

**Enhanced Cutting-Edge Summary of
“CCNP ROUTE 642-902 Official
Certification Guide, by Wendell Odom”**

CCNP ROUTE Summary



Jean-Christophe Manciot

This is a summary of the '[CCNP ROUTE 642-902 Official Certification Guide, by Wendell Odom](#)'

With many improvements, *some annotations (beyond the scope of ROUTE and focused on modern production network)*,

A somewhat different layout and content,

With some excerpts from the '[CCDA 640-864 Official Cert Guide, by Anthony Bruno and Steve Jordan](#)'

by Jean-Christophe Manciot,

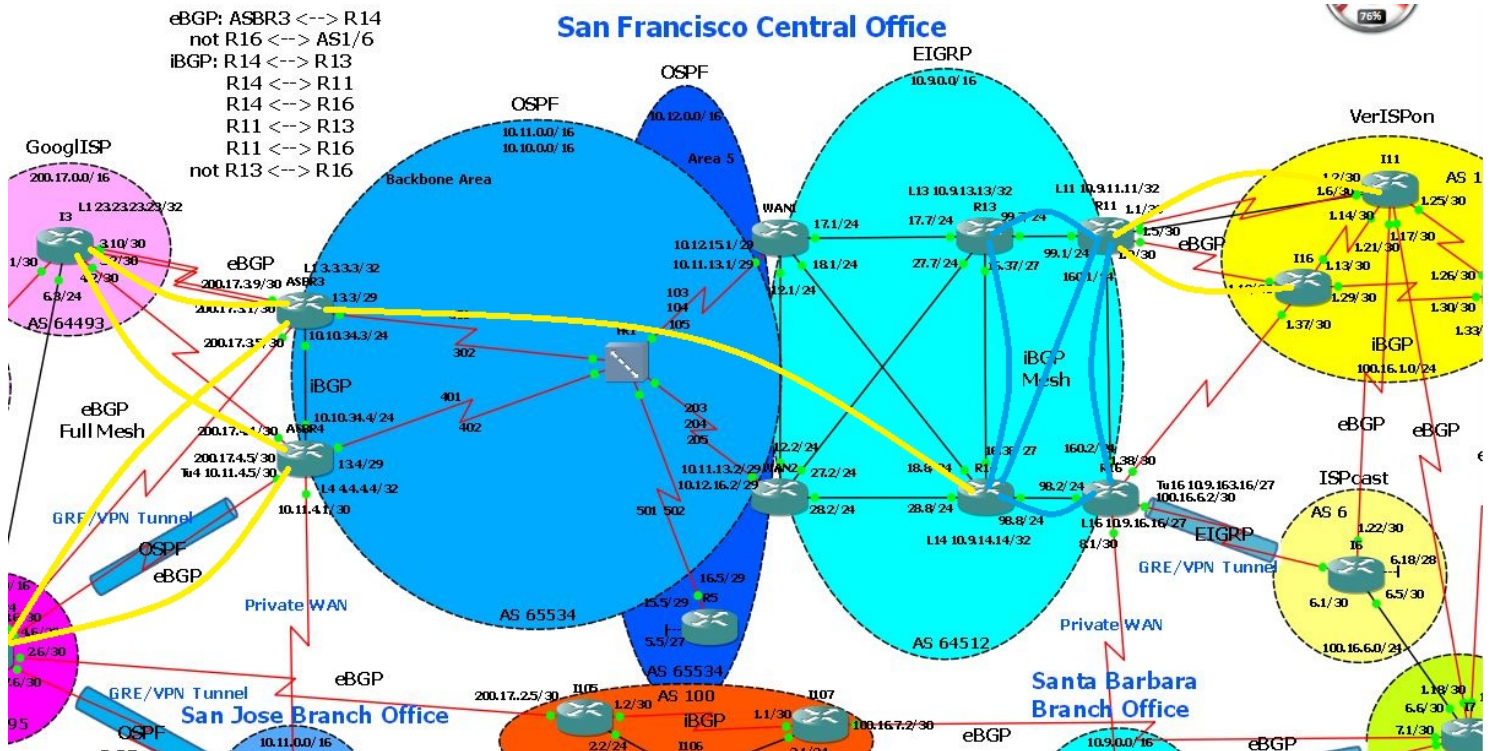


CCNP SWITCH Summary



Extract from [A Collection of GNS3 CCNP Labs](#)

San Francisco Central Office



➤ If you want to request the labs or join in the thread on Cisco Learning Network: [A collection of GNS3 CCNP Labs](#)

*Disclaimer: in order to pass the exam, your answers need to match the OCG, whatever its accurateness is.
The annotations in the Summary are more focused on production networks and innovation
Everyone has the possibility to write comments [here](#) to help improve this project.*

[ROUTE Exam practice](#)

Content at a Glance

1 Planning Tasks for the CCNP Exams.....	15
1.1 Typical Elements in an Implementation Plan.....	15
1.2 Structured Implementation Planning Methodologies.....	15
2 Tools Used Throughout this Document.....	16
2.1 Configuration Commands.....	16
2.2 Verification Commands.....	21
3 EIGRP.....	23
3.1 Establishing Neighborship.....	23
3.2 Exchanging Topology Information.....	25
3.3 Computing Routes with the DUAL Algorithm.....	26
3.4 Optimizing Router Resources.....	30
3.5 Avoiding External Routes Confusion: Router-ID.....	31
3.6 Summary.....	32
3.7 Configuration Commands.....	34
3.8 Verification Commands.....	39
4 OSPFv2.....	41
4.1 Hierarchically Segmenting the Topology.....	41
4.2 Establishing Neighborship.....	43
4.3 Exchanging Link State Database.....	46
4.4 Computing Routes.....	48
4.5 Optimizing Router Resources.....	50
4.6 Extending the Backbone Area with Virtual Links.....	50
4.7 Summary.....	52
4.8 Configuration Commands.....	55
4.9 Verification Commands.....	60
5 Path Control (Traffic Engineering).....	64
5.1 Route Redistribution.....	64
5.2 Redistribution into OSPF.....	65
5.3 Inter-Domain Routing Issues with Multiple Redistribution Points.....	68
5.4 Load-Balancing Summary.....	69
5.5 PBR.....	70
5.6 SLA.....	71
5.7 Configuration Commands.....	72
5.8 Verification Commands.....	80
6 BGP.....	81
6.1 Topological Designs.....	81
6.2 Reserved Values.....	85
6.3 Establishing Peering.....	86
6.4 Exchanging the BGP Table.....	89
6.5 Computing Routes.....	90
6.6 Clearing Peering.....	93
6.7 Summary.....	93

6.8 Configuration Commands.....	95
6.9 Verification Commands.....	101
7 IPv6.....	103
7.1 Hierarchical Prefix Assignment in the Internet.....	103
7.2 Header.....	104
7.3 Addresses.....	105
7.4 ICMPv6.....	111
7.5 NDP.....	111
7.6 DAD.....	112
7.7 IND.....	112
7.8 Migrating from IPv4 → IPv6.....	113
7.9 Routing Protocols.....	129
7.10 IGP Redistribution.....	134
7.11 IPv6 Comparison with IPv4.....	134
7.12 Guidelines.....	135
7.13 Summary of Features – RIPE-544.....	135
7.14 Configuration Commands.....	136
7.15 Verification Commands.....	142
8 Branch Internet Connections.....	146
8.1 Branch Office Broadband Internet Access.....	146
8.2 Dynamic Address Allocation.....	148
8.3 Securing Communications.....	148
8.4 Branch Routing.....	151
8.5 Sharing a few Public Addresses between Private Hosts to Access Internet: NAT/PAT	152
8.6 Protocol Stack on a router with Internet and VPN connections.....	153
8.7 Configuration Commands.....	154
8.8 Verification Commands.....	161

Table of Contents

1 Planning Tasks for the CCNP Exams.....	15
1.1 Typical Elements in an Implementation Plan.....	15
1.2 Structured Implementation Planning Methodologies.....	15
2 Tools Used Throughout this Document.....	16
2.1 Configuration Commands.....	16
2.1.1 For IPv4.....	16
2.1.1.1 ACLs.....	16
2.1.1.1.1 Standard and Extended ACLs.....	16
2.1.1.1.2 ACLs and IP Fragments.....	17
2.1.1.1.3 AS_Path ACLs.....	18
2.1.1.2 Prefix-lists.....	19
2.1.1.3 Route-maps.....	20
2.1.2 For IPv6.....	21
2.1.2.1 ACLs.....	21
2.1.2.2 Prefix-lists.....	21
2.1.2.3 Route-maps.....	21
2.2 Verification Commands.....	21
2.2.1 For IPv4.....	21
2.2.2 For IPv6.....	21
3 EIGRP.....	23
3.1 Establishing Neighborhood.....	23
3.1.1 Requirements.....	23
3.1.2 On WAN.....	24
3.1.2.1 Over Frame Relay.....	24
3.1.2.2 Over L3 MPLS VPN.....	24
3.1.2.3 Over Metro Ethernet.....	25
3.2 Exchanging Topology Information.....	25
3.2.1 Seeding the Topology Table.....	25
3.2.2 The Content of Update Message.....	26
3.2.3 Update Process.....	26
3.3 Computing Routes with the DUAL Algorithm.....	26
3.3.1 Overview.....	26
3.3.2 Metric Coefficients and Formula.....	27
3.3.3 New Metric Coefficients and Formula.....	28
3.3.4 Best Route: Successor.....	28
3.3.5 Backup Routes: Feasible Successors.....	28
3.3.6 No Best nor Backup Route: the Query Process.....	29
3.3.6.1 Converging by Going Active.....	29
3.3.6.2 Stuck-In-Active.....	29
3.3.6.3 Optimizing the Query Process: Stub Routers.....	29
3.4 Optimizing Router Resources.....	30
3.4.1 Limiting Reachability: Route Filtering.....	30
3.4.2 Without Limiting Reachability: Route Summarization.....	30

