a single policy map

43.4.1. Traffic Class

A traffic class contains 3 major elts: a name, a series of **match** commands followed by **match-all** (default) or **match-any**

Match Commands That Can Be Used with MQC

- match access-group
- match any
- match class-map
- match cos
- match destination-address mac
- match discard-class
- match [ip] dscp
- match field
- match fr-dlci
- match input-interface
- match ip rtp
- match mpls experimental
- match mpls experimental topmost
- match not // add a note here
- match packet length -
- etc
- Packets that do not explicitly match a defined class are considered to have matched a special class called **class-default**.

Task: Display All Class Maps and Their Matching Criteria

```
> sh class-map
```

43.4.2. Elements Of a Traffic Policy

- contains 3 elts: name, traffic class, command to enable the QoS feature

Command	Purpose
random-detect discard-class	Configures the WRED parameters for a discard-class value for a class in a policy map.
random-detect discard-class-based	Configures WRED on the basis of the discard class value of a packet.
random-detect ecn	Enables explicit congestion notification (ECN).
random-detect exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class.
random-detect precedence	Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map.
set atm-clp	Sets the cell loss priority (CLP) bit when a policy map is configured.
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set discard-class	Marks a packet with a discard-class value.
set [ip] dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte.
set fr-de	Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface.
set mpls experimental	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
shape adaptive	Configures a Frame Relay interface or a point-to-point subinterface to estimate the available bandwidth by backward explicit congestion notification (BECN) integration while traffic command
shape fecn-adaptive	configures a Frame Relay interface to reflect FECN bits as BECN bits in Q.922 test response messages.

Task: Display the Configuration for the Specified Class Of the Specified Policy Map

```
> show policy-map <policy-name> class <class-name>
```

Task: Display the Configuration Of All Classes for All Existing Policy Maps

```
> show policy-map interface <name number>
```

43.4.3. Service Policy

Task: Enable a Traffic Policy on an Interface

```
(config-if)# service-policy {input | output} policy-map-name
```

43.5. RSVP

43.5.1. Configuration Tasks

```
ip rsvp bandwidth [interface-kbps [single-flow-kbps] ]
```

- default 75 % can be allocated by to RSVP and up to 75% of the total bandwidth is available for a single flow

Chapter 44. First Host Redundancy Protocols

44.1. HSRP

Configuration Guides > First Hop Redundancy Protocols > [HSRP Version 2](#)

- [RFC 2281](#)
- Set of routers sharing one virtual IP address and one virtual MAC address
- Elect one active router with the highest (priority, IP address)
- Standby router takes over if Active router fails
- Elect new standby router if standby router fails or becomes the active router.
- To minimize network traffic, only the active and standby router send periodic HSRP messages.
- A router may participate in multiple groups with separate state and timers for each group
- Unique group id per vlan
- Multicast address = 0000.0c07.ACxy (where xy is the HSRP group number in Hex)
- Send to multicast 224.0.0.2 for v1 and 224.0.0.102 for v2
- UDP port 1985

44.1.1. HSRP Packet

MAC header	IP header	UDP packet port=1985	HSRP Packet
------------	-----------	----------------------	-------------

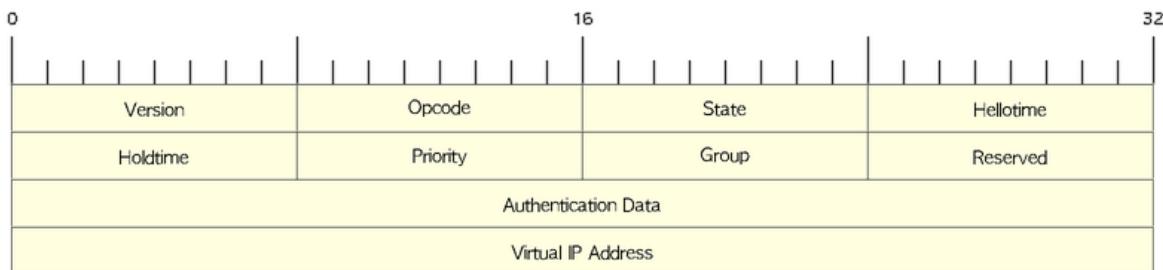


Figure 51. HSRP Packet

44.1.2. HSRP Version

- Version 1 by default
- Version 2 doesn't interoperate with HSRPv1 on the same interface.
 - v1 and v2 can run on different physical interfaces of the same router
 - v1 cannot advertise and learn millisecond timers
 - v1 uses multicast 224.0.0.2 and v2 224.0.0.102

- v2 uses jitter timers (negative for hellotimes and positive for holdtimes)
- Group number: 0..255 for v1 and 0..4095 for v2
- Different virtual MAC address

Task: Change HSRP Version

```
(config-if)# standby version {1 | 2}
```

44.1.3. HSRP OpCode

- Uses the Opcode field for preemption
 - 0: **Hello**: The router is running and is capable of becoming the active or standby router
 - 1: **Coup**: The router wishes to become the active router
 - 2: **Resign**: The router no longer wishes to be active router
- No preemption by default
- Preemption: the router with the highest priority becomes immediately the active router by sending a **coup** message, The previous active router changes to the **speak** state and sends a **resign** message.
- Can specify a delay before take over to allow the router to populate its routing table
 - **minimum** seconds after the last restart:
 - **reload** seconds after the first interface-up event after the device has reloaded, if such an event occurs within 360 seconds from reload. ???
 - **sync** seconds: for IP redundancy clients only ???

Task: Configure HSRP Preemption

```
(config-if)# standby [ <group-number> ] preempt [ delay{ [ minimum <seconds> ] [ reload <seconds> ] [ sync <seconds> ] } ]
```

44.1.4. HSRP State

Code	State	Description
0	Initial	This is the starting state and indicates that HSRP is not running. This state is entered via a configuration change or when an interface first comes up.
1	Learn	The router has not determined the virtual IP address, and not yet seen an authenticated Hello message from the active router. In this state the router is still waiting to hear from the active router.
2	Listen	The router knows the virtual IP address, but is neither the active router nor the standby router. It listens for Hello messages from those routers.

Code	State	Description
4	Speak	The router sends periodic Hello messages and is actively participating in the election of the active and/or standby router. A router cannot enter Speak state unless it has the virtual IP address.
8	Standby	The router is a candidate to become the next active router and sends periodic Hello messages. Excluding transient conditions, there MUST be at most one router in the group in Standby state.
16	Active	The router is currently forwarding packets that are sent to the group's virtual MAC address. The router sends periodic Hello messages. Excluding transient conditions, there MUST be at most one router in Active state in the group.

44.1.5. Priority

- Default value: 100
- The higher (priority || IP address) wins

Task: Configure HSRP Priority

```
(config-if)# standby [group-number] priority <number>
```

44.1.6. HSRP Timers

Hellotime

- 3 seconds by default
- Only meaningful in Hello messages
- Configured on the router or learned from authenticated Hello message from the active router
 - not learned if HSRP hellos < 1 second

Holdtime

- 10 seconds by default
- $\geq 3 * \text{hellotime}$

Task: Configure HSRP Timers

```
(config-if)# standby [group-number] timers[msec] <hellotime> [msec] <holdtime>
```

44.1.7. HSRP Authentication

- Clear-text or MD5 encryption

Task: Configure HSRP Clear-Text Authentication

```
(config-if)# standby [group-number] authentication text <string>
```

Task: Configure HSRP MD5 Authentication

```
(config-if)# standby [group-number] authentication md5 { key-string [ 0 | 7 ] key [ timeout seconds ] | key-chain <name-of-chain> }
```

Task: Debug HSRP Authentication

```
# debug standby errors
```

44.1.8. HSRP and Object Tracking

- Reduce HSRP priority if the monitored interface goes down, allowing another HSRP router to become active if it has preemption enabled.
- Cumulative reduction if multiple tracked interfaces are down
- Configurable decrement value (default = 10)
- Can shutdown/change the HSRP group to the Init state on the basis of the tracked object's state

Task: Configure Interface Tracking

```
(config-if)# standby track { <object-number> [<priority-decrement>] | interface-type <interface-number> [ decrement <priority-decrement> ] } [shutdown]
```

44.1.9. HSRP Support for ICMP Redirects

- Enabled by default with advertisement every 60 seconds and holddown of 180 seconds

Why?

When HSRP is running, preventing hosts from discovering the interface (or real) IP addresses of devices in the HSRP group is important. If a host is redirected by ICMP to the real IP address of a device, and that device later fails, then packets from the host will be lost.

How?

- looks up the next hop IP address in its table of real IP addresses vs virtual IP address
- if match found, replaces the real IP address by the virtual IP addresses in the gateway field of the redirect packet
- if no match (unknown), send the redirect packet to go out unchanged

Restrictions

- Do not redirect to passive HSRP devices

Task: Enable ICMP Redirects on an Interface

```
(config-if)# standby redirect [timers <advertisement> <holddown>] [unknown]
```

Task: Disable ICMP Redirects on an Interface

```
(config-if)# standby redirect [timers <advertisement> <holddown>] [unknown]
```

Task: Configure ICMP Redirect Messages with HSRP Virtual IP Address As the Gateway IP Address

```
(config)# standby redirects [enable | disable]
```

Task: Debug HSRP Support for ICMP Redirects

```
# debug standby events icmp
```

```
10:43:08: HSRP: ICMP redirect not sent to 10.0.0.4 for dest 10.0.1.2
```

```
10:43:08: HSRP: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

44.1.10. HSRP Virtual IP Address and Group

- Can have a name (no longer than 25 chars)

Task: Configure the Virtual IP Address

```
(config-if)# standby [<group-number>] ip [<a.b.c.d> [secondary]]
```

- By default, send one gratuitous ARP when a group becomes active and then another two and four seconds later.
- When HSRP is on the Active state on an interface, Proxy ARP requests are answered with the MAC address of the HSRP group. otherwise, they are ignored.

Task: Configure the Number Of Gratuitous ARP Packets Sent by HSRP Group When It Transitions to the Active State, and How Often the ARP Packets Are Sent

```
(config-if)# standby arp gratuitous [count <number=> interval <seconds>]
```

Task: Configure the Name Of the Standby Group

```
(config-if)# standby name <group-name>
```

44.1.11. Multiple HSRP

- Provides load sharing in an HSRP configuration
 - Because HSRP uses only one Active router at a time, any other HSRP routers are idle.
 - two or more HSRP groups are configured on each HSRP LAN interface, where the configured priority determines which router will be active for each HSRP group.
- requires that each DHCP client and statically configured host are issued a default gateway corresponding to one of the HSRP groups

- requires that they're distributed appropriately.

44.2. GLBP

[Configuration Guides](#) › [First Hop Redundancy Protocols](#) › [Gateway Load Balancing Protocol](#)

- One AVG (active virtual gateway) per group
- Up to 4 primary AVFs (active virtual forwarder)
- Up to 1024 GLBP groups per physical interface
- AVG responds to all ARP requests with MAC from all participating AVF
 - Not with own MAC address like HSRP
- Load-balancing
 - Round-robin (default)
 - Host-dependent
 - Weighted
- Multicast, 224.0.0.102, UDP port 3222
- Virtual MAC address 0007.B400.xxxy, where xx the group and yy in (00-03)

44.2.1. GLBP Packet Type

Hello

Advertise protocol info, multicast, sent by any AVG or AVF in [Speak, Standby, Active] State

Request

Request virtual MAC, unicast sent to AVG

Reply

Get virtual MAC, unicast sent from AVG

44.2.2. Active Virtual Gateway

- Elected for each group max(priority, IP address)
- Assigns up to 4 virtual MAC address to AVFs
- Responds to ARP request for the virtual IP address of the group
- If failed, new AVG election from AVG in listen State
- By default, no AVG preemption
- Load balancing methods
 - Round-robin: by default, each AVF in turn is used for ARP
 - Host-dependent: use same AVF based on the host MAC
 - Weighted: use the weight value advertised by the gateway

Task: Configure GLBP Virtual Ip Address

```
(config-if)# glbp <group:0..1023> ip [<virtual-ip-address> [secondary]]
```

i When the glbp ip command is enabled on an interface, the handling of proxy ARP requests is changed (unless proxy ARP was disabled). the AVG intercepts the ARP requests and replies to the ARP on behalf of the connected nodes. If a forwarder in the GLBP group is active, proxy ARP requests are answered using the MAC address of the first active forwarder in the group. If no forwarder is active, proxy ARP responses are suppressed.

Task: Configure GLBP Priority

```
(config-if)# glbp [<group>] priority <level=100>
```

Task: Configure GLBP Preemption

```
(config-if)# glbp [<group>] preempt [delay minimum <seconds>]
```

Task: Configure GLBP Load Balancing

```
(config-if)# glbp <group> load-balancing [<host-dependent> | <round-robin> | <weighted> ]
```

Task: Configure the Time During Which the AVG Continues to Redirect Clients to a Secondary AVF

```
(config-if)# glbp <group> timers redirect <redirect-seconds=60> <timeout-seconds=1440>
```



- The redirect timer sets the time delay between a forwarder failing on the network and the AVG assuming that the forwarder will not return. The virtual MAC address to which the forwarder was responsible for replying is still given out in Address Resolution Protocol (ARP) replies, but the forwarding task is handled by another router in the GLBP group.
- The timeout interval is the time delay between a forwarder failing on the network and the MAC address for which the forwarder was responsible becoming inactive on all of the routers in the GLBP group. After the timeout interval, packets sent to this virtual MAC address will be lost. The timeout interval must be long enough to allow all hosts to refresh their ARP cache entry that contained the virtual MAC address.

Task: Display Glbp Status

```
# sh glbp

FastEthernet0/0.40 - Group 1
  State is Standby
    1 state change, last state change 00:02:02
  Virtual IP address is 172.16.26.100
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.720 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Preemption disabled
  Active is 172.16.26.6, priority 100 (expires in 11.360 sec)
  Standby is local
  Priority 100 (default)
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: round-robin
  Group members:
    ca02.6150.0000 (172.16.26.2) local
    ca06.618c.0000 (172.16.26.6)
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Listen
    MAC address is 0007.b400.0101 (learnt)
    Owner ID is ca06.618c.0000
    Time to live: 14399.840 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 172.16.26.6 (primary), weighting 100 (expires in 10.944 sec)
  Forwarder 2
    State is Active
      1 state change, last state change 00:02:08
    MAC address is 0007.b400.0102 (default)
    Owner ID is ca02.6150.0000
    Preemption enabled, min delay 30 sec
    Active is local, weighting 100
```

44.2.3. Active Virtual Forwarder

- Primary AVF gets virtual MACs from AVG
- Secondary AVF learns virtual MACs from hellos
- Virtual forwarder preemptive is enabled by default with 30 seconds delay
- Uses weighting and object tracking to determine the forwarding capacity of each device in the GLBP group
 - Decrement or increments the weight when the interface goes down or up
 - Stops being AVF if value below lower threshold
 - Resumes being AVF if value greater than upper threshold
 - When multiple tracked interfaces are down, the configured weighting decrements are

cumulative.

Task: Specify GLBP Initial Weighting Value

```
(config-if)# glbp <group-number> weighting <maximum> [lower <low-value> ] [upper <up-value>]
```

Task: Specify a Tracking Object Where GLBP Weighting Changes Based on the Availability Of the Object Being Tracked

```
(config-if)# glbp <group> weighting track <object-number> [decrement <value>]
```

Task: Configure a Router to Take Over As (AVF) Group If the Current AVF Falls Below Its Low Weighting Threshold

```
(config-if)# glbp <group> preempt forwarder [delay minimum <seconds>]
```

44.2.4. Authentication

- Supports no authentication, plain-text, or MD5 authentication

Task: Configure Glbp Authentication

```
(config-if)# glbp authentication { text <string> | key-chain <name> }
```

44.3. VRRP

- [Configuration Guides](#) › [First Hop Redundancy Protocols](#) › [VRRP](#)
- [RFC 3768](#)
- Similar to HSRP
- Multicast virtual MAC address (0000.5E00.01xx, where xx is the hex VRRP group number).
- Uses the IOS object tracking feature, rather than its own internal tracking mechanism, to track interface states for failover purposes.
- Preemption by default
- The group IP address is the interface IP address of one of the VRRP routers.

Task: Enable VRRP

```
(config-if)# vrrp <group> []
```

Task: Verify VRRP Configuration

```
# sh vrrp
```

Task: Customize VRRP

```
conf t
interface <type number>
  ip address ip-address mask
  vrrp <group> description text
  vrrp <group> priority level
  vrrp <group> preempt [delay minimum seconds]
  vrrp <group> timers learn
```

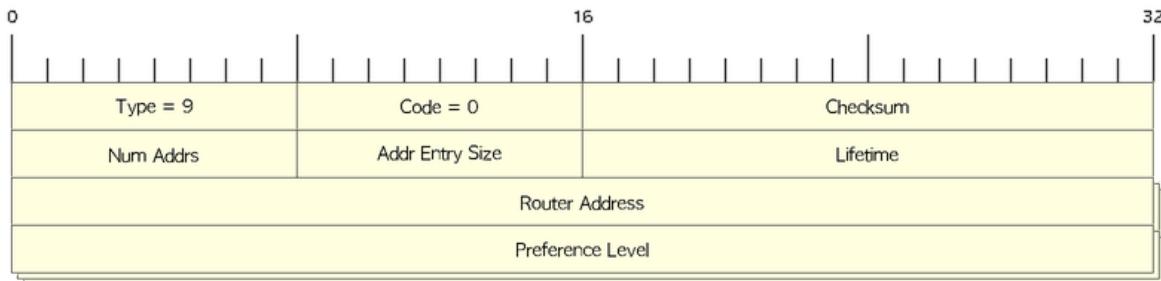
44.4. IDRP

Configuration Guides > First Hop Redundancy Protocols > [IRDP - RFC 1256](#)

- ICMP Router Discovery Protocol allows hosts to locate routers that can be used as gateway to reach IP-based devices on other networks.

44.4.1. Message Format

ICMP Router Advertisement Message



Checksum

The 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP Type. For computing the checksum, the Checksum field is set to 0.

Num Addrs

Number of router addresses advertised in this message

Addr Entry Size

Number of 32-bit words of information per router address (=2 for IPv4)

Lifetime

Maximum number of seconds that the router addresses may be considered valid.

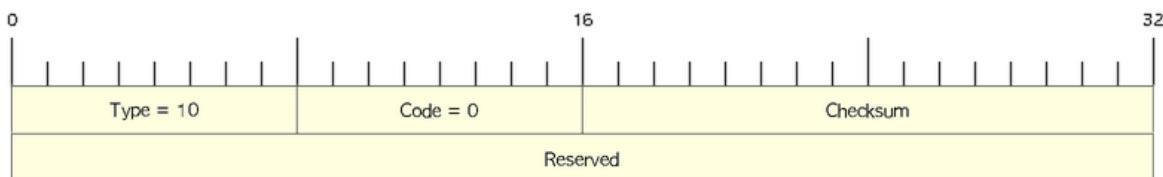
Router Address[i]

Sending router's addresses on the interface from which this message is sent.

Preference Level[i]

Preferability of each Router Address[i] as a default router, relative to other router addresses on the same subnet. Higher values more preferable.

ICMP Router Solicitation Message



Checksum

The 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP Type. For computing the checksum, the Checksum field is set to 0.

Reserved

Sent as 0; ignored on reception.

44.4.2. Configuration

Task: Configure a Host to Discover Routers That Transmit IRDP Router Updates After Disabling IP Routing

```
(config)# no ip routing  
(config-if)# ip gdp irdp [multicast]
```

Task: Enable IRDP on an Interface

```
(config-if)# ip irdp
```

Task: Send IRDP Advertisement to the All-Systems Multicast Addresses

```
(config-if)# ip irdp multicast
```

Task: Set the IRDP Period for Which Advertisements Are Valid.