3.5 Avoiding External Routes Confusion: Router-ID.....	31
3.6 Summary.....	32
3.6.1 Features.....	32
3.6.2 Stack.....	33
3.6.3 Relations With Other Routing Protocols.....	33
3.6.4 “Opening” EIGRP.....	33
3.7 Configuration Commands.....	34
3.7.1 Dynamic Neighborhood.....	34
3.7.2 Controlling Neighborships.....	34
3.7.2.1 and Reducing Multicasts: Static Neighborhood.....	34
3.7.2.2 Preventing Unwanted Neighbors.....	34
3.7.2.2.1 Passive Interfaces.....	34
3.7.2.2.2 Authenticating EIGRP Messages.....	34
3.7.3 Optimizing Convergence delay.....	35
3.7.3.1 Tuning Hello/Hold Timers.....	35
3.7.3.2 Making Feasible Successors or Choosing One Route Instead of Another One: Metric Tuning □ Cf. Load-Balancing Summary.....	35
3.7.3.3 Optimizing the Query Process.....	36
3.7.3.3.1 Stub Routers.....	36
3.7.3.3.2 Active Timer.....	36
3.7.4 Optimizing Bandwidth Utilization.....	36
3.7.4.1 Load Sharing → Cf. Load-Balancing Summary.....	36
3.7.4.1.1 Equal Metric Paths: ECMP.....	36
3.7.4.1.2 Unequal Metric Paths: Variance - UCMP.....	36
3.7.4.2 Bandwidth Percentage Used for EIGRP Messages.....	36
3.7.5 Optimizing Router Resources.....	37
3.7.5.1 Limiting Reachability: Route Filtering.....	37
3.7.5.2 Without Limiting Reachability.....	37
3.7.5.2.1 Route Summarization.....	37
3.7.5.2.1.1 Manual Summarization.....	37
3.7.5.2.1.2 Auto-summary.....	37
3.7.5.2.2 Default Routes.....	38
3.7.5.2.2.1 Static Default Route.....	38
3.7.5.2.2.2 Static Default Classful Network.....	38
3.7.6 Avoiding External Routes Confusion : Router-ID.....	38
3.7.7 Summary of Issues related to EIGRP over Multipoint Frame Relay with a ‘Hub and Spokes’ Topology.....	38
3.8 Verification Commands.....	39
4 OSPFv2.....	41
4.1 Hierarchically Segmenting the Topology.....	41
4.1.1 LSA Types.....	41

4.1.2 Area Types.....	42
4.2 Establishing Neighborship.....	43
4.2.1 The Interface Network Types.....	43
4.2.2 Neighborship Requirements.....	44
4.2.3 On WAN.....	45
4.2.3.1 Over Frame Relay.....	45
4.2.3.2 Over L3 MPLS VPN.....	45
4.2.3.3 Over Metro Ethernet.....	46
4.3 Exchanging Link State Database.....	46
4.3.1 Seeding the LSDB.....	46
4.3.2 Message Types and Functions.....	46
4.3.3 Neighbor State Reference.....	46
4.3.4 DR/BDR Election.....	47
4.3.5 LSDB Exchange Process.....	47
4.3.5.1 Exchange with a DR/BDR.....	47
4.3.5.2 Classes of Neighborship.....	48
4.3.5.3 Area Flooding.....	48
4.3.5.4 Periodic Flooding.....	48
4.4 Computing Routes.....	48
4.4.1 Calculating the Cost of Intra-Area Routes.....	48
4.4.2 Calculating the Cost of Inter-area Routes.....	48
4.4.3 Calculating the Cost of External Routes.....	48
4.4.4 Special Rules Concerning Intra-area and Inter-area Routes on ABRs.....	48
4.4.5 Cost Components.....	49
4.4.6 Backup Routes: LFA Feasible Successors.....	49
4.5 Optimizing Router Resources.....	50
4.5.1 Limiting Reachability: Route Filtering.....	50
4.5.2 Without Limiting Reachability.....	50
4.5.2.1 Route Summarization.....	50
4.5.2.2 Default Routes and Stub Areas.....	50
4.6 Extending the Backbone Area with Virtual Links.....	50
4.7 Summary.....	52
4.7.1 Topology.....	52
4.7.2 Stack.....	53
4.7.3 Features.....	54
4.7.4 Relations With Other Routing Protocols.....	54
4.8 Configuration Commands.....	55
4.8.1 Defining the Interface Network Types.....	55
4.8.2 Dynamic Neighborship.....	55
4.8.3 Controlling Neighborships.....	55
4.8.3.1 ... and Reducing Multicasts: Static Neighborship.....	55
4.8.3.2 Preventing Unwanted Neighbors.....	55
4.8.3.2.1 Passive Interfaces.....	55
4.8.3.2.2 Authenticating OSPF Messages.....	56

4.8.4 Optimizing Convergence delay.....	56
4.8.4.1 Hello/Dead Timers.....	56
4.8.4.2 LFA Feasible Successors.....	56
4.8.5 Controlling DR/BDR Election.....	57
4.8.5.1 Interface Priority.....	57
4.8.5.2 Router ID.....	57
4.8.6 Optimizing Bandwidth Utilization.....	57
4.8.6.1 Load Sharing □□ Cf. Load-Balancing Summary.....	57
4.8.6.1.1 Equal Metric Routes.....	57
4.8.6.1.2 Making Equal Metric Routes ... or choosing One Route Instead of Another: Cost Components.....	57
4.8.6.2 Setting the MTU.....	57
4.8.7 Optimizing Router Resources.....	57
4.8.7.1 Limiting Reachability: Route Filtering.....	57
4.8.7.1.1 Filtering Type 3 LSA on ABRs.....	57
4.8.7.1.2 Filtering Routes Added to the Routing Table on Each Intended Router.....	58
4.8.7.1.3 Filtering Type 5/7 LSA on ASBRs.....	58
4.8.7.2 Without Limiting Reachability.....	58
4.8.7.2.1 Route Summarization.....	58
4.8.7.2.1.1 Manual Inter-Area Summarization on ABRs.....	58
4.8.7.2.1.2 Manual External Summarization on ASBRs.....	58
4.8.7.2.2 Default Routes and Stub Areas.....	59
4.8.7.2.2.1 Domain-wide on ASBRs.....	59
4.8.7.2.2.2 Area-wide on ABRs/ASBRs.....	59
4.8.7.2.2.2.1 Stubby and Totally Stubby Areas.....	59
4.8.7.2.2.2.2 NSSA and Totally NSSA Areas.....	59
4.8.8 Extending the Backbone Area with Virtual Links.....	60
4.8.9 Summary of Issues related to OSPF over Multipoint Frame Relay with a ‘Hub and Spokes’ Topology.....	60
4.9 Verification Commands.....	60
5 Path Control (Traffic Engineering).....	64
5.1 Route Redistribution.....	64
5.1.1 Reasons.....	64
5.1.2 Route Feedback.....	64
5.1.3 Tools.....	64
5.1.4 Methods.....	64
5.1.5 Seed/External Metrics.....	65
5.1.5.1 Configured Values.....	65
5.1.5.2 Default Values.....	65
5.1.6 Route Filtering at a Redistribution Point.....	65
5.2 Redistribution into OSPF.....	65

5.2.1 Determining the Next-Hop for External Routes.....	66
5.2.1.1 Intra-Area.....	66
5.2.1.1.1 E2/N2.....	66
5.2.1.1.2 E1/N1.....	66
5.2.1.2 Inter-Area.....	66
5.2.1.2.1 E2.....	66
5.2.1.2.2 E1.....	66
5.2.1.2.3 Special Case Involving a (T)NSSA ASBR for Inter-Area E2/E1.....	66
5.2.2 A Brief Comparison of E1 and E2 Routes:.....	67
5.2.3 External LSAs with Overlapping Subnets.....	67
5.3 Inter-Domain Routing Issues with Multiple Redistribution Points.....	68
5.3.1 Between 2 Domains.....	68
5.3.2 Between 3 Domains.....	68
5.3.2.1 Setting AD Values for All Routes.....	68
5.3.2.2 Setting Per-Route AD Values.....	68
5.3.2.3 Filtering Routes During Redistribution.....	68
5.3.2.4 Filtering Tagged Routes During Redistribution.....	69
5.3.2.5 Filtering Tagged Routing Updates During Reception.....	69
5.4 Load-Balancing Summary.....	69
5.4.1 With Routing Protocols.....	69
5.4.1.1 EIGRP.....	69
5.4.1.2 OSPF.....	69
5.4.1.3 BGP.....	69
5.4.2 With First Hop Redundancy Protocols.....	69
5.4.2.1 HSRP/VRRP.....	69
5.4.2.2 GLBP.....	69
5.4.3 Caveats.....	69
5.4.3.1 Fast Switching.....	69
5.4.3.2 CEF.....	69
5.5 PBR.....	70
5.6 SLA.....	71
5.7 Configuration Commands.....	72
5.7.1 Redistribution into EIGRP.....	72
5.7.2 Redistribution into OSPF.....	73
5.7.3 Redistribution Default Seed Metrics.....	74
5.7.4 Inter-Domain Routing with Multiple Redistribution Points.....	74
5.7.4.1 Between 2 domains.....	74
5.7.4.1.1 Setting AD Values for all routes.....	74
5.7.4.2 Between 3 domains.....	74
5.7.4.2.1 Setting AD Values for all routes.....	74
5.7.4.2.2 Setting Per-Route AD Values.....	74
5.7.5 RTM.....	75

5.7.6 Default Administrative Distances.....	76
5.7.7 PBR.....	76
5.7.8 SLA.....	77
5.7.8.1 SLA Operation.....	77
5.7.8.2 Tracking Object.....	78
5.7.8.2.1 Tracking IP SLA.....	78
5.7.8.2.2 Tracking an Interface.....	78
5.7.8.2.3 Tracking an IP Route.....	78
5.7.8.2.4 Regulating Flapping of the Tracking Object State.....	79
5.7.8.2.5 Static Route to Track an IP SLA Operation / IP Route / Interface.....	79
5.7.8.2.6 PBR to Track an IP SLA Operation / IP Route / Interface.....	79
5.7.8.2.7 FHRP to Track an IP SLA Operation / IP Route / Interface.....	79
5.8 Verification Commands.....	80
6 BGP.....	81
6.1 Topological Designs.....	81
6.1.1 Single-Homed designs.....	81
6.1.2 Dual-Homed Designs.....	81
6.1.2.1 Choosing One Path over Another for All Destinations Using Default Routes....	82
6.1.2.2 Choosing One Path over Another for Some Destinations Using iBGP.....	82
6.1.3 Single Multi-Homed Designs.....	83
6.1.4 Dual Multi-homed Designs.....	84
6.2 Reserved Values.....	85
6.2.1 Private IPv4 Address Blocks.....	85
6.2.2 Reserved IPv4 Address Registries.....	85
6.2.3 16-Bit ASN Assignment Categories.....	85
6.3 Establishing Peering.....	86
6.3.1 Message Types Summary.....	86
6.3.2 Peer States.....	86
6.3.3 Peering Requirements.....	87
6.3.4 eBGP.....	87
6.3.5 iBGP.....	88
6.3.5.1 In Transit AS.....	88
6.3.5.2 In Large Corporations.....	88
6.3.5.3 Policing Internal AS.....	88
6.3.5.4 QoS Policy Propagation on BGP.....	88
6.3.5.5 Carrying MPLS VPN Information.....	88
6.4 Exchanging the BGP Table.....	89
6.4.1 Seeding the BGP Table.....	89
6.4.2 Exchange Process: Advertising Rules.....	89
6.4.3 Convergence.....	89
6.4.4 New Multi-Paths Capability: ADD-PATH.....	89
6.5 Computing Routes.....	90
6.5.1 Path Attributes.....	90

6.5.2 Best Path Algorithm.....	91
6.5.3 RIB Failures.....	91
6.5.4 Avoiding Routing Loops when Forwarding Toward the Internet.....	92
6.6 Clearing Peering.....	93
6.7 Summary.....	93
6.7.1 Features.....	93
6.7.2 Stack.....	94
6.7.3 Relations With Other Routing Protocols.....	94
6.8 Configuration Commands.....	95
6.8.1 Controlling eBGP/iBGP Peerings.....	95
6.8.1.1 Static Peering.....	95
6.8.1.2 Shutting down the Peering.....	95
6.8.1.3 Authenticating BGP Messages.....	95
6.8.2 Seeding the BGP Table.....	96
6.8.2.1 Using network command.....	96
6.8.2.1.1 No Auto-summary by Default.....	96
6.8.2.1.2 The Effect of Auto-summary.....	96
6.8.2.2 Using Redistribution.....	96
6.8.3 Optimizing Convergence Delay.....	96
6.8.3.1 Keepalive/Hold Timers.....	96
6.8.4 Optimizing Bandwidth Utilization.....	96
6.8.4.1 Equal Path Load Sharing □□ Cf. Load-Balancing Summary.....	96
6.8.5 Choosing One Path Over Another.....	97
6.8.5.1 Influencing Enterprise's Outbound Routes.....	97
6.8.5.1.1 Using Administrative Weight.....	97
6.8.5.1.2 Using Local_Pref.....	97
6.8.5.1.3 Using AS_Path.....	98
6.8.5.2 Influencing Enterprise's Inbound Routes.....	98
6.8.5.2.1 Using AS_Path.....	98
6.8.5.2.2 Using MED.....	98
6.8.6 Optimizing Router Resources.....	99
6.8.6.1 Limiting Reachability: Route Filtering.....	99
6.8.6.1.1 Filtering Tools.....	99
6.8.6.1.2 Clearing Neighbors.....	100
6.8.6.2 Without Limiting Reachability.....	100
6.8.6.2.1 Route Summarization.....	100
6.8.6.2.2 Default Route.....	100
6.8.7 Synchronizing Routes Between iBGP and IGP.....	100
6.9 Verification Commands.....	101
7 IPv6.....	103
7.1 Hierarchical Prefix Assignment in the Internet.....	103

7.2 Header.....	104
7.3 Addresses.....	105
7.3.1 IANA Address Allocation.....	105
7.3.1.1 Assigned and Unassigned Prefixes.....	105
7.3.1.2 PA Address Space.....	105
7.3.1.3 PI Address Space.....	105
7.3.1.4 PI vs PA Address Spaces.....	106
7.3.2 Unicast Address Types.....	106
7.3.2.1 Global Unicast.....	106
7.3.2.2 Unique Local.....	106
7.3.2.3 Link Local.....	107
7.3.2.4 IPv6 Addresses with Embedded IPv4 Addresses.....	107
7.3.2.4.1 IPv4-Compatible IPv6 Address.....	107
7.3.2.4.2 IPv4-Mapped IPv6 Address.....	107
7.3.2.5 Address Assignment.....	107
7.3.2.5.1 Dynamic Configuration.....	108
7.3.2.5.1.1 RA Flags.....	108
7.3.2.5.1.2 DHCPv6.....	108
7.3.2.5.1.3 SLAAC.....	108
7.3.2.5.2 Static Configuration.....	109
7.3.2.5.3 Privacy Extensions Address.....	109
7.3.2.6 On-Link Determination.....	109
7.3.3 Anycast Addresses.....	110
7.3.4 Multicast Addresses.....	110
7.4 ICMPv6.....	111
7.4.1 Messages Types.....	111
7.4.2 Path MTU Discovery.....	111
7.5 NDP.....	111
7.6 DAD.....	112
7.7 IND.....	112
7.8 Migrating from IPv4 → IPv6.....	113
7.8.1 IPv4/IPv6 Dual Stacks.....	113
7.8.2 Tunneling IPv6 Over IPv4: IPv6 -> IPv6.....	113
7.8.2.1 Static Point-to-Point Tunnels.....	115
7.8.2.2 Dynamic Point-to-Multipoint Tunnels.....	116
7.8.2.2.1 6to4 Tunnels.....	117
7.8.2.2.1.1 ‘Reserved Mode’: Using 2002::/16 in the LANs.....	117
7.8.2.2.1.2 ‘Public Mode’: Using IPv6 Global Unicast Addresses in the LANs.....	118
7.8.2.2.2 6over4.....	118
7.8.2.2.3 ISATAP Tunnels.....	119
7.8.2.2.4 6RD Tunnels.....	120

7.8.2.2.5 6PE.....	123
7.8.2.3 Comparing IPv6 Over IPv4 Tunnels.....	123
7.8.3 Tunneling IPv4 Over IPv6: IPv4 -> IPv4.....	123
7.8.3.1 DSTM.....	123
7.8.3.2 DS-Lite.....	123
7.8.3.3 Lightweight 4over6.....	124
7.8.3.4 MAP-E.....	124
7.8.4 Translating IPv4 <-> IPv6.....	124
7.8.4.1 NAT-PT.....	124
7.8.4.2 NAT64/DNS64/SLB64: IPv6 -> IPv4.....	124
7.8.4.3 MAP-T: IPv6 -> IPv4.....	124
7.8.4.4 NAT46/SLB46/SIIT: IPv4 -> IPv6.....	124
7.8.4.5 Issues.....	125
7.8.5 IPv6 Deployment Models.....	126
7.8.5.1 Dual-Stack Model.....	126
7.8.5.2 Hybrid Model.....	126
7.8.5.3 Service Block Model.....	127
7.8.5.4 SIIT Model.....	127
7.8.5.5 IPv6 Deployment Model Comparison.....	128
7.9 Routing Protocols.....	129
7.9.1 RIPng.....	129
7.9.2 EIGRPv6.....	130
7.9.3 OSPFv3.....	131
7.9.3.1 Comparing OSPFv3 with OSPFv2.....	131
7.9.3.2 LSA Types.....	133
7.9.4 IS-IS for IPv6.....	133
7.9.5 MP-BGP, a.k.a BGP4+.....	133
7.9.6 Summary of all IPv4 and IPv6 IGP's Summary Route Metric.....	133
7.10 IGP Redistribution.....	134
7.11 IPv6 Comparison with IPv4.....	134
7.12 Guidelines.....	135
7.12.1 Enterprise Deployment.....	135
7.12.2 Data Center Operation.....	135
7.13 Summary of Features – RIPE-544.....	135
7.14 Configuration Commands.....	136
7.14.1 Addresses.....	136
7.14.2 Static Routing.....	136
7.14.3 Routing Protocols.....	137
7.14.3.1 RIPng.....	137
7.14.3.2 EIGRPv6.....	137
7.14.3.3 OSPFv3.....	138
7.14.4 Migrating from IPv4 → IPv6.....	138
7.14.4.1 Static Point-to-Point Tunnels.....	138
7.14.4.1.1 6in4 : Manually Configured Tunnels.....	138

7.14.4.1.2 GRE Tunnels.....	139
7.14.4.1.3 Comparing Point-to-Point Tunnels.....	139
7.14.4.2 Dynamic Point-to-Multipoint Tunnels.....	139
7.14.4.2.1 6to4 Tunnels.....	139
7.14.4.2.1.1 'Reserved Mode'.....	139
7.14.4.2.1.2 'Public Mode'.....	140
7.14.4.2.2 ISATAP Tunnels.....	141
7.14.4.2.3 6RD Tunnels.....	141
7.14.4.2.4 Comparing Multipoint Tunnels.....	141
7.14.5 IGP Redistribution.....	142
7.15 Verification Commands.....	142
8 Branch Internet Connections.....	146
8.1 Branch Office Broadband Internet Access.....	146
8.1.1 DSL.....	146
8.2 Dynamic Address Allocation.....	148
8.3 Securing Communications.....	148
8.3.1 IPsec VPN.....	148
8.3.1.1 IPsec Protocols.....	149
8.3.1.2 IPsec Direct Encapsulation.....	149
8.3.2 GRE Over IPsec.....	150
8.4 Branch Routing.....	151
8.4.1 Small Branches.....	151
8.4.2 Medium and Large Branches.....	152
8.5 Sharing a few Public Addresses between Private Hosts to Access Internet: NAT/PAT	152
8.6 Protocol Stack on a router with Internet and VPN connections.....	153
8.7 Configuration Commands.....	154
8.7.1 Branch Office Broadband Internet Access.....	154
8.7.1.1 PPP/ATM/DSL.....	154
8.7.2 Dynamic Address Allocation: DHCP Server.....	154
8.7.3 Sharing a few Public Addresses between Private Hosts to Access Internet: NAT/PAT.....	155
8.7.4 Securing Communications.....	155
8.7.4.1 IPsec VPN: 'Policy-based VPN'.....	155
8.7.4.1.1 ISAKMP Policy.....	155
8.7.4.1.2 IPsec Crypto Map.....	156
8.7.4.2 GRE Tunnel: 'Route-based VPN'.....	159
8.7.4.3 IPsec DMVPN.....	159
8.7.5 Summary.....	160
8.8 Verification Commands.....	161

1 Planning Tasks for the CCNP Exams

1.1 Typical Elements in an Implementation Plan

The existing network:	Router and switch hardware IOS versions and feature sets RAM and flash in each device Existing configurations IP Subnet and Addressing Plan, Assignments, and Conventions
Management:	Personnel and roles, contact information Assumptions and dependencies Required management sign-offs New tools, reporting, status update process
New project details:	Design goals (reference to design doc possibly) Hardware upgrades Software upgrades Timelines to make changes Specific configurations for each device Migration issues (assuming a subset of sites are implemented in any one change window) Network diagrams, possibly for each interim step during a migration
Project completion:	Final sign-off requirements Definitions of success Submission of revised site documentation, operational procedures, and any other permanent documentation

1.2 Structured Implementation Planning Methodologies

Method	Owner	Comment
FCAPS	ISO	Fault, Configuration, Accounting, Performance, and Security: This standard focuses on network and systems management. Implementation planning falls into the change management category.
ITIL	Great Britain	Information Technology Infrastructure Library: A set of best practices for systems management that has been widely used in the IT industry. It is managed by the government of Great Britain. www.itil-official-site.com .
TMN	ITU-T	Telecommunications Management Network: Created by the ITU-T's Study Group 4, this ongoing effort defines system management practices from the ITU. Originally based on FCAPS. www.itu.int .
Cisco Lifecycle Services	Cisco	Quoting the Cisco website: "The Cisco Lifecycle Services approach defines the activities needed to help you successfully deploy and operate Cisco technologies and optimize their performance throughout the lifecycle of your network."
PPDIOO	Cisco	Prepare, Plan, Design, Implement, Operate, Optimize: The popular acronym for the steps defined by Cisco Lifestyle Services.