```
(config-if)# ip irdp holdtime <seconds>
```

Task: Sets the IRDP Maximum Interval Between Advertisements.

```
ip irdp maxadvertinterval <seconds>
```

Task: Set the IRDP Minimum Interval Between Advertisements.

```
ip irdp minadvertinterval <seconds>
```

Task: Set the IRDP Preference Level Of the Device

```
(config-if)# ip irdp preference <number>
```

Task: Specify an IRDP Address and Preference to Proxy-Advertise

```
(config-if)# ip irdp address <a.b.c.d> <preference-level>
```

44.5. IPv6 RA/RS

Chapter 45. Multicast

45.1. IP Multicast

IP multicasting

Sending a message from a single source to selected multiple destinations across a Layer 3 network in one data stream.

45.1.1. Multicast IP Addressing

Table 19. Multicast Address Ranges and Their Use

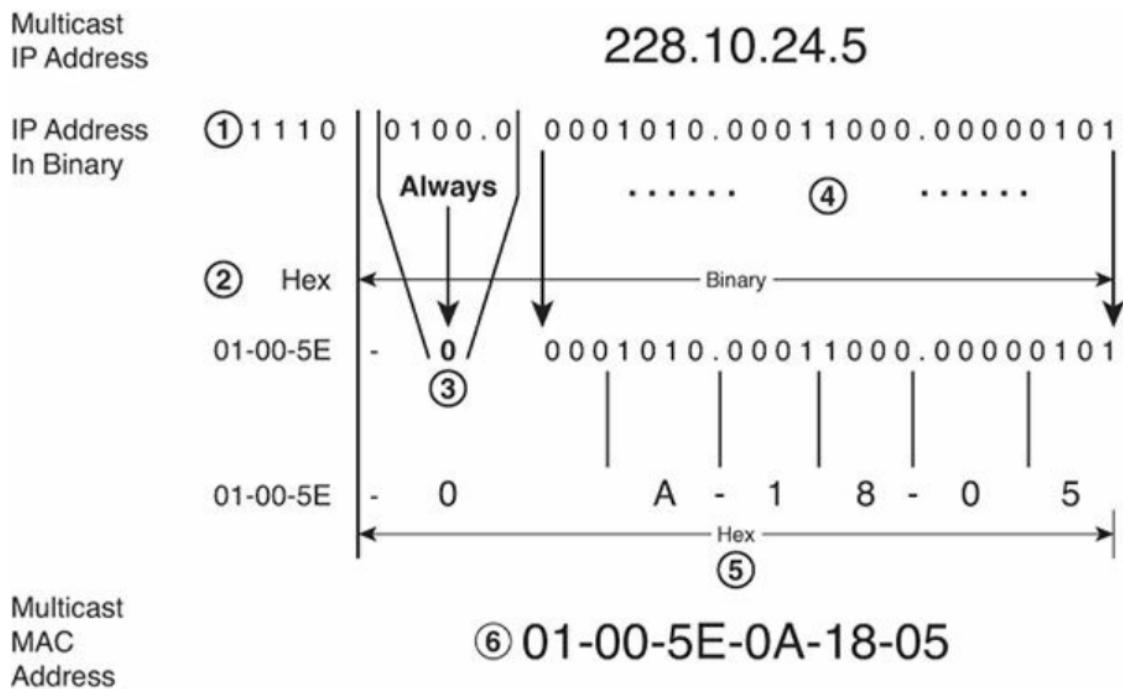
Range	Usage
224.0.0.0 to 239.255.255.255	IPv4 class D reserved for multicast applications
224.0.0.0 to 224.0.0.255	permanent groups; Assigned by IANA for network protocols on a local segment. Routers do not forward packets with destination addresses used from this range.
224.0.1.0 to 224.0.1.255	permanent groups; Assigned by IANA for the network protocols that are forwarded in the entire network. Routers forward packets with destination addresses used from this range.
232.0.0.0 to 232.255.255.255	for SSM applications.
233.0.0.0 to 233.255.255.255	GLOP addressing. It is used for automatically allocating 256 multicast addresses to any enterprise that owns a registered ASN.
239.0.0.0 to 239.255.255.255	Administratively scoped addresses; for private multicast domains
Remaining ranges of addresses	transient groups; Any enterprise can allocate a multicast address from the transient groups for a global multicast application and should release it when the application is no longer in use.

Table 20. Well-Known Multicast Addresses

Address	Usage
224.0.0.1	All multicast hosts
224.0.0.2	All multicast routers
224.0.0.4	DVMRP routers
224.0.0.5	All OSPF routers
2224.0.0.9	RIPv2 routers
224.0.0.10	EIGRP routers
224.0.0.13	PIM routers
224.0.0.22	IGMPv3
224.0.0.25	RGMP
224.0.1.39	Cisco-RP-Announce
224.0.1.40	Cisco-RP-Discovery

45.1.2. Mapping IP Multicast Address to MAC Addresses

ip → mac address = (0x01005E) + 0 bit + last 23 bits of the IP address



45.2. IGMP

Configuration guides > Multicast > **IGMP**

Check also [Catalyst configuration guides](#)

- Group membership protocol used by hosts to inform routers and multilayer switches of the existence of members on their directly connected networks and to allow them to send and receive multicast datagrams.
- IP protocol number: 2
- TTL: 1

45.2.1. IGMP Packet Format

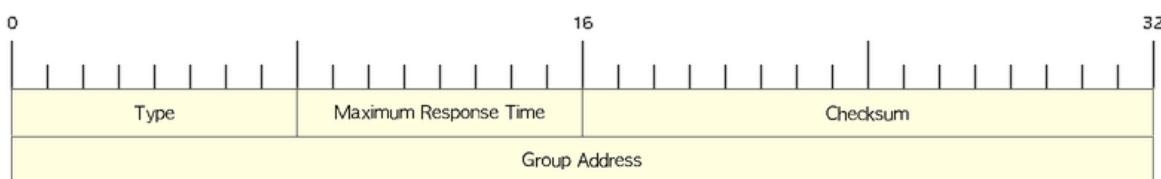


Figure 52. IGMP Version 2 Format

Type

- Membership Query (Type code = 0x11):
 - Used by multicast routers to discover the presence of group members on a subnet.

- A General Membership Query message sets the Group Address field to 0.0.0.0.
- A Group-Specific Query sets the Group Address field to the address of the group being queried.
- It is sent by a router after it receives the IGMPv2 Leave Group message from a host.
- Version 1 Membership Report (Type code = 0x12):
- Used by IGMPv2 hosts for backward compatibility with IGMPv1.
- Version 2 Membership Report (Type Code = 0x16):
 - Sent by a group member to inform the router that at least one group member is present on the subnet.
- Leave Group (Type code = 0x17):
 - Sent by a group member if it was the last member to send a Membership Report to inform the router that it is leaving the group.

Maximum Response Time

Specifies the time limit for the corresponding report.

- The field has a resolution of 100 milliseconds, the value is taken directly.
- only in Membership Query (0x11); in other messages it is set to 0 and ignored by the receiver.

Checksum

Carries the 16-bit checksum computed by the source.

- computed over the entire IP payload, not just over the first 8 octets, even though IGMPv2 messages are only 8 bytes in length.

Group Address

- Set to 0.0.0.0 in General Query messages and to the group address in Group-Specific messages.
- Membership Report messages carry the address of the group being reported in this field
- Leave Group messages carry the address of the group being left in this field.

45.2.2. Messages

- membership queries:
 - general: sent to 224.0.0.1 (all systems on a subnet)
 - group-specific: sent to the group
- membership reports
 - solicited: sent to the group in v2, sent to 224.0.0.22 in v3
 - unsolicited
- Leave Group messages

45.2.3. Default IGMP Configuration

Feature	Default Setting
IGMP version	Version 2 on all interfaces.
IGMP query timeout	60 seconds on all interfaces.
IGMP maximum query response time	10 seconds on all interfaces.
Multilayer switch as a member of a multicast group	No group memberships are defined.
Access to multicast groups	All groups are allowed on an interface.
IGMP host-query message interval	60 seconds on all interfaces.
Multilayer switch as a statically connected member	Disabled.

Task: Display Multicast-Related Information About an Interface.

```
# show ip igmp interface [interface-id]
```

45.2.4. IGMP Version

v1

RFC 1112

- general membership queries
- join
- implicit leave

v2

RFC 2236

- group-specific queries
- explicit leave group process
- explicit max response time field
- querier election
- Backward compatible with v1

v3

RFC 3376

- source filtering SSM
- uses 224.0.0.22 for membership reports
- Backward compatible with v1 and v2

Task: Specify the IGMP Version

```
(config-if)# ip igmp version {1 | 2 | 3}
```

Task: Return to the Default Version

```
(config-if)# no ip igmp version
```



If you change to version 1, you cannot configure the **ip igmp query-interval** or the **ip igmp query-max-response-time** interface configuration commands.

45.2.5. Querier Election

- Selects preferred router to send Query messages when multiple routers are connected on the same subnet
- Each IGMPv2 router sends general query message to 224.0.0.1 with its interface source address.
- The router stops upon reception of query messages with lowest IP address → The router with the lowest IP address wins

45.2.6. IGMPv2 Query Timeout

- period of time before the router takes over as the querier for the interface.
- By default, the router waits twice the query interval. After that time, if the router has received no queries, it becomes the querier.

```
(config-if)# ip igmp querier-timeout <60-300-seconds>
```

45.2.7. Maximum Response Time Field

- v1, fixed at 10 seconds
- v2, can be changed to control the burstiness of the response process especially with large number of active routers.
- Increasing the maximum response timer value also increases the leave latency; the query router must now wait longer to make sure there are no more hosts for the group on the subnet.
- Default:10 seconds, range: 1..25.

Task: Change the Maximum Response Time Field

```
(config-if)# ip igmp query-max-response-time <seconds>
```

45.2.8. Join the Club

Task: Join a Specified Group

```
(config-if)# ip igmp join-group <address>
```

Task: Join a Specified (S,G) Channel

```
(config-if)# ip igmp join-group <address> source <a.b.c.d>
```

Task: Display the Multicast Groups That Are Directly Connected to the Multilayer Switch and That Were Learned Through IGMP.

```
# sh ip igmp groups [group-name | group-address | type number]
```

Task: Forward Multicast Packet Without Accepting Them

```
(config-if)# ip igmp static-group
```



This method allows fast switching. The outgoing interface appears in the IGMP cache, but the switch itself is not a member, as evidenced by lack of an L (local) flag in the multicast route entry.

```
(config-if)# ip igmp static-group
```

45.2.9. Leave Process

- in v1, implicit exit
- in v2,
 - host send leave group message to group address,
 - querier send **igmp-last-member-query-count** group-specific queries at **igmp-last-member-interval** milliseconds
 - querier stops forwarding for the group if no reply within timeout period

Task: Specify the Last Member Query Interval

```
(config-if)# ip igmp last-member-query-interval <milliseconds>
```

Task: Specify the Last Member Query Count

```
(config-if)# ip igmp last-member-query-count <1-7>
```

Task: Minimize the Leave Latency When Only One IGMPv2 Receiver Is Connected to the Interface

```
(config-if)# ip igmp immediate-leave group-list <acl>
```



Can also be in global mode but not combined with the interface mode

45.2.10. IGMP Message Restriction

Task: Restrict Receivers on a Subnet to Join Only Certain Multicast Groups

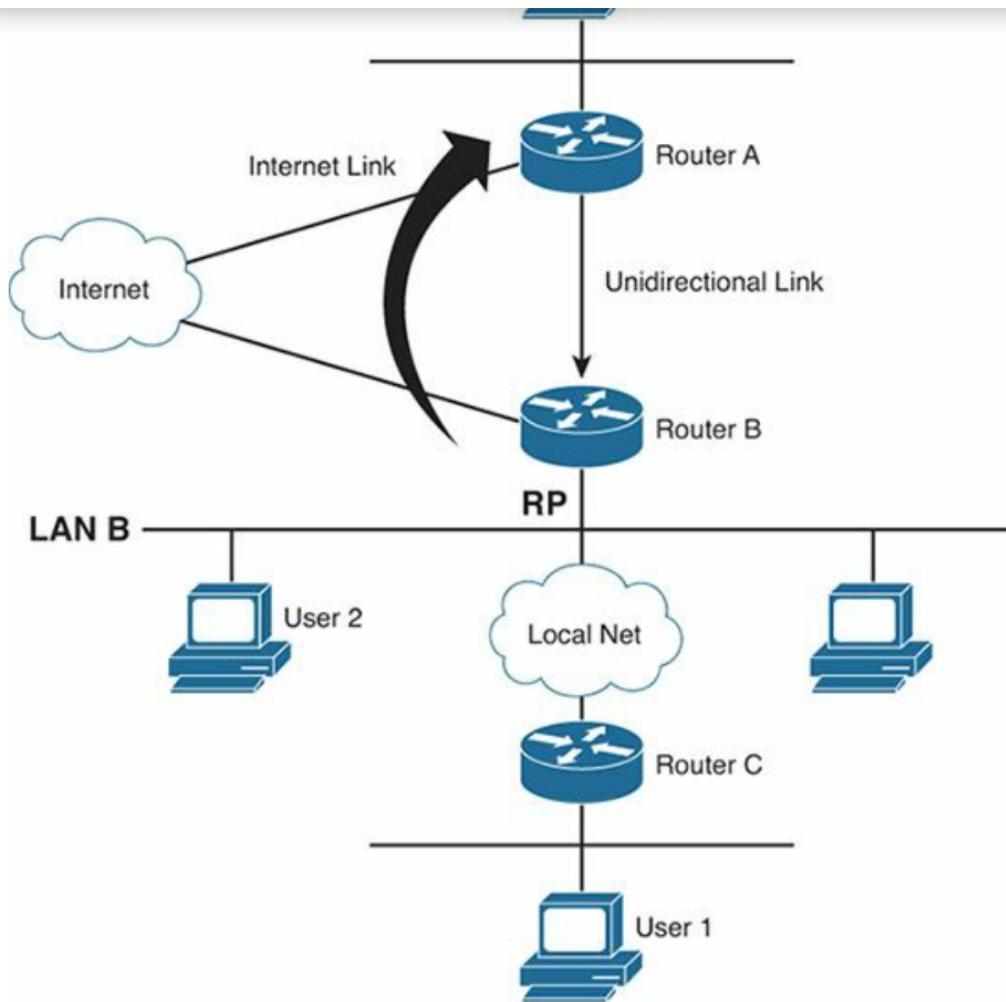
```
(config-if)# ip igmp access-group <standard-acl>
```

Task: Restrict Receivers on a Subnet to Join Multicast Groups from Specific Sources

```
(config-if)# ip igmp access-group <extended-acl>
```

45.2.11. IGMP Proxy

- enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.



- Before you can see how this optimization improves multicast performance, you need to explore what a UDL routing scenario actually is. UDL creates a scenario that would normally be an issue for standard multicast and unicast routing protocols because of the fact that these routing protocols forward data on interfaces from which they have received routing control information. This model works only on bidirectional links for most existing routing protocols like those that we have discussed thus far; however, some networks use broadcast satellite links, which are by their very nature unidirectional. For networks that use broadcast satellite

links, accomplishing two-way communication over broadcast satellite links presents a problem in terms of discovering and sharing knowledge of a network topology through traditional protocols like OSPF or Enhanced Interior Gateway Routing Protocol (EIGRP). This impacts Protocol Independent Multicast (PIM) operation because of PIM reliance on these protocols.

- Specifically, in unicast routing, when a router receives an update message on an interface for a prefix, it forwards data for destinations that match that prefix out that same interface. This is the case in distance vector routing protocols like EIGRP. Similarly, in multicast routing, when a router receives a Join message for a multicast group on an interface, it forwards copies of data destined for that group out that same interface. Based on these principles, existing unicast and multicast routing protocols cannot be supported over UDLs. UDLR was designed to enable the operation of routing protocols over UDLs without changing the routing protocols themselves.
- Read more at “Configuring Unidirectional Link Routing” documentation at Cisco.com or visit the following URL: <http://tinyurl.com/CiscoUDLR>.

45.2.12. CGMP

TODO: Create a separate files for CGMP and IGMP snooping ?

- Cisco proprietary
- Layer 2 protocol, well-known MAC 0X0100.0CDD.DDDD
- configured on both the cisco router and switch.
- permits routers to communicate L2 info it has learned from IGMP to switches
- Only routers send CGMP messages while switches only listens to CGMP messages
- helps switches send group traffic to only those hosts that want it → no wasted bandwidth

Message Format

- GDA: Group Destination Address
- USA: Unicast Source Address

CGMP Messages

,== Type , GDA , USA , Description

Join , Group MAC , Host MAC , Add USA port to group Leave , Group MAC , Host MAC , Delete USA port from group Join , Zero , Router MAC , Learn which port connects to the CGMP router Leave , Zero , Router MAC , Release CGMP router port Leave , Group MAC , Zero , Delete the group from the CAM Leave , Zero , Zero , Delete all groups from the CAM

,==

Task: Clear All the CGMP Entries on the Switches

```
# clear ip cgmp
```

Process

1. When a CGMP-capable router gets connected to the switch, it sends a CGMP Join message with the GDA set to 0 and the USA set to its own MAC address. The CGMP-capable switch now knows that a multicast router is connected to the port on which it received the router's CGMP message. The router repeats the message every 60 seconds. A router can also tell the switch that it no longer participates in CGMP by sending a CGMP Leave message with the GDA set to 0 and the USA set to its own MAC address.
2. When a host joins a group, it sends an IGMP Join message. Normally, a multicast router examines only Layer 3 information in the IGMP Join message, and the router does not have to process any Layer 2 information. However, when CGMP is configured on a router, the router also examines the Layer 2 destination and source MAC addresses of the IGMP Join message. The source address is the unicast MAC address of the host that sent the IGMP Join message. The router then generates a CGMP Join message that includes the multicast MAC address associated with the multicast IP address (to the GDA field of the CGMP join) and the unicast MAC address of the host (to the USA field of the CGMP message). The router sends the CGMP Join message using the well-known CGMP multicast MAC address 0x0100.0cdd.dddd as the destination address.
3. When switches receive a CGMP Join message, they search in their CAM tables for the port number associated with the host MAC address listed in the USA field. Switches create a new CAM table entry (or use an existing entry if it was already created before) for the multicast MAC address listed in the GDA field of the CGMP Join message, add the port number associated with the host MAC address listed in the USA field to the entry, and forward the group traffic on the port.
4. When a host leaves a group, it sends an IGMP Leave message. The router learns the host's unicast MAC address (USA) and the IP multicast group it has just left. Because the Leave messages are sent to the All Multicast Routers MAC address 0x0100.5e00.0002 and not to the multicast group address the host has just left, the router calculates the multicast MAC address (GDA) from the IP multicast group the host has just left. The router then generates a CGMP Leave message, copies the multicast MAC address it has just calculated in the GDA field and unicast MAC address in the USA field of the CGMP Leave message, and sends it to the well-known CGMP multicast MAC address.
5. When switches receive a CGMP Leave message, they again search for the port number associated with the host MAC address listed in the USA field. Switches remove this port from the CAM table entry for the multicast MAC address listed in the GDA field of the CGMP Leave message and stop forwarding the group traffic on the port.

45.2.13. RGMP

- Router-Port Group Management Protocol
- Proprietary with informational RFC 3488
- Layer 2
- doesn't work with CGMP
 - IGMP snooping helps switches control distribution of multicast traffic on ports where multicast hosts are connected, but it does not help switches control distribution of multicast
- works well with ICMP snooping

traffic on ports where multicast routers are connected.

Operations

- When RGMP is enabled on a router, the router sends RGMP Hello messages by default every 30 seconds. When the switch receives an RGMP Hello message, it stops forwarding all multicast traffic on the port on which it received the Hello message.
- When the router wants to receive traffic for a specific multicast group, the router sends an RGMP Join G message, where G is the multicast group address, to the switch. When the switch receives an RGMP Join message, it starts forwarding the requested group traffic on the port on which it received the Hello message.
- When the router does not want to receive traffic for a formerly RGMP-joined specific multicast group, the router sends an RGMP Leave G message, where G is the multicast group address, to the switch. When the switch receives an RGMP Leave message, it stops forwarding the group traffic on the port on which it received the Hello message.
- When RGMP is disabled on the router, the router sends an RGMP Bye message to the switch. When the switch receives an RGMP Bye message, it starts forwarding all IP multicast traffic on the port on which it received the Hello message.

Task: Enable RGMP

```
(config-if)# ip rgmp
```

45.2.14. IGMP Filtering and Throttling

- configures and apply IGMP profiles on a SVI, a per-port, or a per-port per-VLAN basis
- works only with IGMP snooping active globally or on the port
- When an IGMP packet is received, IGMP filtering uses the filters configured by the user to determine whether the IGMP packet should be discarded or allowed to be processed by the existing IGMP snooping code.
 - With IGMP v1/v2, the entire packet is discarded.
 - With IGMPv3 , the packet is rewritten to remove message elements that were denied by the filters.

Task: Display IGMP Filtering and Throttling Configuration

```
# sh ip igmp profile [<number>]
```

IGMP Profiles

Task: Configure IGMP Profile

```
(config)# ip igmp profile <number>
(config-profile)# {permit | deny}
(config-profile)# range <low-mcast-ip-address> [<high-mcast-ip-address>]
```

Task: Apply IGMP Profile

```
(config-if)# ip igmp filter <profile-number>
```



- applicable only to Layer 2 access ports (no routed ports, SVIs, or physical port belonging to EtherChannel);
- only one profile per interface

Maximum Number Of IGMP Groups

Task: Set the Maximum Number Of IGMP Groups

```
(config-if)# ip igmp max-groups <number>
```

IGMP Throttling Action

- When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, specify the action that the interface takes:
 - deny : Drop the report.
 - replace : Replace the existing group with the new group for which the IGMP report was received.
- works only if the maximum number of IGMP groups have been configured
 - If the throttling action is **deny**,
 - the entries that were previously in the forwarding table are aged out.
 - After these entries are aged out and the maximum number of entries is in the forwarding table, the switch drops the next IGMP report received on the interface.
 - If the throttling action is **replace**,
 - the entries that were previously in the forwarding table are removed.
 - When the maximum number of entries is in the forwarding table, the switch replaces a randomly selected entry with the received IGMP report.
 - To prevent the switch from removing the forwarding-table entries, configure the IGMP throttling action before an interface adds entries to the forwarding table.

Task: Configure IGMP Throttling Action

```
(config-if)# ip igmp max-groups action {deny | replace}
```

45.2.15. IGMP Snooping

- Problem: L2 switch forwards multicast packets to all interfaces → wasted traffic
- Solution: Tracks IGMP messages (Join/Leave) to only forward invites to interested parties.

- Add ports when receiving Join message
- Delete ports when Leave messages or no membership reports from clients

Table 21. Default IGMP Snooping Configuration

Feature	Default Setting
IGMP snooping	Enabled globally and per VLAN
Multicast routers	None configured
Multicast router learning method	PIM-DVMRP
IGMP snooping Immediate Leave	Disabled
Static groups	None configured
TCN flood query count	2
TCN query solicitation	Disabled
IGMP snooping querier	Disabled
IGMP report suppression	Enabled

Task: Display IGMP Snooping Information

```
# sh ip igmp snooping
```

Task: Disable IGMP Snooping Globally

```
(config)# no ip igmp snooping
```

Task: Enable VLAN Snooping

```
(config)# ip igmp snooping vlan <1-1001,1006-4094>
```

Task: Change the Snooping Method

```
(config)# ip igmp snooping vlan <vlan-id> mrouter learn {cgmp | pim-dvmrp}
```

45.2.16. Multicast Router Port

Task: Add a Multicast Router Port

```
(config)# ip igmp snooping vlan <id> mrouter interface <type-number>
```

Task: Verify That IGMP Snooping Is Enabled on the VLAN Interface

```
(config)# sh ip igmp snooping mrouter vlan <id>
```

Statically Join a Group

Task: Add a L2 Port to Join a Group

```
ip igmp snooping vlan <vlan-id> static <ip-address> interface <type number>
```



Hosts or L2 ports normally join multicast groups dynamically

Task: Verify the Member Port and the IP Address

```
# sh ip igmp snooping groups
```

IGMP Immediate Leave

Task: Remove a Port Immediately When It Detects an IGMPv2 Leave Message

```
(config)# ip igmp snooping vlan <id> immediate-leave
```

IGMP Leave Timer

Task: Configure the IGMP Leave Timer Globally

```
ip igmp snooping last-member-query-interval <milliseconds>
```

Task: Configure the IGMP Leave Timer on the VLAN Interface

```
ip igmp snooping vlan <id> last-member-query-interval <milliseconds>
```

TCN Events

- when the client changed its location and the receiver is on same port that was blocked but is now forwarding,
- when a port went down without sending a leave message.

Task: Control the Multicast Flooding Time After a TCN Event

```
(config)# ip igmp snooping tcn flood query count <1-2-10>
```

Task: Speed the Process Of Recovering from the Flood Mode Caused by a TCN Event.

```
(config)# ip igmp snooping tcn query solicit
```



- When a topology change occurs, the spanning-tree root sends a IGMP global leave with group 0.0.0.0.
- however, after **ip igmp snooping tcn query solicit** command, the switch sneds the global leave message whether or not it is the spanning-tree root.
- When the router receives this special leave, it immediately sends general queries, which expedite the process of recovering from the flood mode during the TCN event.

Task: Disable the Flooding Of Multicast Traffic During a TCN Event

```
(config-if)# no ip igmp snooping tcn flood
```



- When the swith receives a TCN, multicast traffic is flooded to all the ports until 2 general queries are received.
- If the switch has many ports with attached hosts subscribed to many groups, this flooding might exceed the capacity of the link and cause packet loss.

IGMP Snooping Querier

Task: Enable IGMP Snooping Querier

```
(config)# ip igmp snooping querier
(config)# ip igmp snooping querier address <ip.ad.re.ss>
(config)# ip igmp snooping querier query-interval <seconds>
(config)# ip igmp snooping querier tcn query [count <n> | interval <seconds>]
(config)# ip igmp snooping querier timer expiry <seconds>
(config)# ip igmp snooping querier version {1 | 2}
```

Task: Display Information About the IP Address and Receiving Port for the Most-Recently Received IGMP Query In the VLAN

```
# sh ip igmp snooping querier [vlan <id>] [detail]
```

IGMP Report Suppression

Task: Disable IGMP Report Suppression

```
(config)# no ip igmp snooping report-suppression
```

45.2.17. MVR

- Multicast VLAN Registration
- Problem: How to scale multicast traffic accross an Ethernet ring-based SP network
- Solution : one multicast VLAN shared with subscribers in seperate VLANs

- Use case: broadcast of multiple TV channels over a service-provider network
- works with or without IGMP snooping
 - If both enabled, MVR reacts only to join and leave messages from MVR groups.

Table 22. Default MVR Configuration

Feature	Default Setting
MVR	Disabled globally and per interface
Multicast addresses	None configured
Query response time	0.5 second
Multicast VLAN	VLAN 1
Mode	Compatible
Interface (per port) default	Neither a receiver nor a source port
Immediate Leave	Disabled on all ports

MVR Global Parameters

Task: Enable MVR on the Switch

```
(config)# mvr
```

Task: Configure a Range Of IP Multicast Address on the Switch

```
(config)# mvr group <ip-address> [count]
```



- The **count** parameter configure a contiguous series of MVR group addresses. Default= 1 in 1..256
- Any multicast data sent to the ip address corresponding to one TV channel is sent to all source ports on the switch and all interested receiver ports.

Task: Define the Maximum Time to Wait for IGMP Report Memberships on a Receiver Port

```
(config)# mvr querytime <tenths-of-seconds>
```

Task: Specify the VLAN In Which Multicast Data Is Received

```
(config)# mvr vlan <vlan-id>
```



- All source ports must belong to this VLAN

Task: Specify the MVR Mode Of Operation

```
(config)# mvr mode { dynamic | compatible }
```



- **dynamic**: allows dynamic MVR memberships on source ports.
- **compatible** is the default and does not support ICMP dynamic joins on source ports.

Task: Verify the MVR Global Configuration

```
(config)# sh mvr  
(config)# sh mvr members
```

MVR Interfaces

Task: Configure an MVR Port As Source

```
(config-if)# mvr type source
```



- Configure uplinks ports that receive and send multicast data as source ports
- Subscribers cannot be directly connected to source ports
- All source ports on a switch belong to the single multicast VLAN.

Task: Configure an MVR Port As Receiver

```
(config-if)# mvr type receiver
```



- Configure a port as a receiver port if it is a subscriber port and should only receive multicast data.
- Receiver ports do not receive data unless it becomes a member of the multicast group.
- Receiver ports cannot belong to the multicast VLAN.

Task: Statically Configure a Port to Receive Multicast Traffic

```
(config)# mvr vlan <id> group <ip-address>
```

Task: Enable the Immediate-Leave Feature Of MVR on the Receiver Port

```
(config)# mvr immediate
```

Task: Verify the MVR Interface Configuration

```
# sh mvr interface
```

Task: Display All Receiver and Source Ports That Are Members Of a Multicast Group

```
# sh mvr members [group-ip-address]
```

45.2.18. IGMP Filtering and Throttling

Table 23. Default IGMP Filtering Configuration

Feature	Default Setting
Filters	none applied
profiles	none defined
profile action	deny the range addresses

IGMP Profiles

Task: Configure an IGMP Profile

```
(config)# ip igmp profile <number>
(config-igmp-profile)# permit | deny
(config-igmp-profile)# range <low-ip-address> [<high-ip-address>]
```

Task: Apply IGMP Profile to an Interface

```
(config)# ip igmp filter <profile-number>
```

Task: Verify the Profile Configuration

```
# sh ip igmp profile <number>
```

IGMP Throttling

Task: Set the Maximum Number Of IGMP Groups That the Interface Can Join

```
(config-if)# ip igmp max-groups <count>
```

Task: Specify the Action That the Interface Takes When It Reaches the Maximum Number Of Entries and Receives a New IGMP Report

```
(config-if)# ip igmp max-groups action {deny | replace }
```

45.3. PIM

- Protocol-independent
 - relies on unicast routing table to perform RPF check
 - needs **ip multicast-routing** command

45.3.1. Versions

- Default= v2 since IOS 11.3

Task: Configure the PIM Version on the Interface.

```
(config)# ip multicast-routing  
(config-if)# ip pim version [1|2]
```

Task: Display Information About Interfaces Configured for PIM.

```
show ip pim interface [type number] [count]
```

PIMv1

- Supports Auto-RP : eliminates the need to manually configure the rendezvous point in every router.
 - multiple active RPs for the same group

PIMv2

- Supports BSR (boot strap router) capability.
- single, active RP per multicast group, with multiple backup RPs.
- Sparse mode and dense mode are properties of a group, as opposed to an interface.
- PIM join and prune messages have more flexible encoding for multiple address families.
- A more flexible hello packet format replaces the query packet to encode current and future capability options.
- Register messages to an RP specify whether they are sent by a border router or a designated router.
- PIM packets are no longer inside IGMP packets; they are standalone packets.

PIMv1 and PIMv2 Interoperability

- You can upgrade to PIMv2 incrementally. PIM Versions 1 and 2 can be configured on different routers within one network. Internally, all routers on a shared media network must run the same PIM version. Therefore, if a PIMv2 device detects a PIMv1 device, the Version 2 device downgrades itself to Version 1 until all Version 1 devices have been shut down or upgraded.
- PIMv2 uses the BSR to discover and announce RP-set information for each group prefix to all

the routers in a PIM domain. PIMv1, together with the Auto-RP feature, can perform the same tasks as the PIMv2 BSR.

- When PIMv2 devices interoperate with PIMv1 devices, Auto-RP should have already been deployed. A PIMv2 BSR that is also an Auto-RP mapping agent automatically advertises the RP elected by Auto-RP. That is, Auto-RP sets its single RP on every router in the group. Not all routers and switches in the domain use the PIMv2 hash function to select multiple RPs.
- Dense-mode groups in a mixed PIMv1 and PIMv2 region need no special configuration; they automatically interoperate.
- Sparse-mode groups in a mixed PIMv1 and PIMv2 region are possible because the Auto-RP feature in PIMv1 interoperates with the PIMv2 RP feature. Although all PIMv2 devices can also use PIMv1, we recommend that the RPs be upgraded to PIMv2 (or at least upgraded to PIMv1 in the Cisco IOS Release 11.3 software).

To ease the transition to PIMv2, we have these recommendations:

- Use Auto-RP throughout the region.
- Configure sparse-dense mode throughout the region.

45.3.2. Modes

- dense mode (DM)
- sparse mode (SM)
- sparse-dense mode (PIM DM-SM) : recommended
- By default, no mode is configured.

Task: Enable a PIM Mode on the Interface.

```
(config-if)# pim {dense-mode | sparse-mode | sparse-dense-mode }
```

PIM DM

- employs only SPTs to deliver (S,G) multicast traffic by using a implicit flood and explicit prune method.
 - A separate SPT exists for every individual source sending to each group.
 - (S,G) identifies an SPT where S is the IP address of the source and G is the multicast group address.
- sends prune message to upstream when there are no directly connected members or PIM neighbors present
 - Prunes have a timeout value associated with them, after which the PIM DM device puts the interface into the forwarding state and floods multicast traffic out the interface.
 - sends graft message when a new receiver on a previously pruned branch joins a multicast group

PIM SM

- uses shared trees and SPTs to distribute multicast traffic to multicast receivers in the network.
- needs explicit join towards the RP (Rendez-vous Point)
 - When a host joins a multicast group using IGMP, its directly connected PIM SM device sends PIM join messages toward the RP.
 - This join message travels router-by-router toward the root, constructing a branch of the shared tree as it goes.
 - The RP keeps track of multicast receivers; it also registers sources through register messages received from the source's first-hop router (designated router DR) to complete the shared tree path from the source to the receiver. The branches of the shared tree are maintained by periodic join refresh messages that the PIM SM devices send along the branch.
 - When using a shared tree, sources must send their traffic to the RP so that the traffic reaches all receivers. The special notation \ast,G , (pronounced star comma G) is used to represent the tree, where \ast means all sources and G represents the multicast group.



In addition to using the shared distribution tree, PIM SM can also use SPTs. By joining an SPT, multicast traffic is routed directly to the receivers without having to go through the RP, thereby reducing network latency and possible congestion at the RP. The disadvantage is that PIM SM devices must create and maintain (S,G) state entries in their routing tables along with the (S,G) SPT. This action consumes router resources.

- Prune messages are sent up the distribution tree to prune multicast group traffic. This action permits branches of the shared tree or SPT that were created with explicit join messages to be torn down when they are no longer needed. For example, if a leaf router (a router without any downstream connections) detects that it no longer has any directly connected hosts (or downstream multicast routers) for a particular multicast group, it sends a prune message up the distribution tree to stop the flow of unwanted multicast traffic.

Shared Tree Vs Source Tree

By default, members of a group receive data from senders to the group across a single data-distribution tree rooted at the RP. Figure [PIM Trees](#) shows this type of shared-distribution tree. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

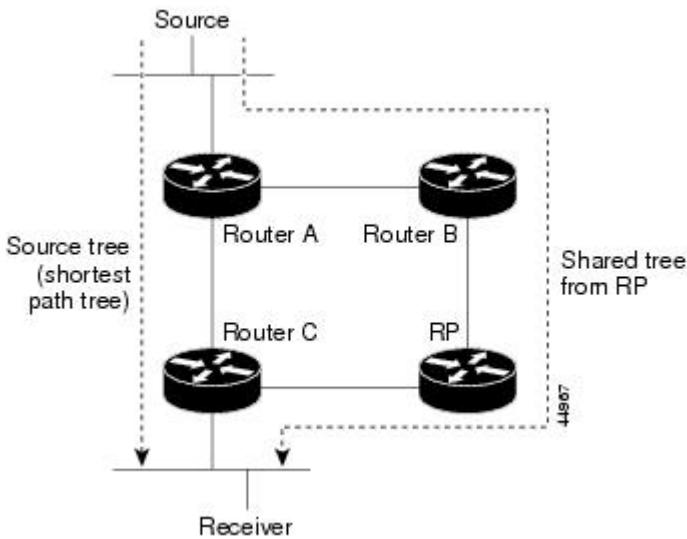


Figure 53. PIM Trees

If the data rate warrants, leaf routers (routers without any downstream connections) on the shared tree can use the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree or source tree. By default, the IOS software switches to a source tree upon receiving the first data packet from a source.

This process describes the move from a shared tree to a source tree:

1. a Receiver Joins a Group; Leaf Router C Sends a Join Message Toward the RP.
2. the RP Puts a Link to Router C In Its Outgoing Interface List.
3. a Source Sends Data; Router a Encapsulates the Data In a Register Message and sends it to the RP.
4. the RP Forwards the Data Down the Shared Tree to Router C and Sends a Join message toward the source. At this point, data might arrive twice at Router C, once encapsulated and once natively.
5. When Data Arrives Natively (Unencapsulated) at the RP, It Sends A register-stop message to Router A.
6. by Default, Reception Of the First Data Packet Prompts Router C to Send A join message toward the source.
7. When Router C Receives Data on (S,G), It Sends a Prune Message for The source up the shared tree.
8. the RP Deletes the Link to Router C from the Outgoing Interface Of (S,G). The RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM device along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router that is directly connected to a source and are received by the RP for the group.

Multiple sources sending to groups use the shared tree.

You can configure the PIM device to stay on the shared tree.

Sparse-Dense Mode

TODO

45.3.3. PIM Designated Routers

- Senders of multicast traffic announce their existence through register messages received from the source's first-hop router (designated router) and forwarded to the RP.

PIM routers send PIM router-query messages to determine which device will be the DR for each LAN segment (subnet). The DR is responsible for sending IGMP host-query messages to all hosts on the directly connected LAN.

With PIM DM operation, the DR has meaning only if IGMPv1 is in use. IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier. With PIM SM operation, the DR is the device that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree. In this case, the DR is the device with the highest IP address.

The default is 30 seconds. The range is 1 to 65535.

Task: Configure the PIM Query Interval