**'Verify' is missing:
PPDIVOO ;)**

2 Tools Used Throughout this Document

2.1 Configuration Commands

2.1.1 For IPv4

2.1.1.1 ACLs

2.1.1.1.1 Standard and Extended ACLs

(config)#**access-list** *number* {**permit|deny**} *source-ip-address* [**log**]

- *number*: 1-99 / 1300-1999: *Standard ACLs*
- *source-ip-address*: **any** or **host** *source-ip-address* or *source-ip-address wildcard-mask*

(config)#**access-list** *number* {**permit|deny**} *protocol* *source-ip-address* [**eq|neq|lt|gt|range**] [*source-port*]
dest-ip-address [**eq|neq|lt|gt|range**] [*dest-port*]
[fragments] [**log**] [**established**] [**dscp dscp**]

- *number*: 100-199 / 2000-2699: *Extended ACLs*
- *protocol*: ip, icmp, udp, tcp, ospf, eigrp, ah, esp, gre...
- *source/dest-ip-address*: **any** or **host** *ip-address* or *ip-address wildcard-mask*
- *source/dest-port*: n° or name of UDP/TCP port
- **fragments**: this access list entry applies only to non-initial fragments of packets (no Layer 4 information available).
- **established**: if *protocol*==TCP: indicates an established connection. A match occurs if the TCP datagram has the ACK or RST control bits set. The non matching case is that of the initial TCP datagram from a connection.
- **dscp dscp**: 0-63 | af11-af43 | cs1-7 | ef | default

(config)#**ip access-list** {{**standard** | **extended**} {*access-list-name* | *access-list-number*}}

(config-std-nacl)#{**permit|deny**} *source-ip-address* [**log**]

(config-ext-nacl)#{**permit|deny**} *protocol* *source-ip-address* [**eq|neq|lt|gt|range**] [*source-port*]
dest-ip-address [**eq|neq|lt|gt|range**] [*dest-port*] [**fragments**] [**log**] [**established**]

Notes for “ip-address wildcard-mask” expression:

With Magic Number = 256 – Last byte of Subnet Mask (MN – 1 = size of subnet):

- *Last significant byte of Address must = $n \times \{Magic\ Number\}$ (i.e. : the address must be a subnet number)*
- *First non-zero byte of WildCard must = $p \times \{Magic\ Number\} - 1$ (avoid 128 – 254 range)*
- *To find the range of numbers matched by an ACL address and wildcard mask values, use the address field as the low end of the range, and simply add the address and wildcard mask to find the high end of the range.*
- *Wildcard masks do not have the Subnet masks rule to begin with a continuous serie of 1s followed by a continuous serie of 0s: bits values can change anywhere so it can be anything, such as 0.255.0.255 or 0.15.240.255...*

(config-if)#**ip access-group** {*acl-number* | *acl-name*} {**in** | **out**} Applies the ACL on the specified interface

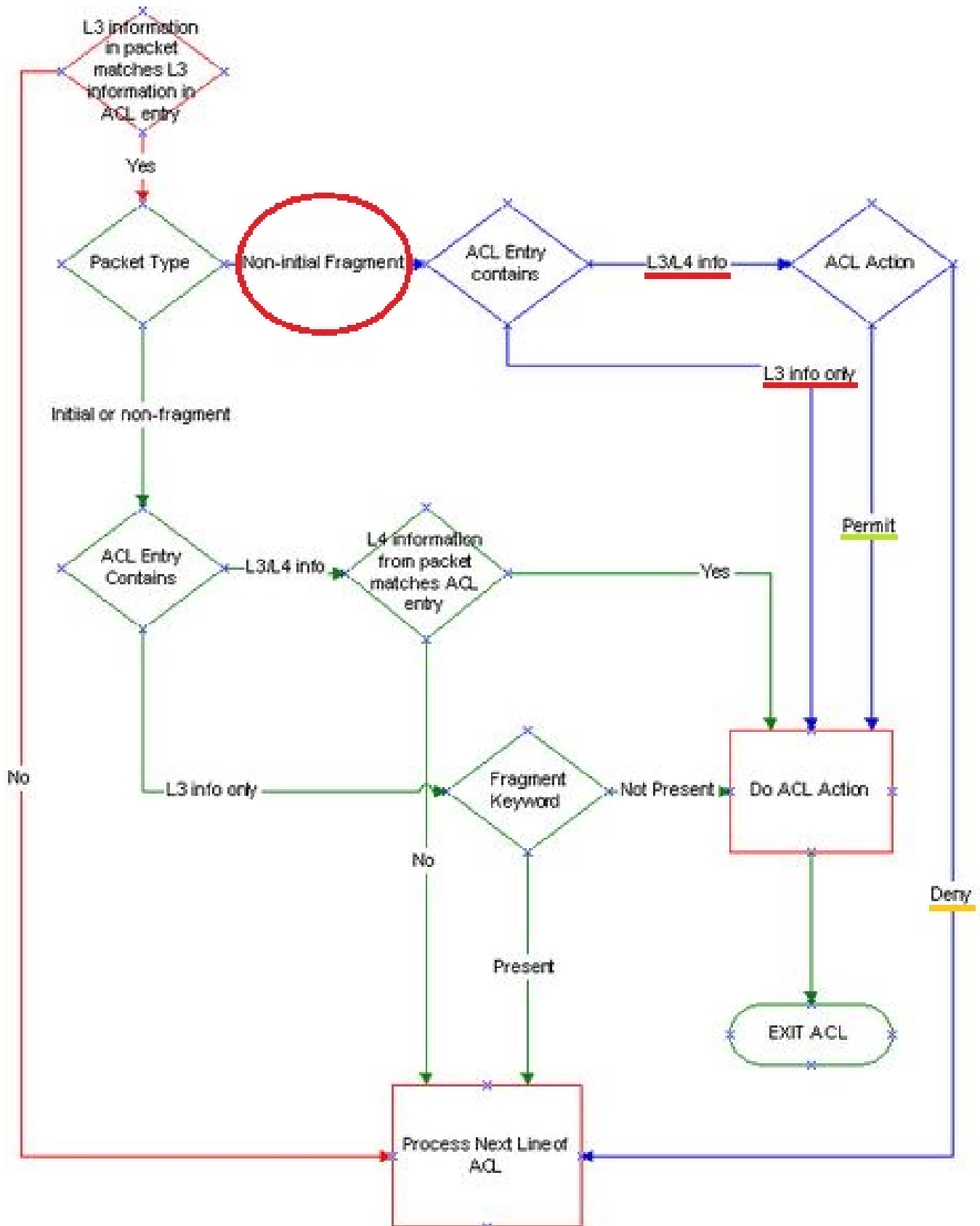
- *acl-number*: 1-99 / 1300-1999: *Standard ACLs*
- *acl-number*: 100-199 / 2000-2699: *Extended ACLs*
- *acl-name*: standard or extended ACL

2.1.1.1.2 ACLs and IP Fragments

The following flowchart illustrates the ACL rules when non-fragments, initial fragments, and non-initial fragments are checked against the ACL.

Note: The non-initial fragments themselves contain only Layer 3, never Layer 4 information, although the ACL may contain both Layer 3 and Layer 4 information.

The interesting and tricky part relates to how ACL rules with L3/L4 info and green underlined permit keyword are applied to non-initial fragments: letting them through in spite of the lack of L4 information in the packet may let fragments which are sourced/destined to another port pass directly through the router; this is a strange behavior.



2.1.1.1.3 AS_Path ACLs

(config)#ip as-path access-list as-path-access-list-number {permit | deny} regexp

Examples of regexp

<u>Beginning with this AS</u>	<u>^asn</u>
<u>Ending with this AS</u>	<u>asn\$</u>
<u>Only this AS</u>	<u>^asn\$</u>
<u>Containing this AS</u>	<u>asn</u>
<u>All AS</u>	<u>.</u> *
<u>Only locally originated networks (no asn)</u>	<u>^\$</u>
<u>Only networks originated from directly connected AS neighbors</u>	<u>^[0-9]+\$</u>
<u>All private AS</u>	<u>_6451[2-9]_ _645[2-9][0-9]_ _64[6-9][0-9][0-9]_ _65[0-4][0-9][0-9]_ _655[0-2][0-9]_ _6553[0-4]_</u>

Regular Expressions, from [Google Docs](#):

Expression	Description	Example	Matches	Does not match
.	A period signifies any character in the given position.	d.	do, dog, dg, ads	fog, jog
*	An asterisk after a character signifies a search for that preceding character repeated 0 or more times.	do*g	dog, dg, dooog	dOg, doug
+	A plus after a character signifies a search for that character displayed 1 or more times.	do+g	dog, dooog	dg, dOg, doug
?	The previous expression is optional.	do?g	dg, dog	dOg, doug
^	A caret must be placed at the beginning of a regular expression and signifies that the string starts with the character(s) or sequence placed after the caret.	^[dh]og	dog, hog	A dog, his hog
\$	A dollar sign must be placed at the end of a regular expression and signifies that the string ends with the character(s) or sequence placed before the dollar sign.	[dh]og\$	dog, hog, hot dog	dogs, hog, doggy
{A, B}	The previous expression is repeated between A and B times, where A and B are numbers.	d(o{1,2})g	dog, doog	dg, dooog, dOg
[x], [xa], [xa5]	A character set indicates that just one of the given character(s) should occur in the current position. For the most part, any characters are valid within brackets, including characters mentioned previously in expressions: [xa,\$5Gg.]	d[ou]g	dog, dug	dg, dOg, dooog
[a-z]	A character set range signifies a search for a character within the given range of characters. Common ranges include a-z, A-Z, and 0-9. Ranges can be combined into a single range: [a-zA-Z0-9]. Ranges can also be combined with character sets (mentioned previously): [a-zA-Z,&*].	d[o-u]g	dog, dug, dpq, drg	dg, dOg, dag
[^a-fDEF]	A character set beginning with a ^ signifies a search for a character that is not within the given set.	d[^aeu]g	dog, dOg, dig, d\$g	dg, dag, deg, dug
\s	Any white space character.	d\s g	d g, d[TAB]g	dg, dog, doug