```
ip pim query-interval <seconds>
```

45.3.4. Rendez-Vous Points

- Receivers of multicast packets use RPs to join a multicast group by using explicit join messages.
- RPs are not members of the multicast group; rather, they serve as a meeting place for multicast sources and group members.
- By default, no PIM RP address is configured.
 - You must configure the IP address of RPs on all routers (including the RP).
 - If there is no RP configured for a group, the multilayer switch treats the group as dense, using the dense-mode PIM techniques.
 - A PIM device can use multiple RPs, but only one per group.

Task: Configure Static RP Address

```
ip pim rp-address <RP-IP-address> [<standard-access-list-number>] [override]
```



- For **access-list-number**, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
- The **override** keyword means that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.

Task: Display the RP That Was Selected for the Specified Group.

```
# sh ip pim rp-hash group
```

Task: Display How the Router Learns Of the RP (Through the BSR or the Auto-RP Mechanism).

```
# sh ip pim rp [ <group-name> | <group-address> | <mapping> ]
```

Task: Display the RP Routers Associated with a Sparse-Mode Multicast Group.

```
# sh ip pim rp [ <group-name> | <group-address>]
```

45.3.5. Auto-RP

- Cisco proprietary feature
- eliminates the need to manually configure the rendezvous point (RP) information in every router and multilayer switch in the network.
- uses IP multicast to automate the distribution of group-to-RP mappings to all Cisco routers in a PIM network.
- Benefits
 - multiple RPs within a network to serve different group ranges.
 - Multiple RPs serve different group ranges or serve as hot backups of each other.
 - load splitting among different RPs and arrangement of RPs according to the location of group participants.
 - no inconsistent and manual RP configurations on every router and multilayer switch

Candidate RP

- Send multicast RP-announce messages to 224.0.1.39 every 60 seconds (default) with holdtime of 180 seconds (default)

Task: Configure Another PIM Device to Be the Candidate RP for Local Groups.

```
ip pim send-rp-announce <interface-id> scope <ttl> group-list <access-list-number>
interval <seconds>
```



- For **interface-id**, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs.
- For scope **ttl**, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255.
- For group-list **access-list-number**, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
- For interval **seconds**, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.

Mapping Agents

- listen to RP-announce messages
- create Group-to-RP mapping cache
- select highest IP candidate as active RP
- send Group-to-RP mapping cache in RP-discovery messages to 224.0.1.40 every 60 seconds with 180 seconds holdtime

Task: Configure RP Mapping Agent

```
(config)# ip pim send-rp-discovery scope <1..255>
```

Task: Configure PIM-SM Interfaces to Use Dense Mode to Flood Auto-RP Traffic to 224.0.1.39 and 224.0.1.40.

```
(config)# ip pim autorp listener
```

Task: Prevent Candidate RP Spoofing

```
ip pim rp-announce-filter rp-list <access-list-number> group-list <access-list-number>
```



- Enter this command on each mapping agent in the network.
- Without this command, all incoming RP-announce messages are accepted by default.
- For **rp-list** access-list-number, configure an access list of candidate RP addresses that, if permitted, is accepted for the group ranges supplied in the group-list access-list-number variable. If this variable is omitted, the filter applies to all multicast groups.
- If more than one mapping agent is used, the filters must be consistent across all mapping agents to ensure that no conflicts occur in the Group-to-RP mapping information.

Task: Prevent Join Messages to False RPs

```
Switch(config)# ip pim accept-rp 172.10.20.1 1
Switch(config)# access-list 1 permit 224.0.1.39
Switch(config)# access-list 1 permit 224.0.1.40
```

Determine whether the **ip pim accept-rp** command was previously configured throughout the network by using the **show running-config** privileged EXEC command. If the **ip pim accept-rp** command is not configured on any device, this problem can be addressed later. In those routers es already configured with the **ip pim accept-rp** command, you must enter the command again to accept the newly advertised RP.



To accept all RPs advertised with Auto-RP and reject all other RPs by default, use the **ip pim accept-rp auto-rp** global configuration command.

If all interfaces are in sparse mode, use a default-configured RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP uses these two well-known groups to collect and distribute RP-mapping information. When this is the case and the **ip pim accept-rp auto-rp** command is configured, another **ip pim accept-rp** command accepting the RP must be configured as follows:

PIM Routers

- listen to RP-discovery messages
- knows with RP to use for groups they support
- if Group-to-RP expires, select statically configured RP or switch to dense-mode operation

45.3.6. Bootstrap Router

- A BSR provides a fault-tolerant, automated RP discovery and distribution mechanism that enables routers to dynamically learn the group-to-RP mappings.
- eliminates the need to manually configure RP information in every router and switch in the network.
- uses hop-by-hop flooding of BSR messages to distribute the mapping information
 - Each router multicasts BSR messages with TTL=1 to all PIM interfaces except the one on which it was received.
 - BSR contains the IP address of the current BSR
- elected BSR based on highest (priority, IP address)
- Candidate RPs send candidate RP advertisements showing the group range for which they are responsible directly to the BSR, which stores this information in its local candidate-RP cache. The BSR periodically advertises the contents of this cache in BSR messages to all other PIM devices in the domain. These messages travel hop-by-hop through the network to all routers and switches, which store the RP information in the BSR message in their local RP cache. The routers and switches select the same RP for a given group because they all use a common RP

hashing algorithm.

Task: Display the Elected BSR

```
# sh ip pim bsr
```

Candidate BSRs

You can configure one or more candidate BSRs. The devices serving as candidate BSRs should have good connectivity to other devices and be in the backbone portion of the network.

Task: Configure Your Multilayer Switch to Be a Candidate BSR.

```
(config)# ip pim bsr-candidate <interface-id> <hash-mask-length> [priority]
```

- For **interface-id**, enter the interface type and number on this switch from which the BSR address is derived to make it a candidate. This interface must be enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs.
- For **hash-mask-length**, specify the mask length (32 bits maximum) that is to be ANDed with the group address before the hash function is called. All groups with the same seed hash correspond to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter.
- For **priority**, enter a number from 0 to 255. The BSR with the larger priority is preferred. If the priority values are the same, the device with the highest IP address is selected as the BSR. The default is 0.



Multicast Forwarding and Reverse Path Check

With multicasting, the source is sending traffic to an arbitrary group of hosts represented by a multicast group address in the destination address field of the IP packet. To determine whether to forward or drop an incoming multicast packet, the router uses a **reverse path forwarding** (RPF) check on the packet as follows and shown in Figure [RPF Check](#):

1. the Router Examines the Source Address Of the Arriving multicast packet to determine whether the packet arrived on an interface that is on the reverse path back to the source.
2. If the Packet Arrives on the Interface Leading Back to the Source, the RPF check is successful and the packet is forwarded to all interfaces in the outgoing interface list (which might not be all interfaces on the router).
3. If the RPF Check Fails, the Packet Is Discarded.

Some multicast routing protocols, such as DVMRP, maintain a separate multicast routing table and use it for the RPF check. However, PIM uses the unicast routing table to perform the RPF check.

Figure [RPF Check](#) shows Gigabit Ethernet interface 0/2 receiving a multicast packet from source 151.10.3.21. A check of the routing table shows that the interface on the reverse path to the source is Gigabit Ethernet interface 0/1, not interface 0/2. Because the RPF check fails, the multilayer switch

discards the packet. Another multicast packet from source 151.10.3.21 is received on interface 0/1, and the routing table shows this interface is on the reverse path to the source. Because the RPF check passes, the switch forwards the packet to all interfaces in the outgoing interface list.

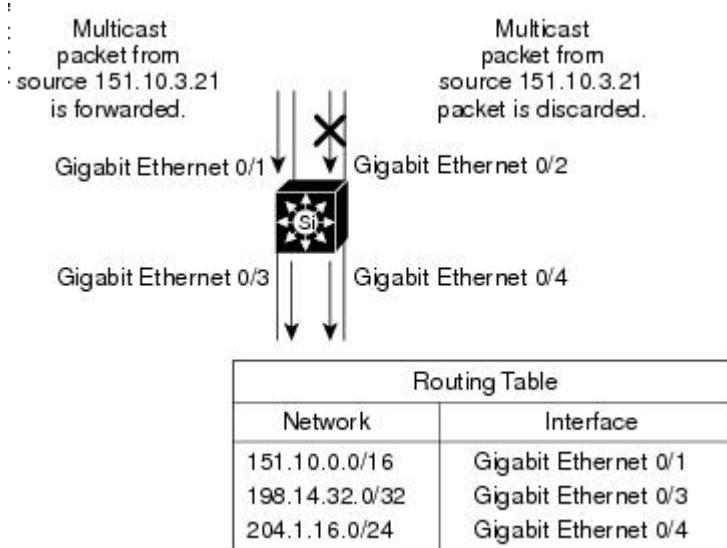


Figure 54. RPF Check

PIM uses both source trees and RP-rooted shared trees to forward datagrams ; the RPF check is performed differently for each:

- If a PIM router has a source-tree state (an (S,G) entry is present in the multicast routing table), it performs the RPF check against the IP address of the source of the multicast packet.
- If a PIM router has a shared-tree state (and no explicit source-tree state), it performs the RPF check on the rendezvous point (RP) address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes:

- (S,G) joins (which are source-tree states) are sent toward the source.
- (*,G) joins (which are shared-tree states) are sent toward the RP.

DVMRP and dense-mode PIM use only source trees and use RPF as previously described.

Task: Display How the Multilayer Switch Is Doing Reverse-Path Forwarding

```
# sh ip rpf { <source-address> | <name>}
```

45.3.7. Neighbor Discovery

- To establish adjacencies, a PIM router sends PIM hello messages to the all-PIM-routers multicast group (224.0.0.13) on each of its multicast-enabled interfaces.
 - The hello message contains a holdtime, which tells the receiver when the neighbor adjacency associated with the sender expires if no more PIM hello messages are received.
 - Keeping track of adjacencies is important for PIM DM operation for building the source distribution tree.

- PIM hello messages are also used to elect the DR (highest IP address) for multi-access networks
 - With PIM DM operation, the DR has meaning only if IGMPv1 is in use; IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier.
 - In PIM SM operation, the DR is the router or switch that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree.

Task: List the PIM Neighbors

```
# sh ip pim neighbor [type number]
```

Task: Query a Multicast Router About Which Neighboring Multicast Devices Are Peering with It.

```
# mrinfo [ <hostname> | <address>] [ <source-address> | <interface>]
```

Task: Display IP Multicast Packet Rate and Loss Information.

```
# mstat source [<destination>] [<group>]
```

Task: Trace the Path from a Source to a Destination Branch for a Multicast Distribution Tree for a Given Group.

```
# mtrace source [<destination>] [<group>]
```

45.3.8. Auto-RP and BSR Configuration Guidelines

There are two approaches to using PIMv2. You can use Version 2 exclusively in your network or migrate to Version 2 by employing a mixed PIM version environment.

- If your network is all Cisco routers , you can use either Auto-RP or BSR.
- If you have non-Cisco routers in your network, you must use BSR.
- If you have Cisco PIMv1 and PIMv2 routers and non-Cisco routers, you must use both Auto-RP and BSR.
- Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents these messages from reaching all routers in your network. Therefore, if your network has a PIMv1 device in it and only Cisco routers and multilayer switches, it is best to use Auto-RP.
- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router . Ensure that no PIMv1 device is on the path between the BSR and a non-Cisco PIMv2 router.
- If you have non-Cisco PIMv2 routers that need to interoperate with Cisco PIMv1 routers , both Auto-RP and a BSR are required. We recommend that a Cisco PIMv2 device be both the Auto-RP mapping agent and the BSR.

45.3.9. PIM Domain Border

As IP multicast becomes more widespread, the chances of one PIMv2 domain bordering another PIMv2 domain is increasing. Because these two domains probably do not share the same set of RPs, BSR, candidate RPs, and candidate BSRs, you need to constrain PIMv2 BSR messages from flowing into or out of the domain. Allowing these messages to leak across the domain borders could adversely affect the normal BSR election mechanism and elect a single BSR across all bordering domains and co-mingle candidate RP advertisements, resulting in the election of RPs in the wrong domain.

Task: Define a PIM Bootstrap Message Boundary for the PIM Domain.

```
(config-if)# ip pim bsr-border
```

Administratively-Scaled Boundary

- uses range 239.0.0.0 to 239.255.255.255

Task: Configure an Administratively Scope Boundary

```
(config-if)# ip multicast boundary <access-list-number> [filter-autorp]
```

TTL Scoping

- The default TTL value is 0 hops, which means that all multicast packets are forwarded out the interface. The range is 0 to 255.
- Only multicast packets with a TTL value greater than the threshold are forwarded out the interface.
- You should configure the TTL threshold only on routed interfaces at the perimeter of the network.

Task: Configure TTL Scoping

```
(config-if)# ip multicast ttl-threshold <value>
```

45.3.10. Delay the Use Of PIM Shortest-Path Tree

The change from shared to source tree happens when the first data packet arrives at the last-hop router. This change occurs because the **ip pim spt-threshold** interface configuration command controls that timing; its default setting is 0 kbps.

The shortest-path tree requires more memory than the shared tree but reduces delay. You might want to postpone its use. Instead of allowing the leaf router to immediately move to the shortest-path tree, you can specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router should join the shortest-path tree for a specified group. If a source sends at a rate greater than or equal to the specified kbps rate, the multilayer switch

triggers a PIM join message toward the source to construct a source tree (shortest-path tree). If the traffic rate from the source drops below the threshold value, the leaf router switches back to the shared tree and sends a prune message toward the source.

You can specify to which groups the shortest-path tree threshold applies by using a group list (a standard access list). If a value of 0 is specified or if the group list is not used, the threshold applies to all groups.

Task: Specify the Threshold That Must Be Reached Before Moving to Shortest-Path Tree

```
ip pim spt-threshold {kbps | infinity} [group-list access-list-number]
```

- For **kbps**, specify the traffic rate in kilobits per second. The default is 0 kbps. The range is 0 to 4294967.
- Specify **infinity** if you want all sources for the specified group to use the shared tree, never switching to the source tree.
- For group-list **access-list-number**, specify the access list created in Step 2. If the value is 0 or if the group-list is not used, the threshold applies to all groups.



45.3.11. Troubleshoot

When debugging interoperability problems between PIMv1 and PIMv2, check these in the order shown:

1. Verify RP Mapping with **Sh Ip Pim Rp-Hash** Making Sure That All Systems Agree on the Same RP for the Same Group.
2. Verify Interoperability Between Different Versions Of DRs and RPs. Make Sure the RPs are interacting with the DRs properly (by responding with register-stops and forwarding decapsulated data packets from registers).

[Load splitting IP multicast traffic over ECMP](#)

45.3.12. Misc

TODO To be added in the text

Table 24. PIM Type Code

Type	Name
0	Hello
1	Register
2	Register Stop
3	Join/Prune
4	Bootstrap
5	Assert
6	Graft

Type	Name
7	Graft-Ack
8	Candidate RP Advertisement
9	State Refresh
10	DF Election
11-14	Unassigned
15	Reserved for extension of type space

SSM

Task: Define the Ssm Range Of IP Multicast Addresses

```
(config)# ip pim [vrf <name>] ssm { default | range a<cccess-list-number> }
```



default defines the ssm range access list to 232/8

45.3.13. PIM snooping

TODO

45.3.14. PIM stub routing

TODO In a network using PIM stub routing, the only allowable route for IP traffic to the user is through a switch that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets

Further Reading <http://goo.gl/UDQbL2>

45.4. MLD

TODO

- IP protocol: 58
- ipv6
- hop limit= 1
- mldv1 ← igmpv2, mldv2 ← igmpv3
- enabled globally when
 - pimv6 enabled
 - statically bind a local mcast group
 - link-local group report enabled

45.4.1. MLD Snooping

Task: Enable pimv6 snooping

```
# sh ipv6 snooping
```

TODO

- ND Inspection
- RA Guard

Chapter 46. Network Optimization

46.1. IP SLA

- Configuration Guides > Network Mgt > **IP SLA**

To implement IP SLAs network performance measurement,

1. Enable the IP SLAs Responder
2. Configure the Required IP SLAs Operation Type
3. Configure Any Options Available
4. Configure Threshold Conditions
5. Schedule the Operation
6. Run the Operation for a Period Of Time to Gather Statistics
7. Display and Interpret the Results with Cisco CLI or NMS with SNMP

46.1.1. IP SLAs Operation Types

- ICMP echo, jitter, path echo , path jitter
- TCP connect
- UDP echo, jitter
- VoIP RTP, UDP jitter, gatekeeper registration delay, post-dial delay
- HTTP, FTP, DNS, DHCP

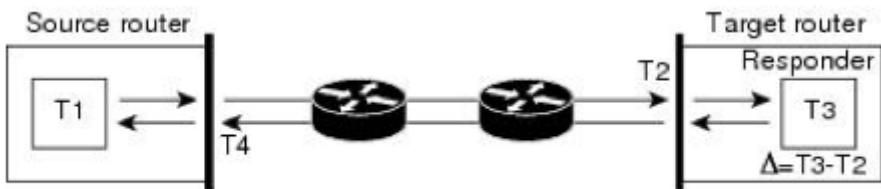
Task: Configure Basic IP SLAs ICMP Echo Operation on the Source Device

```
(config)# ip sla <operation-number>
(config-ip-sla)# icmp-echo {<destination-ip-address>| <destination-hostname>}
[<source-ip {<src-ip> | src-hostname} | source-interface
<interface-name>]
(config-ip-sla-echo)# frequency <seconds>
```

46.1.2. IP SLAs Responder

- for Cisco device only
- listens on specific port for control protocol messages sent by IP SLA operation
- on receipt of the control, open specified TCP or UDP port for specified duration
- disables the port after responding to the IP SLA packet or specified timeout
- may use MD5 authentication for control message
- takes two timestamps at the target devices to eliminate processing time
- can track one-way delay, jitter, and directional packet loss

- requires NTP synchronization except for one-way jitter measurement



$$RTT \text{ (Round-trip time)} = T4 \text{ (Time stamp 4)} - T1 \text{ (Time stamp 1)} - \Delta$$

46.1.3. IP SLAs Operation Scheduling

Task: Schedule IP SLA Operation

```
(config)# ip sla schedule <operation-number>
          [ life {forever} | <seconds> ]
          [start-time {now | pending | after <hh:mm:ss> |
<hh:mm:ss>} ] [month day | day month ] [ageout <seconds>]
          [recurring]
```

Task: Verify IP SLA Configuration

```
# sh ip sla configuration
```

- You can create a multioperation group
 - The frequency of all operations scheduled must be the same
 - The list of operation ID numbers must be limited to a max of 125 chars including commas

Task: Schedule a Group Of IP SLA Operations

```
(config)# ip sla group schedule <group-operation-number> <operation-id-numbers>
          { schedule-period <schedule-period-range | schedule-together
          }
          [ frequency <group-operation-frequency> ]
          [ life {forever} | <seconds> ]
          [start-time {now | pending| after <hh:mm:ss> |
<hh:mm:ss>} ] [month day | day month ] [ageout <seconds>]
          [recurring]
```

Task: Verify IP SLA Multioperation Configuration

```
# sh ip sla group schedule
```

46.1.4. IP SLAs Operation Threshold Monitoring

- can send SNMP traps that are triggered by connection loss, timeout, round-trip time threshold,

average jitter threshold, one-way packet loss, one-way jitter, one-way mean opinion score, one-way latency

- can trigger another IP SLA operation

46.1.5. MPLS VPN Awareness

- IP SLA operations can be configured for a specific VPN

46.1.6. History Statistics

Aggregated statistics

By default, two hours of aggregated statistics for each operation. Value from each operation cycle is aggregated with the previously available data within a given hour.

Operation snapshot history

Snapshot of data for each operation instance that matches a configurable filter, such as all, over threshold, or failures. The entire dataset is available and no aggregation takes place.

Distribution statistics

Frequency distribution over configurable intervals. Each time IP SLAs starts an operation, a new history bucket is created until the number of history buckets matches the specified size or the lifetime of the operation expires. By default, the history for an IP SLAs operation is not collected. If history is collected, each bucket contains one or more history entries from the operation. History buckets do not wrap.

46.1.7. Troubleshooting Tips

- If IP SLAs operation is not running and not generating stats, add the **verify-data** command in ip sla configuration mode
- Use **debug ip sla trace** and **debug ip sla error** commands

46.2. Enhanced Object Tracking

- separates definition of tracked object vs action to be taken when tracked object change
- tracked object can be: interface, ip route or ip sla operation or list of objects

46.2.1. Interface Tracking

- can be line protocol or ip route
- may consider the carrier delay timer
- may have a delay between the changes and the notification
- may change the polling frequency

Interface Line Protocol Tracking

Task: Track the Line Protocol State Of an Interface

```
(config)# track <object-number> interface <type number> line protocol
```

Interface IP Routing Tracking

- The IP routing State is up if
 - ip routing is enable and active on the interface
 - known ip address (static, dhcp , ppp/ipcp, unnumbered)
 - interface line protocol up

Task: Track the IP Routing State Of an Interface

```
(config)# track <object-number> interface <type number> ip routing
```

EOT Support for Carrier Delay

- If a link fails, by default there is a two-second timer that must expire before an interface and the associated routes are declared as being down. If a link goes down and comes back up before the carrier delay timer expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. You can extend the timer up to 60 seconds.
- When EOT is configured on an interface, the tracking may detect the interface is down before a configured carrier-delay timer has expired. This is because EOT looks at the interface state and does not consider the carrier delay timer.

Task: (Optional) Enables EOT to Consider the Carrier-Delay Timer When Tracking the Status Of an Interface.

```
(config-track)# carrier-delay
```

Task: (Optional) Specifies a Period Of Time (In Seconds) to Delay Communicating State Changes Of a Tracked Object.

```
(config-track)# delay {up <seconds> [down <seconds>] | down <seconds> [up <seconds>]}
```

46.2.2. IP Route Tracking

- Up if the route exists in the RIB and the route is accessible
- polls the ip route state every 15 seconds

Task: Track an IP Route

```
(config)# track <object-number> ip route <a.b.c.d/prefix> reachability
```

Task:(Optional) Specifies the Interval In Which the Tracking Process Polls the Tracked Object.

```
(config)# track timer ip route { <seconds> | msec <milliseconds> }
```

TODO: scaled metrics

46.2.3. IP SLA Operation Tracking

- tracks the state or the reachability of IP SLA operations -

Task: Track the Reachability Of an IP SLA Host

```
(config)# track <object-number> ip sla <operation-number> reachability
```

Example

```
# show track 3
Device# show track 3

Track 3
  IP SLA 1 reachability
  Reachability is Up
  1 change, last change 00:00:47
  Latest operation return code: over threshold
  Latest RTT (millisecs) 4
  Tracked by:
    HSRP Ethernet0/1 3
```

46.2.4. Tracked List

- can be constructed with
 - boolean expression
 - threshold and weight
 - threshold and percentage

Task: Configure Tracked List Object with a Boolean Expression

```
(config)# track <list-object-number> list boolean {and | or}
(config-track)# object <object-number> [not]
```

Task: Configure Tracked List Object with Threshold and Weight

```
(config)# track <list-object-number> list threshold weight  
(config-track)# object <object-number> [weight <number>]  
(config-track)# object <object-number> [weight <number>]  
(config-track)# threshold weight {up <number> | down <number>| up <number> down  
<number> }
```

Task: Configure Tracked List Object with Threshold and Percentage

```
(config)# track <list-object-number> list threshold percentage  
(config-track)# object <object-number>  
(config-track)# object <object-number>  
(config-track)# threshold percentage {up <number> | down <number>| up <number> down  
<number> }
```

Task: Configure Tracked List Default

```
(config-track)# default { delay | object <object-number> | threshold percentage }
```

46.3. NetFlow

- Cisco IOS application
- provides statistics on packets flowing through the routing

46.3.1. NetFlow Flows

- unidirectional stream of packets between a given source and destination
- combination of 7 fields: source/destination IP address/Port number, layer 3 protocol type, ToS, input logical interface
- can include accounting fields (such as AS number in v5)
- stored/captured in **netflow cache**
- exported to the flow collector (NetFlow Collection Engine)
- prerequisites
 - ip routing
 - cef, dcef or fast switching
 - enough CPU and memory
- data capture
 - from ingress: ip-to-ip, ip-to-mpls, FR and ATM terminated packets
 - for egress: ip-to-ip(with Netflow Accounting), mpls-to-ip (NetFlow MPLS Egress)

Task: Enable NetFlow Capture

```
(config-if)# ip flow { ingress | egress }
```

Task: Verify That NetFlow Is Operational

```
# sh ip cache [verbose] flow
```

46.3.2. NetFlow Version

v5

--

v9

- flexible and extensible
- template based
- uses 2 flow record type: template flowset, and data flowset
- v5 and v9 use the same packet structure

IP Header	UDP Header	NetFlow Header	Flow Record	Flow Record	..	Flow Record
-----------	------------	----------------	-------------	-------------	----	-------------

Version 5

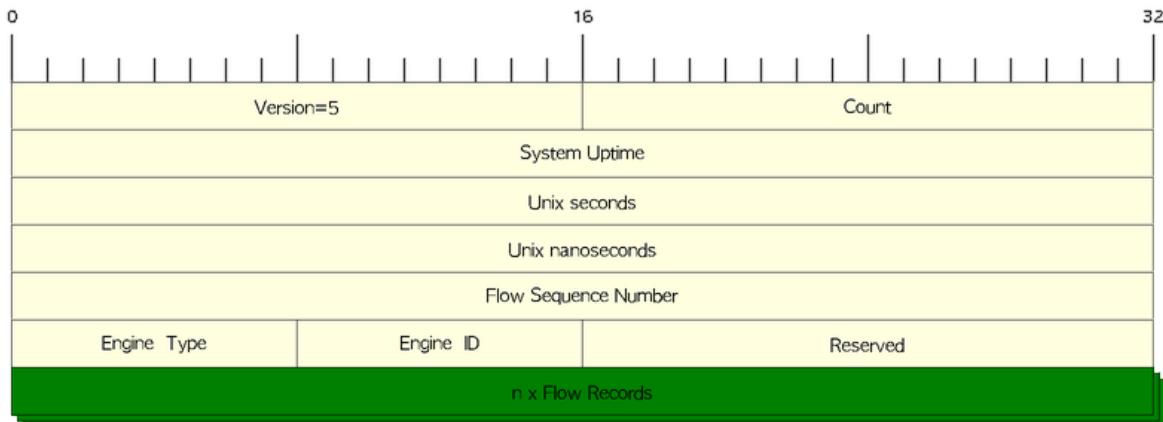


Figure 55. NetFlow Header Format

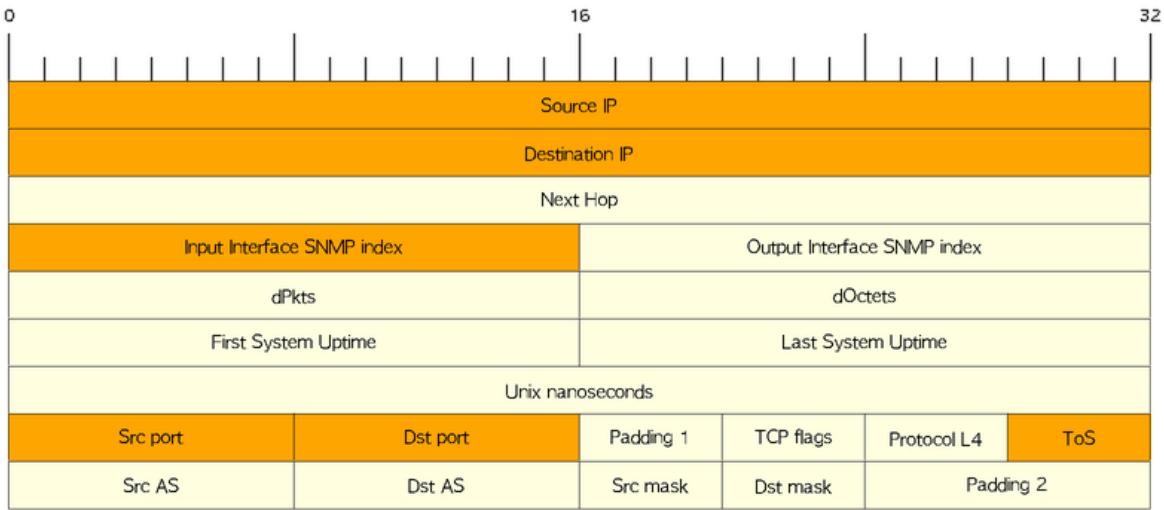


Figure 56. Flow Record Format

Version 9

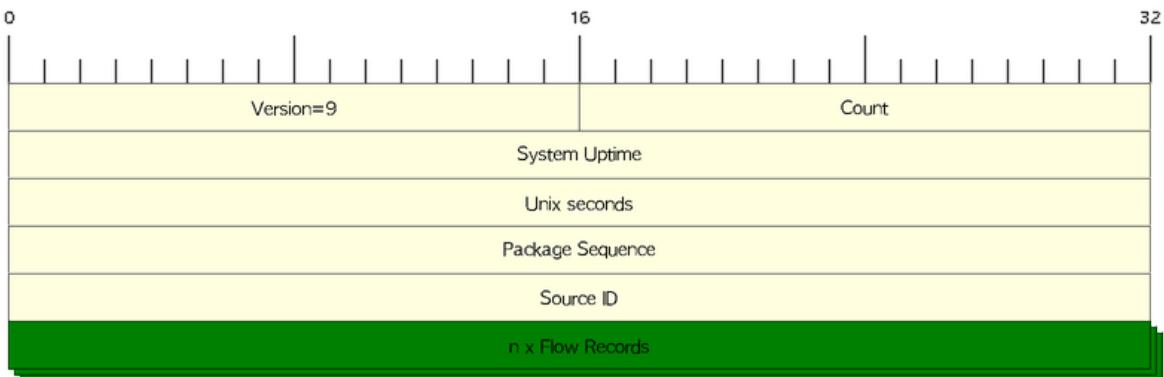


Figure 57. NetFlow Header Format

46.3.3. NetFlow Cache

- contains flow record for all active flows.
- up to 64K flow entries, each cache entry requires 64 bytes
- removes flows if
 - flow transport is completed (TCP FIN or RST)
 - flow cache full
 - flow becomes inactive after 15 seconds
 - flow active for more than 30 minutes

Task: Configure the Size Of the NetFlow Cache

```
(config)# ip flow-cache entries <size=64000>
```

Task: Configure the Flow Cache Timeout for Inactive Flow

```
(config)# ip flow-cache timeout inactive <seconds=15>
```

Task: Configure the Flow Cache Timeout for Active Flows

```
(config)# ip flow-cache timeout active <minutes=30>
```

46.3.4. NetFlow Data Export

- Send NetFlow cache entries to workstation running NetFlow Collection Engine
- supports only two export destinations

Task: Export NetFlow Information to a Workstation

```
(config)# ip flow-export destination {<ip-address | hostname>} <udp-port>
```

Task: (Optional) Use Version 5 Export Format

```
(config)# ip flow-export version 5
```

Task: (Optional) Use Version 9 Export Format

```
(config)# ip flow-export version 9
```

Task: Verify That NetFlow Data Export Is Operational

```
# sh ip flow export
```

46.4. Embedded Event Manager

[Configuration Guides](#) › [Embedded Management](#) › [Embedded Event Manager Overview](#)

- in-box, event tracking management
- EEM Components:
 - EEM server – an internal process that handles the interaction between the publishers and subscribers.
 - EEM publisher (detector) – software that screens events, publishes if there is a policy match. Some of the different detectors are CLI, NetFlow, SNMP, syslog, timers and counters.
 - EEM subscriber (policy) – defines an event and actions to be taken. There are two policy types, applets with IOS CLI and scripts written using TCL.

The creation of an EEM policy involves:

- Selecting the event for which the policy is run.
- Defining the event detector options associated with logging and responding to the event.
- Defining the environment variables, if required.
- Choosing the actions to be performed when the event occurs.

46.5. Embedded Packet Capture

[Configuration Guides](#) › [Embedded Management](#) › [Embedded Packet Capture](#)

- onboard packet capture facility
- captures packets flowing to, through, and from the device
- to analyze them locally or save and export them for offline analysis by using a tool such as Wireshark.
- restrictions:
 - only captures multicast packets on ingress and does not capture the replicated packets on egress.
 - Currently, the capture file can only be exported off the device; for example, TFTP or FTP servers and local disk.
- benefits:
 - Ability to capture IPv4 and IPv6 packets in the CEF path.
 - A flexible method for specifying the capture buffer parameters.
 - Filter captured packets.
 - Methods to decode data packets captured with varying degree of detail.
 - Facility to export the packet capture in PCAP format suitable for analysis using an external tool.
 - Extensible infrastructure for enabling packet capture points.
- The network administrator may define the capture buffer size and type (circular, or linear) and the maximum number of bytes of each packet to capture. The packet capture rate can be throttled using further administrative controls. For example, options allow for filtering the packets to be captured using an Access Control List and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

46.5.1. Capture Buffer

- area in memory for holding the packet data. You can specify unique names, size and type of the buffer, and configure the buffer to handle incoming data as required.
- can stored the following types of data are stored in a capture buffer: Packet data, Metadata
 - The packet data starts from datagramstart and copies a minimum of the per-packet-capture size or datagramszie to the capture buffer.
 - The metadata contains descriptive information about a set of packet data. It contains:

- A timestamp of when it is added to a buffer.
 - The direction in which the packet data is transmitted—egress or ingress.
 - The switch path captured.
 - Encapsulation type corresponding to input or output interface to allow the decoding of L2 decoders.
- The following actions can be performed on capture buffers:
 - Define a capture buffer and associate it with a capture point.
 - Clear capture buffers.
 - Export capture buffers for offline analysis. Export writes off the file using one of the supported file transfer options: FTP, HTTP, HTTPS, PRAM, RCP, SCP, and TFTP.
 - Display content of the capture buffers.

Task: Defines a capture buffer

```
(config)# monitor capture buffer <buffer-name>
[clear
 | export <location>
 | filter access-list {<acl>}
 | limit {allow-nth-pak <nth-packet> | duration <seconds> | packet-count <total-packets> | packets-per-sec <packets>}
 | [max-size <element-size>] [size <buffer-size>] [circular| linear]
 ]
```

Task: Export EPC data for analysis

```
(config)# monitor capture buffer <name> export <location>
```

Task: Display EPC data

```
# show monitor capture {buffer { <capture-buffer-name> [parameters] | all parameters |
merged <capture-buffer-name1> <capture-buffer-name2>}
[dump] [filter filter-parameters]
}
| point {all | capture-point-name}
}
```

Example

```
# sh monitor capture buffer PKTRACE dump

11:13:00.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1
65B6F500: 080020A2 44D90009 E94F8406 08004500 .. "DY..i0....E.
65B6F510: 00400F00 0000FE01 92AF5801 13025801 .@....~.../X...X.
65B6F520: 58090800 4D1A1169 00000000 0005326C X...M..i.....21
65B6F530: 01CCABCD ABCDABCD ABCDABCD ABCDABCD .L+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCD00 +M+M+M+M+M+M+M.
11:13:20.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1
```

46.5.2. Capture Point

- traffic transit point where a packet is captured and associated with a buffer.
- must be associate to one and only one capture buffer
- The following capture points are available:
 - IPv4 CEF/interrupt switching path with interface input and output
 - IPv6 CEF/interrupt switching path with interface input and output
- possible actions on the capture point:
 - Associate or disassociate capture points with capture buffers.
 - Destroy capture points.
 - Activate packet capture points on a given interface. Multiple packet capture points can be made active on a given interface. For example, Border Gateway Protocol (BGP) packets can be captured into one capture buffer and Open Shortest Path First (OSPF) packets can be captured into another capture buffer.
 - Access Control Lists (ACLs) can be applied to capture points.

Task: Define a capture point

```
(config)# monitor capture point {ip| ipv6}
          {cef <capture-point-name> interface-name <interface-type> { both | in | out} | process-switched <capture-point-name> {both| from-us| in | out}}
```

Task: Associate the capture point with the capture buffer specified.