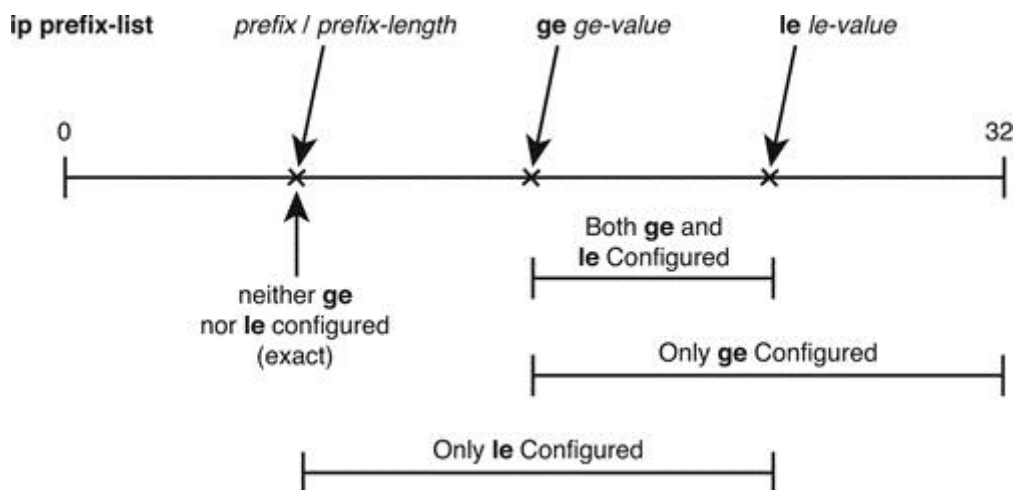
NOTE: When trying to search for actual instances of any character that has a specific meaning in regular expressions, like ^ and \$, you need to "escape" the character in your search query by placing a backslash in front of it. For example, if you wanted to search for an instance of the \$ character, you'd write \\$.

<https://t.me/learningnets>

2.1.1.2 Prefix-lists

(config)#**ip prefix-list** *list-name* [**seq** *seq-value*] {**permit**|**deny** *prefix/prefix-length*} [**ge** *ge-value*] [**le** *le-value*]

- the *list-name* is case sensitive,
- a *sequence number* can be used to allow later deletion of individual commands and insertion of commands into a particular sequence position.
- The following statements summarize the logic:
 - Step 1: The route prefix must be within the *range of addresses* implied by the prefix-list command's prefix/prefix-length parameters.
 - Step 2: The route prefix length must match the *range of prefix lengths* implied by the prefix-list command's prefix-length, ge (*greater-than-or-equal-to*), and le (*less-than-or-equal-to*) parameters (with $ge \leq pl \leq le$).
- One of four configuration combinations affect the logic of matching prefix lengths, as listed in the following tables:



- 0.0.0.0/0: matches the default route
- 0.0.0.0/0 le 32: matches all routes
- **There's an implicit deny any statement as its last match condition.**
- Using IP prefix lists for route filtering has several advantages over ACLs :
 - IP prefix lists allow matching of the prefix length, whereas the ACLs used by the EIGRP **distribute-list** command cannot in many configurations (*standard ACLs and IGP distribute-lists even with extended ACLs*),
 - The internal processing of the IP prefix lists uses an internal tree structure that results in *faster matching of routes* as compared with ACLs.

2.1.1.3 Route-maps

(config)#**route-map** *map-name* [**permit|deny**] [*sequence-number*]

(config-route-map)# **match** *keyword*

All match commands must pass

(config-route-map)# **set** *keyword*

Summary of Match Command Options (all topics)

match Command	Description
match interface <i>interface-type interface-number</i> [... <i>interface-type interface-number</i>]	Looks at outgoing interface of routes
*match ip address [{ <i>access-list-number</i> <i>access-list-name</i> } <i>prefix-list prefix-list-name</i>]	Examines route destination prefix and prefix length
* match ip next-hop [<i>access-list-number</i> <i>access-list-name</i>]	Examines route's next-hop address
* match ip route-source [<i>access-list-number</i> <i>access-list-name</i>]	Matches advertising router's IP address
match metric <i>metric-value</i> [+ deviation]	Matches route's metric, or a range (plus/minus the configured deviation)
match route-type [internal external [type-1 type-2] level-1 level-2] nssa-external	Matches route type
match tag <i>tag-value</i> [... <i>tag-value</i>]	Matches the route tag, which requires that another router has earlier set the tag

* Can reference multiple *standard and extended - numbered and named ACLs on a single match command*

PBR → **match length** *min max* Matches the IP length of a packet

BGP → **match as-path** *as-path-acl-number* Matches the AS_PATH attribute

- With an *extended* ACL, IOS compares the *source-ip-address* parameter to the *subnet number* of the route and the *dest-ip-address* to the *subnet mask* of the route.

Summary of Set Command Options (all topics)

PBR {	set interface <i>int ... int</i>	Sets first UP outgoing interface
	set default interface <i>int ... int</i>	Replaces Routing Table's default route
	set ip next-hop <i>ip-address ... ip-address</i>	Sets first next-hop addresses for which the associated interface are UP
	set ip default next-hop <i>ip-addr ... ip-addr</i>	Replaces Routing Table's default route

RIP
OSPF → **set metric** *metric-value* Sets the route's metric for OSPF, RIP, and IS-IS

IS-IS
EIGRP → **set metric** *bandwidth delay reliability loading mtu* Sets the EIGRP route's metric values

OSPF → **set metric-type** [Internal | external [type-1|type-2]] Sets type of route for OSPF

set tag *tag-value* Sets the unitless tag value in the route

set ip precedence *value* Sets ancient precedence (first TOS 3 bits), (*now first DS 6 bits Differentiated Service Code Point – DSCP are used*)

set ip tos *value* Sets ancient TOS byte (*now called DS byte*)

BGP { **set weight** *value* Sets Cisco-proprietary Weight

set local-preference *value* Sets PA LOCAL_PREF

set as-path prepend *asn asn asn...* Inserts other ASs in PA AS_PATH
Only for eBGP sessions

2.1.2 For IPv6

2.1.2.1 ACLs

(config)#ipv6 access-list access-list-name

Defines an IPv6 ACL.

- Does not support numbered ACLs.
- An IPv4 ACL and an IPv6 ACL cannot share the same name.
- Every IPv6 ACL has implicit “permit icmp any any nd-na”, “permit icmp any any nd-ns”, and “deny ipv6 any any” statements as its last match conditions. (The former two match conditions allow for ICMPv6 neighbor discovery.)
- IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

(config-ipv6-acl)#{permit|deny} protocol source-ip-address [eq|neq|lt|gt|range] [source-port] dest-ip-address [eq|neq|lt|gt|range] [dest-port] [fragments] [log] [established] [dscp dscp]

- protocol: **ipv6, icmp, udp, tcp, ospf, eigrp, ahp, esp, gre...**
- source/dest-ip-address: **any or host ip-address or ip-address/len**
- source/dest-port: **n° or name of UDP/TCP port**

(config-if)#ipv6 traffic-filter access-list-name {in | out}

Applies this ACL to that interface.

2.1.2.2 Prefix-lists

(config)#ipv6 prefix-list list-name [seq seq-value] {permit|deny prefix/prefix-length} [ge ge-value] [le le-value]

- The **ipv6 prefix-list** command is similar to the **ip prefix-list** command, except that it is IPv6-specific.
- There's an implicit deny any statement as its last match condition.
- **::/0:** matches the default route
- **::/0 le 128:** matches all routes

2.1.2.3 Route-maps

(config)#route-map map-name [permit|deny] [sequence-number]

(config-route-map)# match keyword

All match commands must pass

(config-route-map)# set keyword

- The keywords are similar to the IPv4 route-map command, except that they are IPv6-specific (**ip** → **ipv6**).
- Any route that does not match at least one match clause relating to a route-map command will be ignored.

2.2 Verification Commands

2.2.1 For IPv4

#show ip access-list [number | name] Lists ACL configuration and counters that increment each time IOS matches an entry.

#show ip prefix-list detail [name] Lists prefix configuration and counters that increment each time IOS matches an entry.

#show route-map Shows overall statistics for the number of packets matching the route map for PBR purposes.

2.2.2 For IPv6

#show ipv6 access-list [number | name] Lists ACL config. and counters that increment each time IOS matches an entry.

#show ipv6 prefix-list detail [name] Lists prefix config. and counters that increment each time IOS matches an entry.

#show route-map Shows overall statistics for the number of packets matching the route map for PBR purposes.

<https://t.me/learningnets>

3 EIGRP

[Enhanced Interior Gateway Routing Protocol](#), Cisco

3.1 Establishing Neighborhood

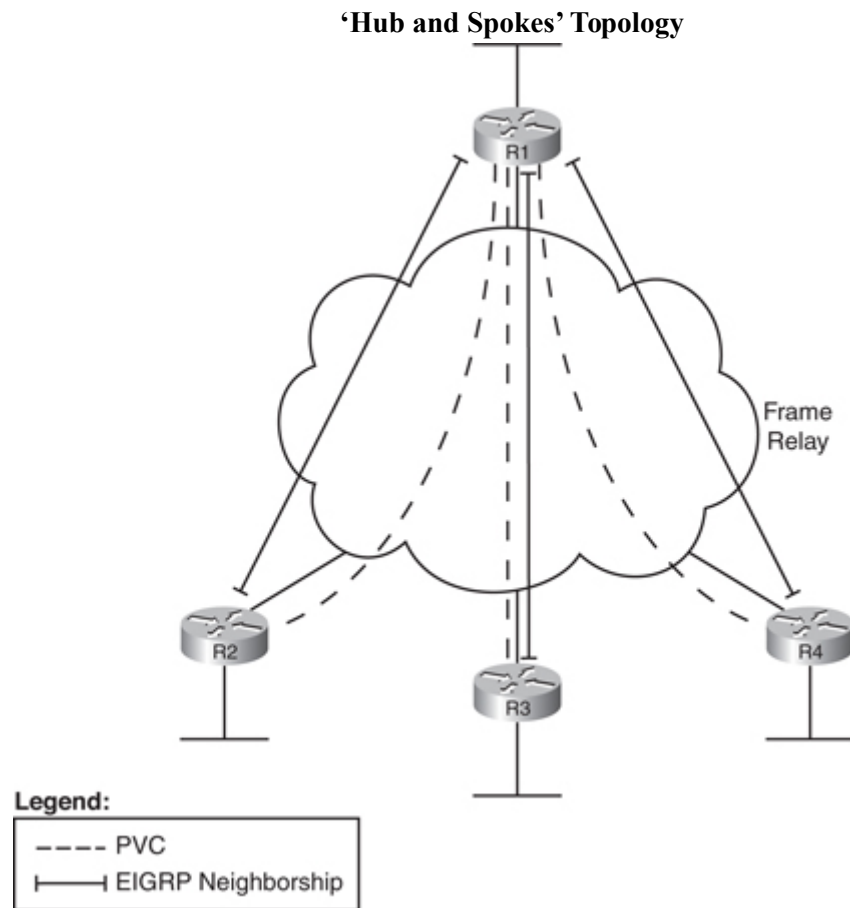
3.1.1 Requirements

	Requirement	EIGRP	OSPF
	The routers must be able to send/receive IP packets to one another.	Yes	Yes
*	Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
*	Must not be passive on the connected interface.	Yes	Yes
*	Must use the same ASN (EIGRP) or process-ID (OSPF) on the router configuration command.	Yes	No
	Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	No	Yes
*	Must pass neighbor authentication (if configured).	Yes	Yes
	Must be in same area.	N/A	Yes
	IP MTU must match.	No	Yes
*	K-values (used in metric calculation) must match.	Yes	N/A
	Router IDs must be unique.	No ¹	Yes

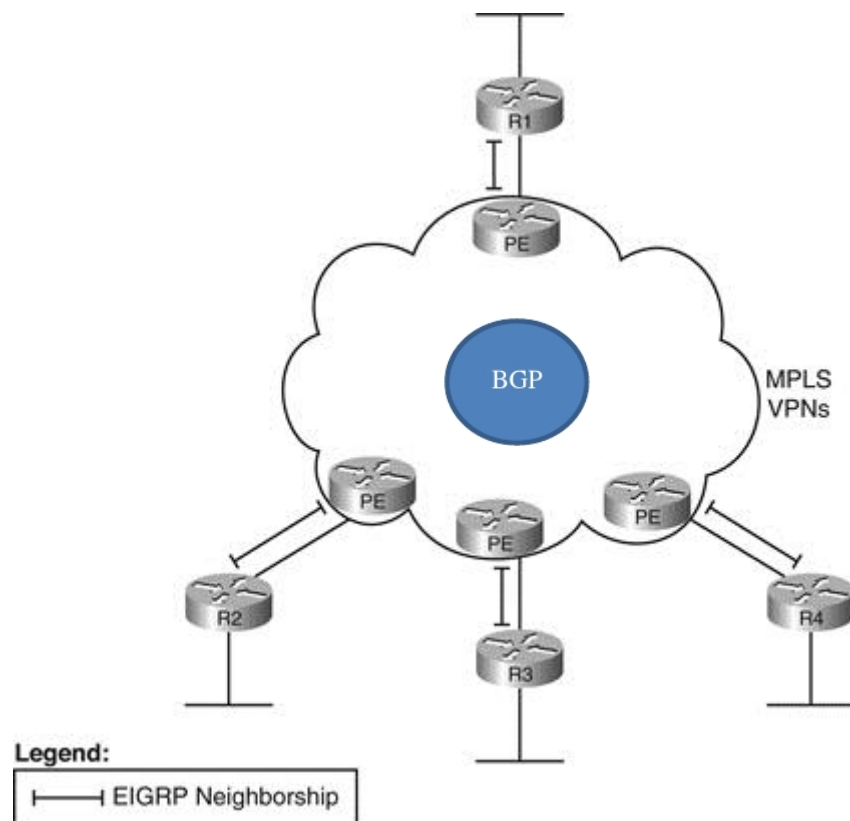
¹Duplicate EIGRP RIDs do not prevent routers from becoming neighbors, but it can cause problems when adding external EIGRP routes to the routing table.

3.1.2 On WAN

3.1.2.1 Over Frame Relay



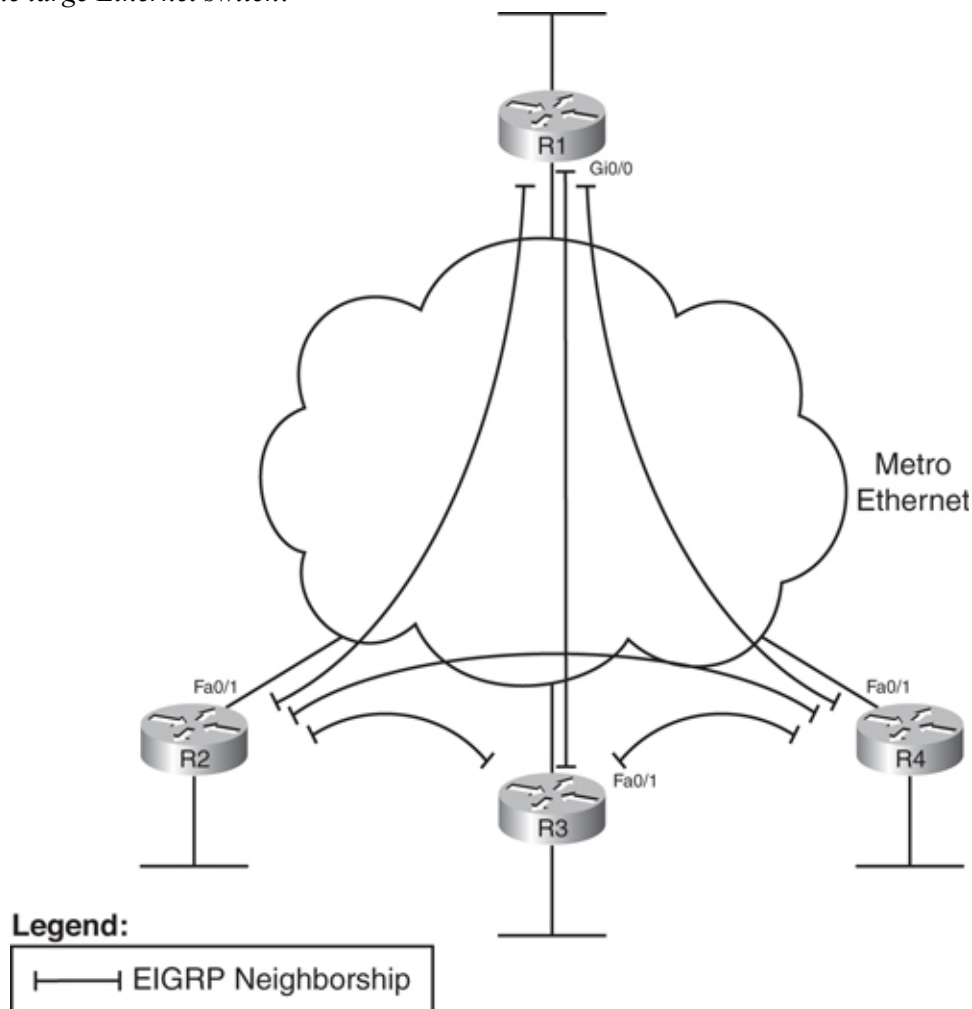
3.1.2.2 Over L3 MPLS VPN



3.1.2.3 Over Metro Ethernet

MetroE encompasses several underlying technologies to create the service. Both technical specifications allow for connections using Ethernet links, with the service forwarding Ethernet frames. Because it is a Layer 2 service, MetroE does not have any Layer 3 awareness, and the customer routers (typically referenced as with the more general service provider term *customer premise equipment*, or CPE) see the *MetroE service as a VLAN*. Because the customer routers connect to the service as a VLAN, all the routers connected to the service can become EIGRP neighbors.

- *Virtual Private Wire Service (VPWS) focuses on point-to-point topologies,*
- *Virtual Private LAN Service (VPLS) supports multipoint, approximating the concept of the entire WAN service acting like one large Ethernet switch.*



3.2 Exchanging Topology Information

3.2.1 Seeding the Topology Table

Before a router can send EIGRP topology data to a neighbor, that router needs to add *topology information* (same as content of Update message) for some prefixes in its topology table to get the process started. A router's EIGRP process adds subnets to its local topology table, without learning the topology data from an EIGRP neighbor, from two sources (static configuration) :

- The default route and prefixes of connected subnets for interfaces on which EIGRP has been enabled on that router using the **network** command,
- Prefixes learned by the **redistribution** of routes into EIGRP from other routing protocols or routing information sources.

Each router then adds any dynamically learned prefix information to their topology table, and then that router advertises the new information to other neighbors. Eventually, all routers in the EIGRP domain learn about all prefixes—unless some other feature, such as route summarization or route filtering, alters the flow of topological information.

3.2.2 The Content of Update Message

EIGRP uses seven protocol messages to do its work, and two of them are part of the topology data exchange process: Update and Ack :

- **Hello**
- **Update**: *topological information* :
 - Prefix
 - Prefix length
 - Metric components: min bandwidth (in kbits/s), total delay (in tens-of-microsecond), max load and min reliability
 - Offset list
 - Non metric items: min MTU, tag, next-hop and hop count (+1)
- **Ack** : acknowledges receipt of the Update
- **Query**
- **Reply**
- **SIA-Query**
- **SIA-Reply**

3.2.3 Update Process

When EIGRP neighbors first become neighbors, they begin exchanging topological information using Update messages using these rules:

- When a neighbor first comes up, the routers exchange *full updates*, meaning the routers exchange all topology information.
 - After all prefixes have been exchanged with a neighbor, the updates cease with that neighbor if no changes occur in the network. There is no periodic reflooding of topology data.
 - If something changes—for example, one of the metric components change, links fail, links recover, new neighbors advertise additional topology information—the routers send *partial updates* about only the prefixes whose status or metric components have changed.
 - If neighbors fail and then recover, or new neighbor adjacencies are formed, full updates occur over these adjacencies.
 - EIGRP uses *Split Horizon* rules on most interfaces by default, which impacts exactly which topology data EIGRP sends during both full and partial updates.
- *Split Horizon* limits the prefixes that EIGRP advertises out an interface. Specifically, if the currently best route (Successor) for a prefix lists a particular *outgoing interface*, Split Horizon means that EIGRP will not include that prefix in the Update sent out that same interface.

3.3 Computing Routes with the DUAL Algorithm

3.3.1 Overview

DUAL, the **Diffusing Update ALgorithm**, is the algorithm used by Cisco's EIGRP routing protocol to ensure that a given route is recalculated globally whenever it might cause a routing loop. DUAL responds to changes in the routing topology and dynamically adjusts the topology and routing tables of the router automatically.

EIGRP uses a **feasibility condition** to ensure that only loop-free backup routes are ever selected. The feasibility condition is conservative: when the condition is true, no loops can occur, but the condition might under some circumstances reject all routes to a destination although some are loop-free.

When no feasible route to a destination is available, the DUAL algorithm invokes a **Diffusing Computation** to ensure that all traces of the problematic route are eliminated from the network. At which point the normal **Bellman–Ford algorithm** is used to recover a new route.