```
(config)# monitor capture point associate <capture-point-name> <capture-buffer-name>
```

Task: start packet data capture

```
(config)# monitor capture point start {<capture-point-name> | all}
```

Task: Stop packet data capture

```
(config)# monitor capture point stop {<capture-point-name> | all}
```

Task: Enable packet capture infra debugs

```
# debug packet-capture
```

46.5.3. Using Wireshark trace analyzer

Beginning with Cisco IOS Release XE 3.3.0SG, the Catalyst 4500 series switch supports Wireshark, a packet analyzer program, also known as Ethereal, which supports multiple protocols and presents information in a text-based user interface. The key concepts around IOS XE based wireshark are:

- Capture points (a capture point is the central policy definition of the Wireshark feature)
- Attachment points (it refers to Interfaces and traffic directions)
- Filters (filters are attributes of a capture point that identify and limit the subset of traffic traveling through the attachment point of a capture point, which is copied and passed to Wireshark)
- Actions
- Storing captured packets to memory buffers

Further Reading <http://goo.gl/n67lEF>

46.6. Performance Monitor

TODO

Cisco performance monitor enables you to monitor the flow of packets in your network and become aware of any issues that might impact the flow before it starts to significantly impact the performance of the application in question. Performance monitoring is especially important for video traffic because high quality interactive video traffic is highly sensitive to network conditions (such as packet drops). Even minor issues that may not affect other applications can have dramatic effects on video quality.

Further Reading <http://goo.gl/UReuBR>

Part VI : Evolving Technologies

Chapter 47. Cloud

IT resources and services that are abstracted from the underlying infrastructure and provided **on-demand** and **at scale** in a **multitenant** environment.

Characteristics

- on-demand self-services
- broad network access
- resource pooling
- rapid elasticity
- measured services

47.1. Compare and Contrast Cloud Deployment Models

- Public
- Private
- virtual Private
- inter-cloud

47.1.1. Infrastructure, Platform, and Software Services [XaaS]

- SaaS : Application services
- PaaS: run-time environment and software development frameworks and components presented as API
- IaaS: compute, network and storage
- IT foundation: basic building blocks, core technologies

47.1.2. Performance and Reliability

add comparisons table here

47.1.3. Security and Privacy

add comparison table here

47.1.4. Scalability and Interoperability

add comparison table here

47.2. Describe Cloud Implementations and Operations

How to connect to the cloud

- Private WAN (like MPLS L3VPN)
- Internet Exchange Point (IXP)
- Internet VPN

47.2.1. Automation and Orchestration

- Automation: single task
- Orchestration: process/workflow ordered set of tasks glued together with conditions

47.2.2. Workload Mobility

- share workloads across a collection of resources
- different from VM mobility

47.2.3. Troubleshooting and Management

- Scripting languages: Python, Ruby,
- NETCONF
 - transport protocol by which configurations are installed and changed
 - [RFC 6241](#)
 - 4 layers:
 - Operations layer: RPC methods with XML-encoded parameters
 - Secure transport layer: communication path between the client and the server.
 - Content layer: outside the scope,
 - Messages layer: transport-independent framing mechanisms for encoding RPCs and notifications
- YANG
 - modeling language used to represent device configuration and state (~ XML)
 - [RFC 6020](#)

47.2.4. OpenStack Components

Compute (Nova)

Fabric controller (the main part of an IaaS system). Manages pools of computer resources. A compute resource could be a VM, container, or bare metal server. Side note: Containers are similar to VMs except they share a kernel. They are otherwise independent, like VMs, and are considered a lighter-weight yet secure alternative to VMs.

Networking (Neutron)

Manages networks and IP addresses. Ensures the network is not a bottleneck or otherwise limiting factor in a production environment. This is technology-agnostic network abstraction which allows the user to create custom virtual networks, topologies, subnets, gateway addresses,

DNS, etc.

Block Storage (Cinder)

Manages creation, attaching, and detaching of block storage devices to servers. This is not an implementation of storage itself, but provides an API to access that storage. Many storage appliance vendors often have a Cinder plug-in for OpenStack integration; this ultimately abstracts the vendor-specific user interfaces from the management process. Storage volumes can be detached and moved between instances (an interesting form of file transfer, file example) to share information and migrate data between projects.

Identity (Keystone)

Directory service contains users mapped to services they can access. Somewhat similar to group policies applied in corporate deployments. Tenants are stored here which allows them to access resources/services within OpenStack; commonly this is access to the OpenStack Dashboard (Horizon) to manage an OpenStack environment.

Image (Glance)

Provides discovery, registration, and delivery services. These images are like ordinary images to template new virtual services.

Object Storage (Swift)

Storage system with built-in data replication and integrity. Objects and files are written to disk using this interface which manages the I/O details. Scalable and resilient storage for all objects like files, photos, etc. This means the customer doesn't have to deploy a block-storage solution themselves, then manage the storage protocols (iSCSI, NFS, etc).

Dashboard (Horizon)

The GUI for administrators and users to access, provision, and automate resources. The dashboard is based on Python Django framework and is layered on top of service APIs. Logging in relies on Keystone for identity management which secures access to the GUI. The dashboard supports different tenants (business units, groups/teams, customers, etc) with separate permissions and credentials; this is effectively role-based access control. The GUI provides the most basic/common functionality for users without needing CLI access, which is supported for advanced functions. "Security group" abstractions to enforce access control (often need to configure this before being able to access the new instances).

Orchestration (Heat)

Service to orchestrate multiple cloud applications via templates using a variety of APIs.

Workflow (Mistral)

Manages user-created workflows which can be triggered manually or by some event.

Telemetry (Ceilometer)

Provides a Single Point of Contact for billing systems used within the cloud environment.

Database (Trove)

This is a Database-as-a-service provisioning engine.

Elastic Map Reduce (Sahara)

Automated way to provision Hadoop clusters, like a wizard.

Bare Metal (Ironic)

Provisions bare metal machines rather than virtual machines.

Messaging (Zaqar)

Cloud messaging service for Web Developments (full RESTful API) used to communicate between SaaS and mobile applications.

Shared File System (Manila)

Provides an API to manage shares in a vendor agnostic fashion (create, delete, grant/deny access, etc).

DNS (Designate)

Multi-tenant REST API for managing DNS (DNS-as-a-service).

Search (Searchlight)

Provides search capabilities across various cloud services and is being integrated into the Dashboard.

Key Manager (Barbican)

Provides secure storage, provisioning, and management of secrets (passwords).

47.2.5. Resources and References

- http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/CiscoCloudComputing_WP.pdf
- [Open Stack Components](#)
- [Cloud Overview](#)
- <http://www.unleashingit.com/>
- <http://www.cisco.com/go/cloud>
- <https://www.openstack.org/software/>
- [Designing Networks and Services for the Cloud](#)

Chapter 48. SDN

SDN: Software-Defined Network

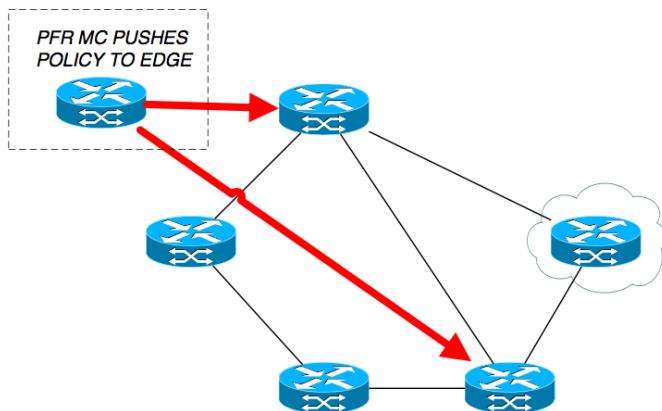
- approach to computer networking that allows network administrators to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces and abstraction of lower-level functionality.

48.1. Models

Distributed

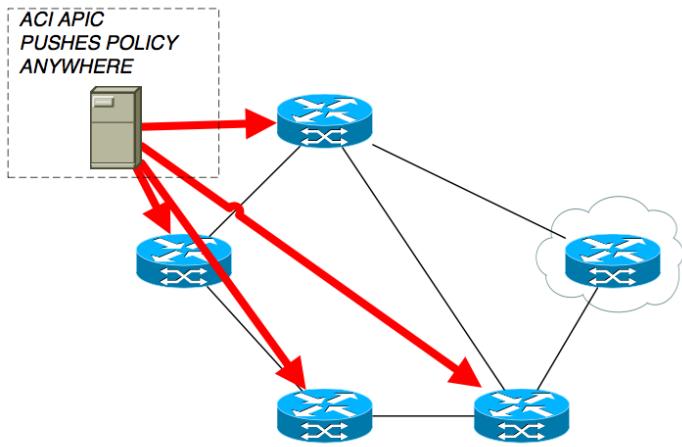
- control-plane distributed across all devices.
- the status quo, not “SDN” model at all.
- each network devices have their own control-plane components which rely on distributed routing protocols (such as OSPF, BGP, etc).

Augmented



- fully distributed control-plane by adding a centralized controller that can apply policy to parts of the network at will. Such a controller could inject shorter-match IP prefixes, policy-based routing (PBR), security features (ACL), or other policy objects.
- good compromise between distributing intelligence between nodes to prevent single points of failure (which a controller introduces) by using a known-good distributed control-plane underneath. The policy injection only happens when it “needs to”, such as offloading an overloaded link in a DC fabric or from a long-haul fiber link between two points of presence (POPs) in an SP core.
- Examples:
 - Cisco’s Performance Routing (PfR)
 - offline path computation element (PCE) servers for automated MPLS TE tunnel creation.
- lower impact on the existing network because the wholesale failure of the controller simply returns the network to the distributed model, which is known to work “well enough” in many cases.

Hybrid



- like the augmented model except that controller-originated policy can be imposed anywhere in the network.
- additional granularity to network administrators;
- the main benefit over the augmented model is that the hybrid model is always topology-independent.
 - The controller can overwrite the forward table of any device which means that topological restrictions are removed.
- Example: Cisco's Application Centric Infrastructure (ACI) separates reachability from policy, which is critical from both survivability and scalability perspectives.
- The failure of the centralized control in these models has an identical effect to that of a controller in the augment model; the network falls back to a distributed control-plane model. The impact of a failed controller is a little more significant since more devices are affected by the controller's policy.

Centralized

- single controller, which hosts the entire control-plane.
 - commands all of the devices in the forwarding-plane.
 - writes their forwarding tables with the proper information (which doesn't necessarily have to be an IP-based table, it could be anything) as specified by the administrators.
- offers very granular control, in many cases, of individual flows in the network.
- can use inexpensive hardware forwarders commoditized into white boxes (or branded white boxes, sometimes called brite boxes)
- offers more flexibility to the business because network "device" can be almost anything: router, switch, firewall, load-balancer, etc. Emulating software functions on generic hardware
- single point of failure
 - Some SDN scaling architectures suggest simply adding additional controllers for fault tolerance or to create a hierarchy of controllers for larger networks. While this is a valid technique, it somewhat invalidates the “centralized” model because with multiple controllers, the distributed control-plane is reborn. The controllers still must synchronize

their routing information using some network-based protocol and the possibility of inconsistencies between the controllers is real. When using this multi-controller architecture, the network designer must understand that there is, in fact, a distributed control-plane in the network; it has just been moved around. The failure of all controllers means the entire failure domain supported by those controllers will be inoperable. The failure of the communication paths between controllers could likewise cause inconsistent/intermittent problems with forward, just like a fully distributed control-plane.

- Example: OpenFlow

TODO

48.2. Describe Functional Elements Of Network Programmability and How They Interact

48.2.1. Controllers

- responsible for programming forwarding tables of data-plane devices
- can be physical routers, like Cisco's PfR operating as a master controller (MC), or they could be software-only appliances, as seen with OpenFlow networks or Cisco's Application Policy Infrastructure Controller (APIC) used with ACI.

48.2.2. APIs

- standard way of interfacing with a software application or operating system.
- typically use REST (Representational State Transfer)
 - represents an “architectural style” of transferring information between clients and servers.
 - used with stateless HTTP by combining traditional HTTP methods (GET, POST, PUT, DELETE, etc) and Universal Resource Identifiers (URI).
 - The end result is that API requests look like URIs and are used to fetch/write specific pieces of data to a target machine.
 - promotes automation, especially for web-based applications or services.

48.2.3. Scripting

- Ruby, Python

TODO ACI Application Centric Infrastructure

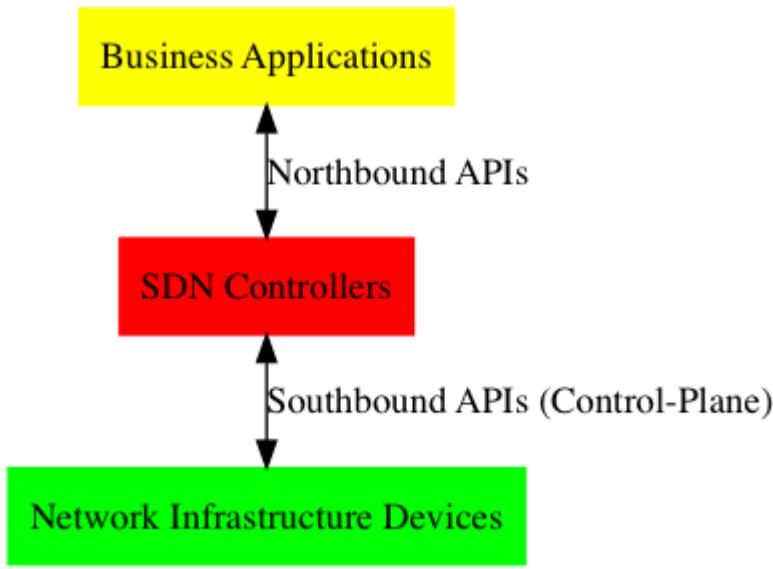
48.2.4. Agents

- typically on-box software components that allow an infrastructure device to report traffic conditions back to the controller.
 - Given this information, the controller can sense congestion, route around failures, and

perform all manner of fancy traffic-engineering as required by the business applications.

- perform the same general function as SNMP yet offer increased flexibility and granularity as they are programmable.
- can be used for non-management purposes, at least from a general view.
- Interface to the Routing System (I2RS) is an SDN technique where a specific control-plane agent is required on every data-plane forwarder.
 - This agent is effectively the control-plane client that communicates upstream towards the controller. This is the channel by which the controller consults its RIB and populates the FIB of the forwarding devices. The same is true for OpenFlow (OF) which is a fully centralized SDN model. The agent can be considered an interface to a data-plane forwarder for a control-plane SDN controller.
- A simple categorization method is to quantify management strategies as “agent based” or “agent-less based”.
 - Agent is pull-based, which means the agent connects with master. Changes made on master are pulled down when agent is “ready”. This can be significant since if a network device is currently tolerating a microburst, the management agent can wait until the contention abates before passing telemetry data to the master.
 - Agent-less is push-based like SNMP traps, where the triggering of an event on a network device creates a message for the controller in unsolicited fashion. The other direction also true; a master can use SSH to access a device for programming whenever the master is “ready”.
- Examples
 - Puppet: by Puppet labs, agent-based (requiring software installed on the client) and pushes complex data structures to managed nodes from the master server. Puppet manifests are used as data structures to track node state and display this state to the network operators. Uses Ruby.
 - Chef (by Chef Software): very similar to Puppet in that it requires agents and manages devices using complex data structures. The concepts of cookbooks and recipes are specific to Chef (hence the name) which contribute to a hierarchical data structure management system. A Chef cookbook is loosely equivalent to a Puppet manifest.
 - Ansible (by Redhat): lighter-weight than Puppet or Chef given that management is agent-less. No custom software needs to be installed on any device provided that it supports SSH. This can be a drawback since individual device CLIs must be exposed to network operators (or, at best, the Ansible automation engine) instead of using a more abstract API design. Uses Python and Powershell.
 - SaltStack: CLI-based, master-client or non-centralized environments

48.2.5. Northbound Vs. Southbound Protocols



Northbound interfaces

- APIs interfaces to existing business applications.
- used so that applications can make requests of the network, which could include specific performance requirements (bandwidth, latency, etc). Because the controller “knows” this information by communicating with the infrastructure devices via management agents, it can determine the best paths through the network to satisfy these constraints.
- loosely analogous to the original intent of the Integrated Services QoS model using Resource Reservation Protocol (RSVP) where applications would reserve bandwidth on a per-flow basis.
- It is also similar to MPLS TE constrained SPF (CSPF) where a single device can source-route traffic through the network given a set of requirements.
- The logic is being extended to applications with a controller “shim” in between, ultimately providing a full network view for optimal routing.
- Typically REST API

Southbound interfaces

- include the control-plane protocol between the centralized controller and the network forwarding hardware. These are the less intelligent network devices used for forwarding only (assuming a centralized model).
- Example: OpenFlow

48.3. Describe Aspects Of Virtualization and Automation In Network Environments

- Creation of virtual topologies using a variety of technologies to achieve a given business goal. Sometimes these virtual topologies are overlays, sometimes they are forms of multiplexing, and sometimes they are a combination of the two:
 - a. **Ethernet VLANs** using 802.1q encapsulation. Often used to create virtual networks at layer 2 for security segmentation, traffic hair pinning through a service chain, etc. This is a form

of data multiplexing over Ethernet links. It isn't a tunnel/overlay since the layer 2 reachability information (MAC address) remains exposed and used for forwarding decisions.

- b. **VRF tables or other layer-3 virtualization techniques.** Similar uses as VLANs except virtualizes an entire routing instance, and is often used to solve a similar set of problems. Can be combined with VLANs to provide a complete virtual network between layers 2 and 3. Can be coupled with GRE for longer-range virtualization solutions over a core network that may or may not have any kind of virtualization. This is a multiplexing technique as well but is control-plane only since there is no change to the packets on the wire, nor is there any inherent encapsulation (not an overlay).
- c. **Frame Relay DLCI encapsulation.** Like a VLAN, creates segmentation at layer 2 which might be useful for last-mile access circuits between PE and CE for service multiplexing. The same is true for Ethernet VLANs when using EV services such as EV-LINE, EV-LAN, and EV-TREE. This is a data-plane multiplexing technique specific to frame relay.
- d. **MPLS VPNs.** Different VPN customers, whether at layer 2 or layer 3, are kept completely isolated by being placed in a different virtual overlay across a common core that has no/little native virtualization. This is an example of an overlay type of virtual network.
- e. **VXLAN.** Just like MPLS VPNs; creates virtual overlays atop a potentially non-virtualized core. Doesn't provide a native control-plane, but that doesn't matter; it's still a virtualization technique. Could be paired with BGP EVPN if MAC routing is desired. This is another example of an overlay type of virtual network.
- f. **OTV.** Just like MPLS VPNs; creates virtual overlays atop a potentially non-virtualized core, except provides a control-plane for MAC routing. IP multicast traffic is also routed intelligently using GRE encapsulation with multicast destination addresses. This is another example of an overlay type of virtual network.

48.3.1. DevOps Methodologies, Tools and Workflows

- Culture: People over Process over Tools
- CI/CD: Continuous Integration / Continuous Deployment
 - a. Everyone can see the changes: Dev, Ops, Quality Assurance (QA), management, etc
 - b. Verification is an exact clone of the production environment, not simply a smoke-test on a developer's test bed
 - c. The build and deployment/upgrade process is automated
 - d. Provide SW in short timeframes and ensure releases are always available in increments
 - e. Reduce friction, increase velocity
 - f. Reduce silos, increase collaboration

48.3.2. Network/Application Function Virtualization [NFV, AFV]

NFV and AFV refer to taking specific network functions, virtualizing them, and assembling them in a sequence to meet a specific business need. NFV and AFV by themselves, in isolation, are generally synonymous with creating virtual instances of things which were once physical. Many vendors

offer virtual routers (Cisco CSR1000v, Cisco IOS-XR9000v, etc), security appliances (Cisco ASA, Cisco NGIPSv, etc), telephony and collaboration components (Cisco UCM, CUC, IM&P, UCCX, etc) and many other things that were once physical appliances. Separating these things into virtual functions allows a wide variety of organizations, from cloud providers to small enterprises, to select only the components they require.

48.3.3. Service Function Chaining

- service chaining is taking NFV/AFV components and sequencing them to create some customized “chain of events” to solve a business problem. NFV/AFV by itself isn’t terribly useful if specific services cannot be easily linked in a meaningful way. Service chaining, especially in cloud environments, can be achieved in a variety of technical ways. For example, one organization may require routing and firewall, while another may require routing and intrusion prevention. The per- customer granularity is a powerful offering of service chaining in general. The main takeaway is that all of these solutions are network virtualization solutions of sorts.
 - a. MPLS and Segment Routing. Some headend LSR needs to impose different MPLS labels for each service in the chain that must be visited to provide a given service. MPLS is a natural choice here given the label stacking capabilities and theoretically-unlimited label stack depth.
 - b. Networking Services Header (NSH). Similar to the MPLS option except is purpose-built for service chaining. Being purpose-built, NSH can be extended or modified in the future to better support new service chaining requirements, where doing so with MPLS shim header formats is less likely. MPLS would need additional headers or other ways to carry “more” information.
 - c. Out of band centralized forwarding. Although it seems unmanageable, a centralized controller could simply instruct the data-plane devices to forward certain traffic through the proper services without any in-band encapsulation being added to the flow. This would result in an explosion of core state which could limit scalability, similar to policy-based routing at each hop.
 - d. Cisco vPath: This is a Cisco innovation that is included with the Cisco Nexus 1000v series switch for use as a distributed virtual switch (DVS) in virtualized server environments. Each service is known as a virtual service node (VSN) and the administrator can select the sequence in which each node should be transited in the forwarding path. Traffic transiting the Nexus 1000v switch is subject to redirection using some kind of overlay/encapsulation technology. Specifically, MAC-in-MAC encapsulation is used for layer-2 tunnels while MAC-in-UDP is used for layer-4 t

48.3.4. Performance, Availability, and Scaling Considerations

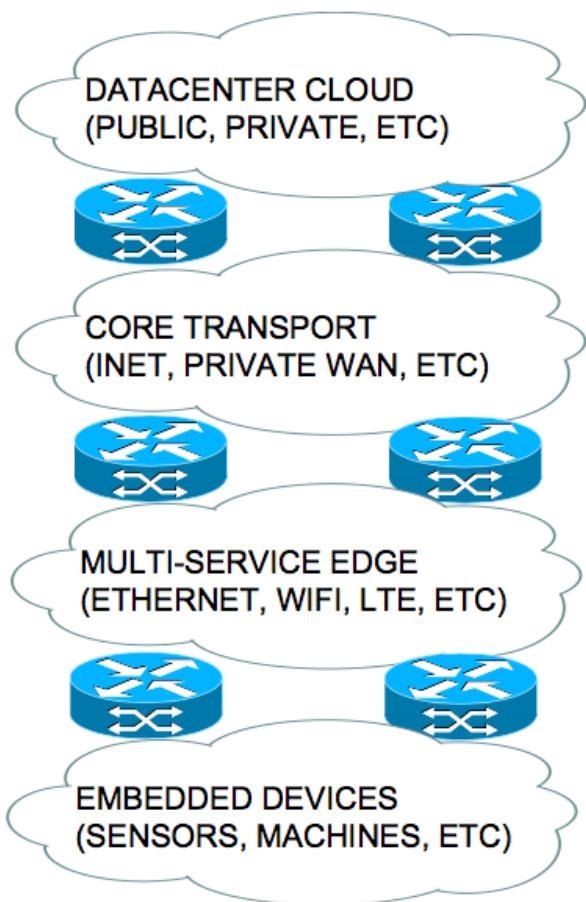
Model	Distributed	Augmented	Hybrid	Centralized
Availability	Dependent on the protocol convergence times and redundancy in the network	Dependent on the protocol convergence times and redundancy in the network. Doesn't matter how bad the SDN controller is ... its failure is tolerable	Dependent on the protocol convergence times and redundancy in the network. Doesn't matter how bad the SDN controller is ... its failure is tolerable	Heavily reliant on a single SDN controller, unless one adds controllers to split failure domains or to create resilience within a single failure domain (which introduces a distributed control-plane in both cases)
Granularity / control	Generally low for IGPs but better for BGP. All devices generally need a common view of the network to prevent loops independently. MPLS TE helps somewhat.	Better than distributed since policy injection can happen at the network edge, or a small set of nodes. Can be combined with MPLS TE for more granular selection.	Moderately granular since SDN policy decisions are extended to all nodes. Can influence decisions based on any arbitrary information with a datagram	Very highly granular; complete control over all routing decisions based on any arbitrary information with a datagram
Scalability (assume flow-based policy and state retention)	Very high in a properly designed network (failure domain isolation, topology summarization, reachability aggregation, etc)	High, but gets worse with more policy injection. Policies are generally limited to key nodes (such as border routers)	Moderate, but gets worse with more policy injection. Policy is proliferated across the network to all nodes (though the exact quantity may vary per node)	Depends; all devices retain state for all transiting flows. Hardware-dependent on TCAM and whether SDN can use other tables such as L4 information, IPv6 flow labels, etc

ONOS - Open Network Operating Systems by the Linux Foundation

Chapter 49. Internet Of Things

- sometimes called Internet of Everything (IoE),
- concept that many non-person entities (NPEs) or formerly non-networked devices in the world would suddenly be networked.
 - typically includes things like window blinds, light bulbs, water treatment plant sensors, home heating/cooling units, street lights, and anything else that could be remotely controlled or monitored.
- substantial business drivers for IoT:
 - electrical devices (like lights and heaters) could consume less energy by being smartly adjusted based on changing conditions,
 - window blinds can open and close based on the luminosity of a room,
 - chemical levels can be adjusted in a water treatment plant by networked sensors.
- Low-power and Lossy Networks (LLN) describes the vast majority of IoT networks with the following basic characteristics (incomplete list):
 - Bandwidth constraints
 - Highly unreliable
 - Limited resources (power, CPU, and memory)
 - Extremely high scale (hundreds of millions and possibly more)

49.1. Describe Architectural Framework and Deployment Considerations for IoT



49.1.1. Data center (DC) Cloud

- Although not a strict requirement, the understanding that a public cloud infrastructure exists to support IoT is a common one.
- A light bulb manufacturer could partner with a networking vendor to develop network-addressable light bulbs which are managed from a custom application running in the public cloud. This might be better than a private cloud solution since, if the application is distributed, regionalized instances could be deployed in geographically dispersed areas using an “anycast” design for scalability and performance improvements.

49.1.2. Core Networking and Services

- transports to connect the public cloud to the sensors
- can be private WAN, IXP or Internet VPN
- use a common set of technologies/services: seen within this layer include IP, MPLS, mobile packet core, QoS, multicast, security, network services, hosted cloud applications, big data, and centralized device management (such as a network operations facility).

49.1.3. Multi-service Edge (access network)

- Like most SP networks, the access technologies tend to vary greatly based on geography, cost, and other factors. Access networks can be optically-based to provide Ethernet handoffs to IoT devices; this would make sense for relatively large devices that would have Ethernet ports and would be generally immobile. Mobile devices, or those that are small or remote, might use

cellular technologies such as 2G, 3G, or 4G/LTE for wireless backhaul to the closest POP. A combination of the two could be used by extending Ethernet to a site and using 802.11 WIFI to connect the sensors to the WLAN. The edge network may require use of “gateways” as a short-term solution for bridging (potentially non-IP) IoT networks into traditional IP networks. The gateways come with an associated high CAPEX and OPEX since they are custom devices to solve a very specific use-case. Specifically, gateways are designed to perform some subset of the following functions, according to Cisco:

1. Map semantics between two heterogeneous domains: The word semantics in this context refers to the way in which two separate networks operate and how each network interprets things. If the embedded systems network is a transparent radio mesh using a non-standard set of protocols while the multi-service edge uses IP over cellular, the gateway is responsible for “presenting” common interfaces to both networks. This allows devices in both networks to communicate using a “language” that is common to each.
2. Perform translation in terms of routing, QoS security, management, etc: These items are some concrete examples of semantics. An appropriate analogy for IP networkers is stateless NAT64; an inside-local IPv4 host must send traffic to some outside-local IPv4 address which represents an outside-global IPv6 address. The source of that packet becomes an IPv6 inside-global address so that the IPv6 destination can properly reply.
3. Do more than just protocol changes: The gateways serve as interworking devices between architectures at an architectural level. The gateways might have a mechanism for presenting network status/health between layers, and more importantly, be able to fulfill their architectural role in ensuring end-to-end connectivity across disparate network types.

49.1.4. Embedded Systems (Smart Things Network)

This layer represents the host devices themselves. They can be wired or wireless, smart or less smart, or any other classification that is useful to categorize an IoT component. Often times, such devices support zero-touch provisioning (ZTP) which helps with the initial deployment of massive-scale IoT deployments. For static components, these components are literally embedded in the infrastructure and should be introduced during the construction of a building, factory, hospital, etc. These networks are rather stochastic (meaning that behavior can be unpredictable).

- General categories based on RF-level transmission methods :
 1. Long range: Some devices may be placed very far from their RF base stations/access points and could potentially be highly mobile. Smart automobiles are a good example of this; such devices are often equipped with cellular radios, such as 4G/LTE. Such an option is not optimal for supporting LLNs given the cost of radios and power required to run them. To operate a private cellular network, the RF bands must be licensed (in the USA, at least), which creates an expensive and difficult barrier for entry.
 2. Short range with “better” performance: Devices that are within a local area, such as a building, floor of a large building, or courtyard area, could potentially use unlicensed frequency bands while transmitting at low power. These devices could be CCTV sensors, user devices (phones, tablets, laptops, etc), and other general-purpose things whereby maximum battery life and cost savings are eclipsed by the need for superior performance. IEEE 802.11 WIFI is commonly used in such environments. IEEE 802.16 WIMAX specifications could also be used, but in the author’s experience, it is rare.

3. Short range with “worse” performance: Many IoT devices fall into this final category whereby the device itself has a very small set of tasks it must perform, such as sending a small burst of data when an event occurs (i.e., some nondescript sensor). Devices are expected to be installed one time, rarely maintained, procured/operated at low cost, and be value-engineered to do very few things. These devices are less commonly deployed in home environments since many homes have WIFI; they are more commonly seen spread across cities. Examples might include street lights, sprinklers, and parking/ticketing meters. IEEE has defined 802.15.4 to support low-rate wireless personal area networks (LR-PANS) which is used for many such IoT devices. Note that 802.15.4 is the foundation for upper-layer protocols such as ZigBee and WirelessHART. ZigBee, for example, is becoming popular in homes to network some IoT devices, such as thermostats, which may not support WIFI in their hardware.

IEEE 802.15.4 is worth a brief discussion by itself. Unlike WIFI, all nodes are “full-function” and can act as both hosts and routers; this is typical for mesh technologies. A device called a PAN coordinator is analogous to a WIFI access point (WAP) which connects the PAN to the wired infrastructure; this technically qualifies the PAN coordinator as a “gateway” discussed earlier.

As a general comment, one IoT strategy is to “mesh under” and “route over”. This loosely follows the 7- layer OSI model by attempting to constrain layers 1-2 to the IoT network, to include RF networking and link-layer communications, then using some kind of IP overlay of sorts for network reachability.

49.1.5. Performance, Reliability and Scalability

The performance of IoT devices is going to be a result of the desired security and the access type. Many IoT devices will be equipped with relatively inexpensive and weak hardware; this is sensible from a business perspective as the device only needs to perform a few basic functions. This could be seen a compromise of security since strong ciphers typically require more computational power for encryption/decryption functionality. In addition, some IoT devices may be expected to last for decades while it is highly unlikely that the same is true about cryptographic ciphers. In short, more expensive hardware is going to be more secure and resilient.

The access type is mostly significant when performance is discussed. Although 4G LTE is very popular and widespread in the United States and other countries, it is not available everywhere. Some parts of the world are still heavily reliant on 2G/3G cellular service which is less capable and slower. A widely distributed IoT network may have a combination of these access types with various levels of performance. Higher performing 802.11 WIFI speeds typically require more expensive radio hardware, more electricity, and a larger physical size. Physical access types (wired devices) will be generally immobilized which could be considered a detriment to physical performance, if mobility is required for an IoT device to do its job effectively.

49.1.6. Mobility

The mobility of an IoT device is going to be largely determined by its access method. Devices that are on 802.11 WIFI within a factory will likely have mobility through the entire factory, or possibly the entire complex, but will not be able to travel large geographic distances. For some specific manufacturing work carts (containing tools, diagnostic measurement machines, etc), this might be an appropriate method. Devices connected via 4G LTE will have greater mobility but will likely

represent something that isn't supposed to be constrained to the factory, such as a service truck or van. Heavy machinery bolted to the factory floor might be wired since it is immobile.

49.1.7. Security and Privacy

Providing security and privacy for IoT devices is challenging mostly due to the sheer expanse of the access network and supported clients (IoT devices). Similar concepts still apply as they would for normal hosts except for needing to work in a massively scalable and distributed network:

- a. Use IEEE 802.1x for wired and wireless authentication for all devices. This is normally tied into a Network Access Control (NAC) architecture which authorizes a set of permissions per device.
- b. Encrypt wired and wireless traffic using MACsec/IPsec as appropriate.
- c. Maintain physical accounting of all devices, especially small ones, to prevent theft and reverse engineering.
- d. Do not allow unauthorized access to sensors; ensure remote locations are secure also.
- e. Provide malware protection for sensors so that the compromise of a single sensor is detected quickly and suppressed.
- f. Rely on cloud-based threat analysis (again, assumes cloud is used) rather than a distributed model given the size of the IoT access network and device footprint. Sometimes this extension of the cloud is called the “fog” and encompasses other things that product and act on IOT data.

Another discussion point on the topic of security is determining how/where to “connect” an IoT network. This is going to be determined based on the business need, as always, but the general logic is similar to what traditional corporate WANs use:

- a. Fully private connections: Some IoT networks have no legitimate need to be accessible via the public Internet. Such examples would include Government sensor networks which may be deployed in a battlefield support capacity. More common examples might include Cisco's “Smart Grid” architecture which is used for electricity distribution and management within a city. Exposing such a critical resource to a highly insecure network offers little value since the public works department can likely control it from a dedicated NOC. System updates can be performed in-house and the existence of the IoT network can be (and often times, should be) largely unknown by the general population. In general, IoT networks that fall into this category are “producer-oriented” networks.
- b. Public Internet: Other IoT networks are designed to have their information shared or made public between users. One example might be a managed thermostat service; users can log into a web portal hosted by the service provider to check their home heating/cooling statistics, make changes, pay bills, request refunds, submit service tickets, and the like. Other networks might be specifically targeted to sharing information publicly, such as fitness watches that track how long an individual exercises. The information could be posted publicly and linked to one's social media page so others can see it. A more practical and useful example could include public safety information via a web portal hosted by the Government. In general, IoT networks that fall into this category are “consumed-oriented” networks.

49.1.8. Standards and Compliance

Controlling access and identifying areas of responsibility can be challenging with IoT. Cisco provides the following example: For example, Smart Traffic Lights where there are several interested parties such as Emergency Services (User), Municipality (owner), Manufacturer (Vendor). Who has provisioning access? Who accepts Liability?

There is more than meets the eye with respect to standards and compliance for street lights. Most municipalities (such as counties or townships within the United States) have ordinances that dictate how street lighting works. The light must be a certain color, must not “trespass” into adjacent streets, must not negatively affect homeowners on that street, etc. This complicates the question above because the lines become blurred between organizations rather quickly. In cases like this, the discussions must occur between all stakeholders, generally chaired by a Government/company representative (depending on the consumer/customer), to draw clear boundaries between responsibilities.

Radio frequency (RF) spectrum is a critical point as well. While WIFI can operate in the 2.4 GHz and 5.0 GHz bands without a license, there are no unlicensed 4G LTE bands at the time of this writing. Deploying 4G LTE capable devices on an existing carrier’s network within a developed country may not be a problem. Doing so in developing or undeveloped countries, especially if 4G LTE spectrum is tightly regulated but poorly accessible, can be a challenge.

Several new protocols have been introduced specifically for IoT, some of which are standardized:

RPL

IPv6 Routing Protocol for LLNs (RFC 6550): RPL is a distance-vector routing protocol specifically designed to support IoT. At a high-level, RPL is a combination of control-plane and forwarding logic of three other technologies: regular IP routing, multi-topology routing (MTR), and MPLS traffic-engineering (MPLS TE). RPL is similar to regular IP routing in that directed acyclic graphs (DAG) are created through the network. This is a fancy way of saying “loop-free shortest path” between two points. These DAGs can be “colored” into different topologies which represent different network characteristics, such as high bandwidth or low latency. This forwarding paradigm is similar to MTR in concept. Last, traffic can be assigned to a colored DAG based on administratively-defined constraints, including node state, node energy, hop count, throughput, latency, reliability, and color (administrative preference). This is similar to MPLS TE’s constrained shortest path first (CSPF) process which is used for defining administrator-defined paths through a network based on a set of constraints, which might have technical and/or business drivers behind them.

6LoWPAN

- IPv6 over Low Power WPANs (RFC 4919): This technology was specifically developed to be an adaptation layer for IPv6 for IEEE 802.15.4 wireless networks. Specifically, it “adapts” IPv6 to work over LLNs which encompasses many functions:
 - MTU correction: The minimum MTU for IPv6 across a link, as defined in RFC2460, is 1280 bytes. The maximum MTU for IEEE 802.15.4 networks is 127 bytes. Clearly, no value can mathematically satisfy both conditions concurrently. 6LoWPAN performs fragmentation and reassembly by breaking the large IPv6 packets into IEEE 802.15.4 frames for transmission across the wireless network.

- Header compression: Many compression techniques are stateful and CPU-hungry. This strategy would not be appropriate for low-cost LLNs, so 6LoWPAN utilizes an algorithmic (stateless) mechanism. RFC4944 defines some common assumptions:
 - The version is always IPv6.
 - Both source and destination addresses are link-local.
 - The low-order 64-bits of the link-local addresses can be derived from the layer-2 network addressing in an IEEE 802.15.4 wireless network.
 - The packet length can be derived from the layer-2 header.
 - Next header is always ICMP, TCP, or UDP.
 - Flow label and traffic class are always zero.

As an example, an IPv6 header (40 bytes) and a UDP header (8 bytes) are 48 bytes long when concatenated. This can be compressed down to 7 bytes by 6LoWPAN.

- Mesh routing: Somewhat similar to WIFI, mesh networking is possible, but requires up to 4 unique addresses. The original source/destination addresses can be retained in a new “mesh header” while the per-hop source/destination addresses are written to the MAC header.
- MAC level retransmissions: IP was designed to be fully stateless and any retransmission or flow control was the responsibility of upper-layer protocols, such as TCP. When using 6LoWPAN, retransmissions can occur at layer-2.

CoAP

- Constrained Application Protocol (RFC7252) designed for LLNs and M2M communications
- At a high-level, very similar to HTTP in terms of the capabilities it provides.
 - support the transfer of application data using common methods such as GET, POST, PUT, and DELETE.
 - runs over UDP port 5683 by default (5684 for secure CoAP) and was specifically designed to be lighter weight and faster than HTTP.
- Supports multicast: Because it is UDP-based, IP multicast is possible. This can be used both for application discovery (in lieu of DNS) or efficient data transfer.
- Built-in security: CoAP supports using datagram TLS (DTLS) with both pre-shared key and digital certificate support. As mentioned earlier, CoAP DTLS uses UDP port 5684.
- Small header: The CoAP overhead adds only 4 bytes.
- Fast response: When a client sends a CoAP GET to a server, the requested data is immediately returned in an ACK message, which is the fastest possible data exchange.
- Despite CoAP being designed for maximum efficiency, it is not a general replacement for HTTP.
 - It only supports a subset of HTTP capabilities and should only be used within IoT environments.
 - To interwork with HTTP, one can deploy an HTTP/CoAP proxy as a “gateway” device between the multi-service edge and smart device networks.

MQTT

- Message Queuing Telemetry Transport (not standardized)
- MQTT is, in a sense, the predecessor of CoAP in that it was created in 1999 and was specifically designed for lightweight, web-based, machine-to-machine communications. Like HTTP, it relies on TCP, except uses ports 1883 and 8883 for plain-text and TLS communications, respectively. Being based on TCP also implies a client/server model, similar to HTTP, but not necessary like CoAP. Compared to CoAP, MQTT is losing traction given the additional benefits specific to modern IoT networks that CoAP offers.

Table 25. CoAP, MQTT, and HTTP comparison

	CoAP	MQTT	HTTP
Transport and ports	UDP 5683/5684	TCP 1883/1889	TCP 80/443
Security support	DTLS via PSK/PKI	TLS via PSK/PKI	TLS via PSK/PKI
Multicast support	Yes but no encryption support yet	No	No
Lightweight	Yes	Yes	No
Standardized	Yes	No; in progress	Yes
Rich feature set	No	No	Yes

49.1.9. Migration

Migrating to IoT need not be swift. For example, consider an organization which is currently running a virtual private cloud infrastructure with some critical in-house applications in their private cloud. All remaining COTS applications are in the public cloud. Assume this public cloud is hosted locally by an ISP and is connected via an MPLS L3VPN extranet into the corporate VPN. If this corporation owns a large manufacturing company and wants to begin deploying various IoT components, it can begin with the large and immobile pieces.

The multi-service edge (access) network from the regional SP POP to the factory likely already supports Ethernet as an access technology, so devices can use that for connectivity. Over time, a corporate WLAN can be extended for 802.11 WIFI capable devices. Assuming this organization is not deploying a private 4G LTE network, sensors can immediately be added using cellular as well. The migration strategy towards IoT is very similar to adding new remote branch sites, except the number of hosts could be very large. The LAN, be it wired or wireless, still must be designed correctly to support all of the devices.

49.1.10. Environmental Impacts on the Network

Environment impacts are especially important for IoT given the scale of devices deployed. Although wireless technologies become more resilient over time, they remain susceptible to interference and other natural phenomena which can degrade network connectivity. Some wireless technologies are even impacted by rain, a common occurrence in many parts of the world. The significance of this with IoT is to consider when to use wired or wireless communication as for a sensor. Some sensors may even be able to support multiple communications styles in an active/standby method. As is

true in most networks, resilient design is important in ensuring that IoT-capable devices are operable.

Appendices

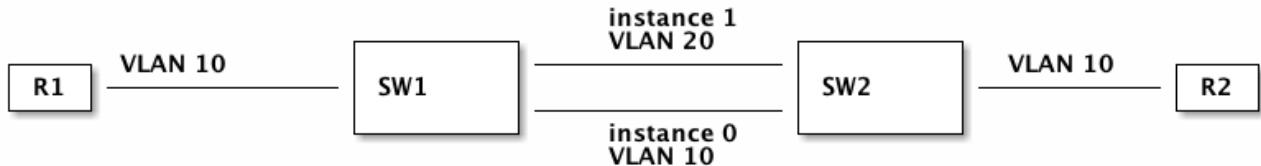
Chapter 50. Lab Equipment and IOS Releases

- Cisco ISR 2900 Series routers running IOS version 15.3T Universal Software release
- Catalyst 3560X Series switches running IOS version 15.0SE Universal (IP Services) Software release

Chapter 51. Labs

51.1. MST

- Source: <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24248-147.html>



- sw1 and sw2 connects with access ports each located in different VLANs
- SW1 and SW2 are in MST name CCIE revision 1
- SW1 is the root
- VLAN 10 is mapped to instance 0, VLAN 20 is mapped to instance 1
- Why R1 can not ping R2?