3.3.2 Metric Coefficients and Formula

- **BW**: Slowest bandwidth on the path in kbits/s, *weighted by K1*
- **Delay**: Sum of all delays on the path in tens of μ s, *weighted by K3*
- **Load**: Maximum load on the path. *weighted by K2*; its value is *between 1 and 255* (255 represents 100%)
- **Reliability**: Minimum link reliability on the path, *weighted by K4 and K5*; its value is *between 1 and 255* (255 represents 100%)

The metric weights K1, K2, K3, K4 and K5 define which components are used in the metric calculation of an interface. By default, only K1 and K3 are set to 1, which means that only the first two components are used, with $10^n = 10^n$

$$\text{Metric} = 256 * \left[K1 \left(\frac{10^7}{\text{BW}} \right) + \frac{K2 \left(\frac{10^7}{\text{BW}} \right)}{(256 - \text{Load})} + K3(\text{delay}) \right] * \left[\frac{K5}{(\text{Reliability} + K4)} \right]$$

$$\text{Default Metric} = 256 * \left\{ \frac{10^7}{\text{slowest bandwidth (kbits/s)}} + \sum_{k=0}^n \text{delays (10 * } \mu\text{s)} \right\}$$

Out of the scope of CCNP ROUTE: From [Ivan Pepelnjak's Blog](#) :

- *EIGRP routing updates are triggered only by a change in network topology (interface up/down event, IP addressing change or configured bandwidth/delay change) and not by change in interface load or reliability. The load/reliability numbers are thus a snapshot taken at the moment of the topology change and should be ignored. This explains the inability of EIGRP to adapt to changing load/reliability conditions.*
- *Sending EIGRP updates whenever there's a significant change in load or reliability would be technically feasible, but would diminish the benefits of replacing distance vector behavior with DUAL.*

Actually, EIGRP is in essence an enhanced distance vector routing protocol. The fact that it does not send updates at regular intervals does not transform its nature. This merely saves bandwidth and it sends updates when it is only appropriate, that is when changes (interface up/down event, IP addressing change or configured bandwidth/delay change) are detected.

So I don't see any reason why "Sending EIGRP updates whenever there's a significant change in load or reliability ... would diminish the benefits of replacing distance vector behavior with DUAL."

I feel this would be a great enhancement to this routing protocol, since this simple ability to detect loaded or unreliable networks would be very valuable to medium-sized networks. It would require the implementation of 2 thresholds to trigger the sending of route updates to avoid network instability, with the help of two delaying timers, OVER and BELOW, like the UP and DOWN object tracking timers:

- *load threshold: when the interface load gets over this percentage, send a routing update to reflect the change, once the OVER timer has elapsed,*
- *reliability threshold: when the interface reliability gets under this percentage, send a routing update to reflect the change, once the BELOW timer has elapsed.*

It would be the responsibility of the network engineer to correctly tweak these values to avoid many route flaps.

To join in CLN thread: "[EIGRP load/reliability metric components](#)"

- *MTU is a tie-breaker. If you have more than "maximum-paths" equal-cost EIGRP paths, the ones with the highest MTU value are used.*

3.3.3 New Metric Coefficients and Formula

Out of the scope of CCNP ROUTE: from current [EIGRP Savage's Draft](#)

- **Throughput:** Minimum throughput on the path in *kbits/s*, **weighted by K1**
- **Latency:** Sum of all *one-way delays on the path in ps* (picoseconds, 10^{-12} s), **weighted by K3**
- **Load:** Maximum load on the path. **weighted by K2**; its value is *between 1 and 255* (255 represents 100%)
K2 has the greatest effect on the metric when the load increases beyond **90%**.
- **Reliability:** Minimum link reliability on the path, **weighted by K4 and K5**; its value is *between 1 and 255*
- **Jitter and Energy:** extended attributes, sum of all jitter in μ s and energy in power per kbytes per second of usage on the path, **weighted by K6**

The metric weights K1, K2, K3, K4, K5 and K6 define which components are used in the metric calculation of an interface. If K5 is 0, then the whole related quotient is defined as "1".

The default metric has slightly changed, when only K1 and K3 are set to 1, which means that only the first two components are used.

$$\text{Metric} = [(K1 * \text{Net}_{\text{throughput}}) + \text{Latency} + (K6 * \text{ExtAttr})] * \left\{ \frac{K5}{(K4 + \text{Min_reliability})} \right\}$$

$$\text{Net}_{\text{throughput}} = \text{Max}_{\text{throughput}} + \left\{ \frac{K2 * \text{Max}_{\text{throughput}}}{256 - \text{Max_load}} \right\}$$

$$\text{Max}_{\text{throughput}} = K1 * \frac{65536 * 10^7}{\text{Min_bandwidth (kbits/s)}}$$

$$\text{Latency} = K3 * \frac{65536 * \sum_{k=0}^n \text{delays (ps)}}{10^6}$$

$$\text{Default Metric} = 65536 * \left\{ \frac{10^7}{\text{Min_bandwidth (kbits/s)}} + \frac{\sum_{k=0}^n \text{delays (ps)}}{10^6} \right\}$$

3.3.4 Best Route: Successor

- **Feasible Distance (FD):** Integer metric for the route, from the local router's perspective, used by the local router to choose the best route for that prefix based on the lowest metric → **Successor**

3.3.5 Backup Routes: Feasible Successors

- **Reported Distance (RD)/Advertised Distance (AD):** Integer metric for the route, from the neighboring router's perspective (the neighbor that told the local router about the route). Used by the local router to calculate the FD and when converging to new routes → **Feasible Successors**
- **Feasibility condition:** if a Non-Successor route's RD < Successor's FD, the route is a *Feasible Successor* route. This condition is *sufficient*, but *not necessary*: there might be other loop-free routes which do not match this condition.
- **Out of the scope of CCNP ROUTE:** in order to consider the feasibility condition " $\leq$ " as OK, we would need to enhance IP routing stack to prevent routing loop in some unfavorable situations.
- **To join in CLN thread:** [Follow-up : Proposal for a new EIGRP "Enhanced Feasibility Condition" to allow faster convergence](#).

3.3.6 No Best nor Backup Route: the Query Process

3.3.6.1 Converging by Going Active

When EIGRP removes a Successor route and no FS route exists, the router begins a process by which the router discovers if any loop-free alternative route reaches that prefix. This process is called *going active* on a route. Routes for which the router has a successor route, and no failure has yet occurred, remain in a *Passive* state. Routes for which the Successor route fails, with no Feasible Successor routes, move to an *Active* state, as follows:

- Change the state, from Passive (P) to Active (A).
- *Multicasts* EIGRP *Query* messages to every **non-stub** neighbor except the neighbor in the failed route. The Query asks a neighbor whether it has a loop-free route for the listed prefix/length.
- The neighbor considers itself to have a loop-free route if that neighbor is *Passive* for that prefix/length or for a *Summary* route that includes the prefix/prefix length. If so, the neighbor :
 - 1) *unicasts* an EIGRP *Reply* message, telling the original router that it does indeed have a loop-free route
 - 2) does not forward the Query.
- If the neighbor itself is *Active* on this route, that neighbor :
 - 1) floods EIGRP Query messages to its neighbors,
 - 2) does not immediately send an EIGRP Reply back to the original router—instead waiting on replies to its own set of Query messages.
- When a router has received Reply messages from all neighbors to which it sent any Query messages, that router can then send a Reply message to any of its neighbors as necessary.
- When a router has received a Reply for all its Query messages, that router may safely use the best of the routes confirmed to be loop free.

3.3.6.2 Stuck-In-Active

Routes for which a router does not receive all Replies within the Active Timer are considered to be *Stuck-in-Active* (*SIA*) routes.

IOS has two major branches of logic when reacting to SIA routes :

- Earlier versions of IOS took a rather drastic action, bringing down the “uncooperative” neighbors that had yet to send back an EIGRP Reply for that route.
- Later IOS versions (beginning in the 12.2 mainline) make an attempt to avoid failing the neighborhood. At the halfway point through the Active timer—a seemingly long 90 seconds by default—a router sends an *SIA-Query* (Stuck-in-Active Query) EIGRP message to each neighbor that has yet to send back a Reply. The purpose of the message is to either get an *SIA-Reply* back, meaning that the neighbor really is still waiting for replies to its own queries, or to get nothing in reply. In the first case, because the neighbor is alive and still working, there is no need to kill the neighborhood. In the second case, the neighbor was not able to reply, so the action of failing the neighborhood is reasonable.

➤ To join in online CLN thread: [“Proposals to optimize the EIGRP "Stuck In Active" algorithm”](#)

3.3.6.3 Optimizing the Query Process: Stub Routers

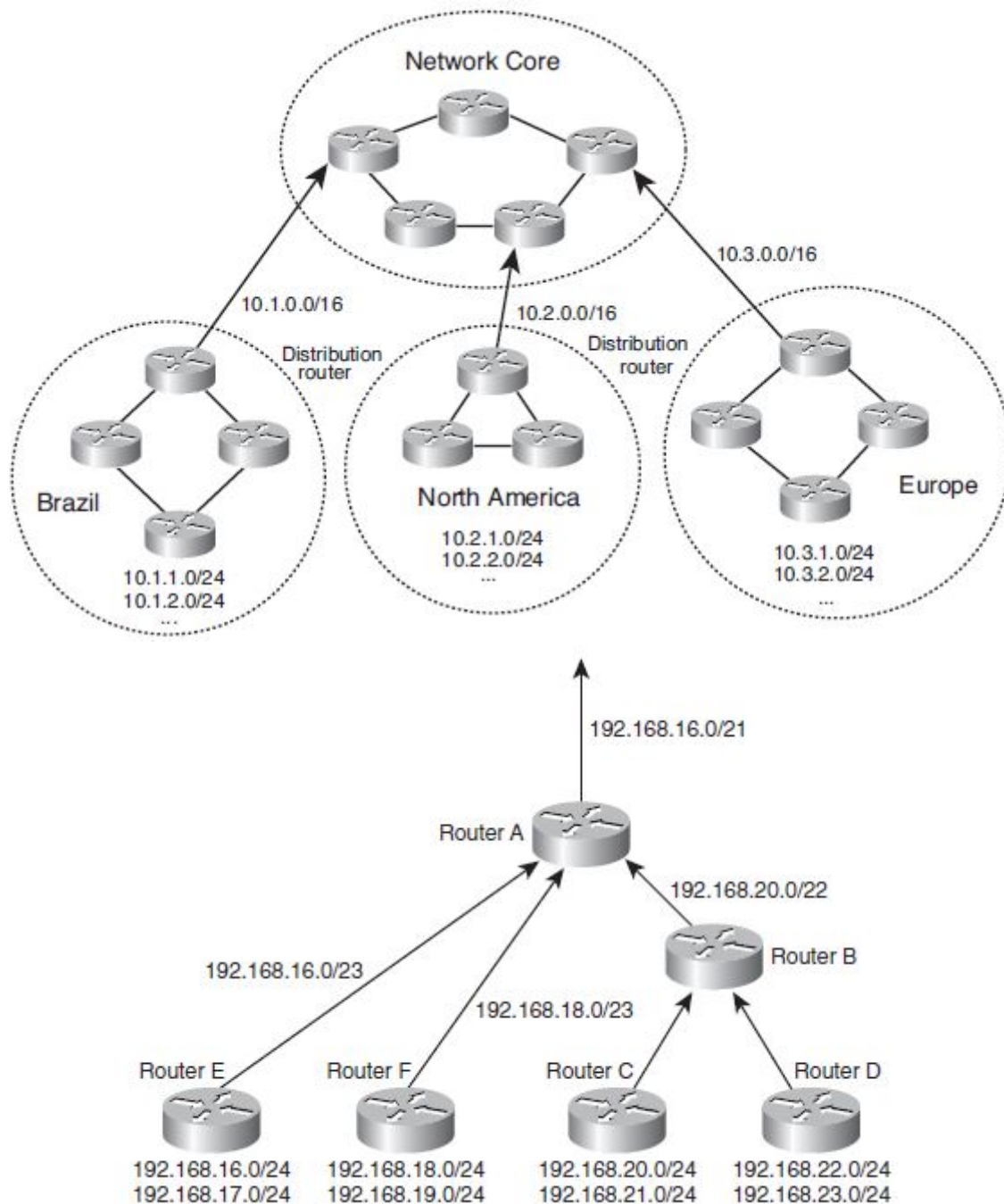
Stub router:: a router that does not advertise EIGRP-learned routes from one neighbor to other EIGRP neighbors.