```
!R1 en conf t hostname R1 line con 0 logging sync int g0/1 ip addr 10.0.0.1 255.255.255.0 no shut end
```

```
!R2 en conf t hostname R2 line con 0 logging sync int g0/1 ip addr 10.0.0.2 255.255.255.0 no shut end
```

```
!SW1 en conf t hostname SW1 line con 0 logging sync int g0/0 shut int g0/1 swi access vlan 10 swi mode access int g0/2 swi access vlan 20 swi mode access span port-priority 64 int g0/3 swi access vlan 10 swi mode access ! span mode mst span mst configuration name CCIE revision 1 instance 1 vlan 20 end
```

```
!SW2 en conf t hostname SW2 line con 0 logging sync int g0/0 shut int g0/1 swi access vlan 10 swi mode access int g0/2 swi access vlan 20 swi mode access int g0/3 swi access vlan 10 swi mode access ! span mode mst span mst configuration name CCIE revision 1 instance 1 vlan 20 end
```

!! problem This issue is explained by the fact that MST information is conveyed with only one BPDU (IST BPDU), regardless of the number of internal instances. Individual instances do not send individual BPDUs. When Switch A and Switch B exchange STP information for VLAN 20, the switches send an IST BPDU with an MRecord for instance 1 because that is where VLAN 20 is mapped. However, because it is an IST BPDU, this BPDU also contains information for instance 0. This means that the IST instance is active on all ports inside an MST region, whether these ports carry VLANs mapped to the IST instance or not.

!! solution . use one instance for vlan 10 . allow vlan 10 in all ports

```
!SW1 en conf t hostname SW1 line con 0 logging sync int g0/0 shut int g0/1 swi access vlan 10 swi mode access int g0/2 swi access vlan 20 swi mode access span port-priority 64 int g0/3 swi access vlan 10 swi mode access ! span mode mst span mst configuration name CCIE revision 1 instance 1 vlan 20 instance 2 vlan 10 end
```

```
!SW2 en conf t hostname SW2 line con 0 logging sync int g0/0 shut int g0/1 swi access vlan 10 swi  
mode access int g0/2 swi access vlan 20 swi mode access int g0/3 swi access vlan 10 swi mode access  
! span mode mst span mst configuration name CCIE revision 1 instance 1 vlan 20 instance 2 vlan 10  
end
```

51.1.1. Two VLANs mapped to the same instance block the same ports

Chapter 52. IOS

Chapter 53. IOS-XE

- runs IOS as a process on top of Linux OS
- modular → extensibility
- high availability
 - Can upgrade one package without rebooting the whole device
- separates control plane (Forwarding and Feature Manager) and data plane (Forwarding Engine Driver)
 - The FFM provides a set of APIs used to manage the control plane processes. The resulting outcome is that the FFM programs the data plane through the FED and maintains all forwarding states for the system. It is the FED that allows the drivers to affect the data plane, and it is provided by the platform.

Cisco IOS XE consists of different modules called sub-packages that provide a specific function:

- RPBase: provides the operating system software for the route processor.
- RPControl: controls the control plane processes that interface between the IOS process and the rest of the platform.
- RPAccess: used for access to the router through protocols like SSH / SSL.
- RPIOS: provides the Cisco IOS kernel
- ESPBase: provides the ESP operating system and control processes, and the ESP software. The ESP (Embedded Services Processor) is responsible for the data plane and all flows through the data plane. It is also responsible for features/tasks like QoS, ACLs, VPNs, Netflow, NAT, etc.
- SIPBase: this controls the SIP operating system and control processes. A SIP (Shared Port Adapter Interface Processor) is a carrier card that you insert in a router slot. The SIP can hold one or more SPAs and it provides the connection between the route processor and SPA.
- SIPSPA: provides the SPA driver and Field Programmable Device (FPD). The SPA (Shared Port Adapter) is inserted in the subslot of a SIP and provides the interface between the network and SIP.

The complete image that has all sub-packages is called a consolidated package. This is the most simple solution since it's a single image file. It's also possible to run individual sub-packages, the advantage of this is that the router will only run the software that you require on your router so you will save some memory and your router will boot faster.

<http://dtdccie.blogspot.co.za/2014/08/differences-between-ios-and-ios-xe.html>

Chapter 54. IOS System Management

- Configuration Fundamentals Configuration Guide, Cisco IOS Release 15M&T
- Interface and Hardware Component Configuration Guide, Cisco IOS Release 15M&T
- Loading and Managing System Images Configuration Guide, Cisco IOS Release 15M&T
- Maintaining System Memory Configuration Guide, Cisco IOS Release 15M&T
- Managing Configuration Files Configuration Guide, Cisco IOS Release 15M&T
- Software Activation Configuration Guide, Cisco IOS Release 15M&T
- The Integrated File System Configuration Guide, Cisco IOS Release 15M&T

54.1. Configuration files

- startup-config
- running-config

```
# configure [terminal|memory|network]
```

copy {ftp: | rcp: | tftp:}system:running-config}

task: Create a configuration archive

```
conf t
archive
  path <url>
  maximum <number>
  time-period <minutes>
end
```

task: Save the running configuration file to the configuration archive

```
# archive config
```

task: Replace the running configuration file with a saved configuration file

```
# configure replace <source-url> [nolock][list][force][ignorecase][time
<minutes>][reverttrigger[error][timer <minutes>]]
```

source-url

URL of the saved Cisco IOS configuration file that is to replace the current running configuration

list

displays a list of the command lines applied by the Cisco IOS software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed.

force

replaces the current running configuration file with the specified saved Cisco IOS configuration file without prompting you for confirmation.

time

specify minutes within which you must enter the `configureconfirm` command to confirm replacement of the current running configuration file. If the `configureconfirm` command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the `configurereplace` command).

**nolock**

disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation.

ignorecase

allows the configuration to ignore the case of the confirmation command

reverttrigger

set the following triggers for reverting to the original configuration:

- `error`—Reverts to the original configuration upon error.
- `timerminutes`—Reverts to the original configuration if specified time elapses.

task: Cancel the timed rollback immediately

```
# configure revert now
```

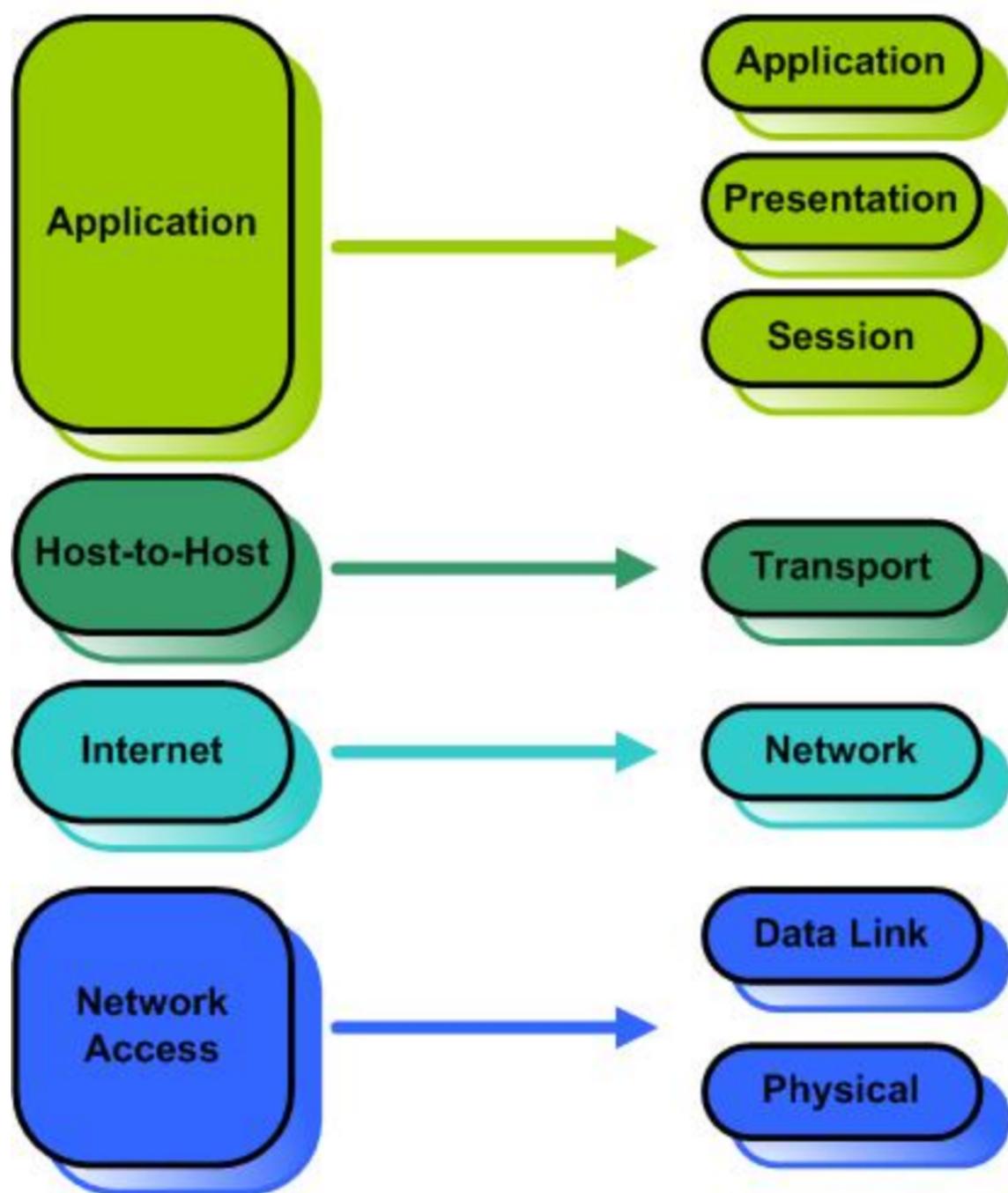
task: Specify a new revert time

```
# configure revert timer <minutes>
```

task: Set the maximum allowable time period of no activity before reverting to the saved configuration

```
# configure revert timer idle <minutes>
```

Chapter 55. General



Chapter 56. Notes To Self

2017-06-17 Sat 14:53

Do not follow the blueprint order for the book. Create a graph of technologies/concepts to master and start at the root (or from the node with less dependencies)

2017-06-18 Sun 04:52

- I am considering to bring chapter on security closed to the associate feature chapter.
- Example: DHCP snooping and DHCP, Dynamic ARP Inspection near ARP.