Additionally, and possibly more significantly, non-stub routers note which EIGRP neighbors are stub routers, and the **non-stub routers do not send Query messages to the stub routers**. This action greatly reduces the scope of Query messages when a route goes Active, in addition to preventing the long, circuitous, and possibly harmful route.

3.4 Optimizing Router Resources

3.4.1 Limiting Reachability: Route Filtering

3.4.2 Without Limiting Reachability: Route Summarization



➤ All **Benefits** :

- **Better network stability**,
- Smaller routing **and topology** tables, while all destinations still reachable,
- **Smaller impact on bandwidth usage by EIGRP Updates**,
- **Lower convergence time** by optimizing the DUAL algorithm by reducing Query scope: EIGRP Query stops at a router that has a summary route that includes the subnet listed in the Query but not the specific route listed in the Query,
- EIGRP supports summarization at **any location in the internetwork**.
- The summary has **the metric of the best of the subnets being summarized**: cf. [Summary of all IPv4 and IPv6 IGPs](#)
[Summary Route Metric](#)

➤ All **drawbacks** :

- Can cause suboptimal routing.
- Packets destined for inaccessible destinations will flow to the summarizing router before being discarded.

The **recommended practices** regarding summarization include the following:

- Implement summarization at WAN connectivity and remote-access points *toward the network core*, to reduce the routing table size.
- Summarize at the distribution layer for all network at interfaces that point *to the network core*.

3.5 Avoiding External Routes Confusion: Router-ID

Each router determines its RID when the EIGRP process starts, using the same general rules as does OSPF for determining the OSPF RID, as follows:

- **Step 1:** Use the configured value.
- **Step 2:** Use the highest IPv4 address on an up/up loopback interface.
- **Step 3:** Use the highest IPv4 address on an up/up non-loopback interface.

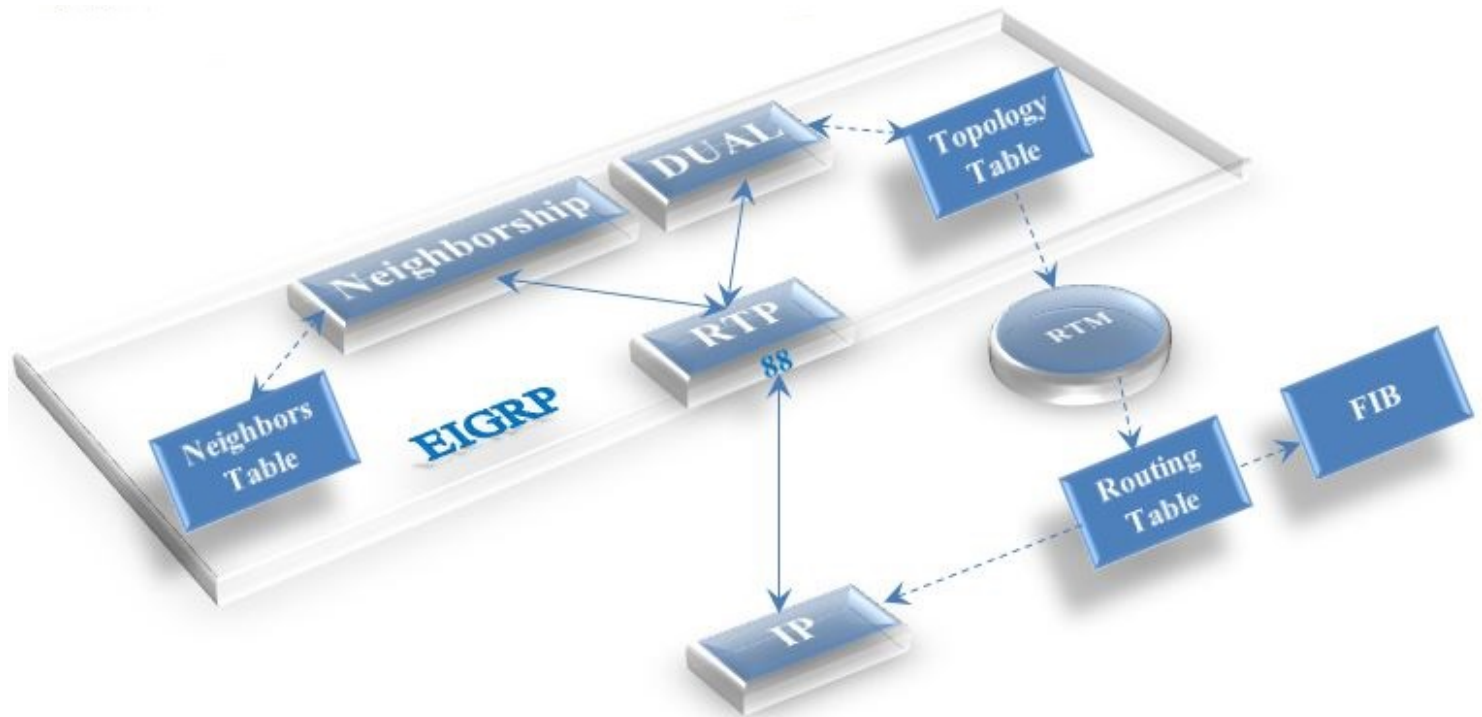
The only time the value of EIGRP RIDs matters is when injecting external routes into EIGRP. In that case, the routers injecting the external routes must have unique RIDs to avoid confusion.

3.6 Summary

3.6.1 Features

Feature	Description
Routing Protocol	Hybrid routing protocol: a distance-vector protocol that has link-state protocol characteristics.
Transport	RTP, IP protocol type 88 (does not use UDP or TCP). Reliable Transport Protocol Ensures the reliable and ordered delivery of route updates
Hello interval	Interval at which a router sends EIGRP Hello messages on an interface. Hellos are multicasted at 224.0.0.10.
Hold Timer	Timer used to determine when a neighboring router has failed, based on a router not receiving any EIGRP messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.10, with retransmissions being sent to each neighbor's unicast IP address. Can also be sent to the neighbor's unicast IP address. Hellos acknowledge updates with unicasts.
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used only when there are changes
Authentication	Supports MD5 authentication only, with a pre-shared key
VLSM/classless	EIGRP includes the mask with each route, also allowing it to support discontinuous networks and VLSM.
Route Tags	Allows EIGRP to tag routes as they are redistributed into EIGRP.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Diffusing Update Algorithm (DUAL)	Determines loop-free paths to destinations
Metric	Based on constrained bandwidth and cumulative delay by default, and optionally load and reliability.
Load-Balancing	By default, equal-cost load balancing. Unequal-cost load sharing possible.
Administrative Distance	90 for EIGRP internal routes, 170 for EIGRP external (E1/E2) routes, and 5 for EIGRP summary routes.
Topological Requirements	Does not require a hierarchical physical topology.
Manual route summarization	Allows route summarization at any point in the EIGRP network.
Automatic Summarization	EIGRP supports, and defaults to use, automatic route summarization at classful network boundaries.
Scalability	High: used in large networks
Diameter	100 by default; up to 255
Multiprotocol	Supports the advertisement of IPX and AppleTalk routes, and IP Version 6 (EIGRPv6)

3.6.2 Stack



3.6.3 Relations With Other Routing Protocols

Cf. [RTM](#)

3.6.4 “Opening” EIGRP

Out of CCNP ROUTE scope

Cisco has opened up its EIGRP routing protocol as an open standard in order to help companies operate in a multi-vendor environment. Any networking vendor can now freely implement EIGRP on their equipment, and interoperate with thousands of networks running EIGRP today.

Cisco has released an open basic EIGRP to the IETF as an [Informational RFC](#) (work in progress as a draft).

But as [Anthony Burke](#) pointed out:

- The “stub” router advanced feature of EIGRP has not been released to the IETF.
- Informational RFC allows Cisco to retain control of the EIGRP protocol.

Cisco is taking a conservative path; let's hope that they will progressively set EIGRP free ;) More information can be found in this [Packet Pushers podcast](#).

3.7 Configuration Commands

3.7.1 Dynamic Neighborship

1. (config)#**router eigrp** *asn* Neighboring routers' commands must be configured with the same ASN parameter to become neighbors.
2. (config-router)#**network** *net-id wildcard-mask* IOS enables only EIGRP on interfaces matched by this command. When enabled, the router does the following:
 - a. Attempts to discover EIGRP neighbors on that interface by sending multicast EIGRP Hello messages,
 - b. Advertises to other neighbors about the subnet connected to the interface.

3.7.2 Controlling Neighborships

3.7.2.1 and Reducing Multicasts: Static Neighborship

(config-router)#**neighbor** *ip-address outgoing-interface* Defines a neighbor (to be done on both routers). **Disables all EIGRP incoming and outgoing multicast messages on that interface, uses only unicast messages.** The EIGRP configuration still needs a **network** command that matches the interface referenced by this command

- **Cisco should drop the unnecessary outgoing-interface parameter.**
- If at least one EIGRP static neighbor is defined on an interface, no dynamic neighbors can be either discovered or continue to work if already discovered.

3.7.2.2 Preventing Unwanted Neighbors

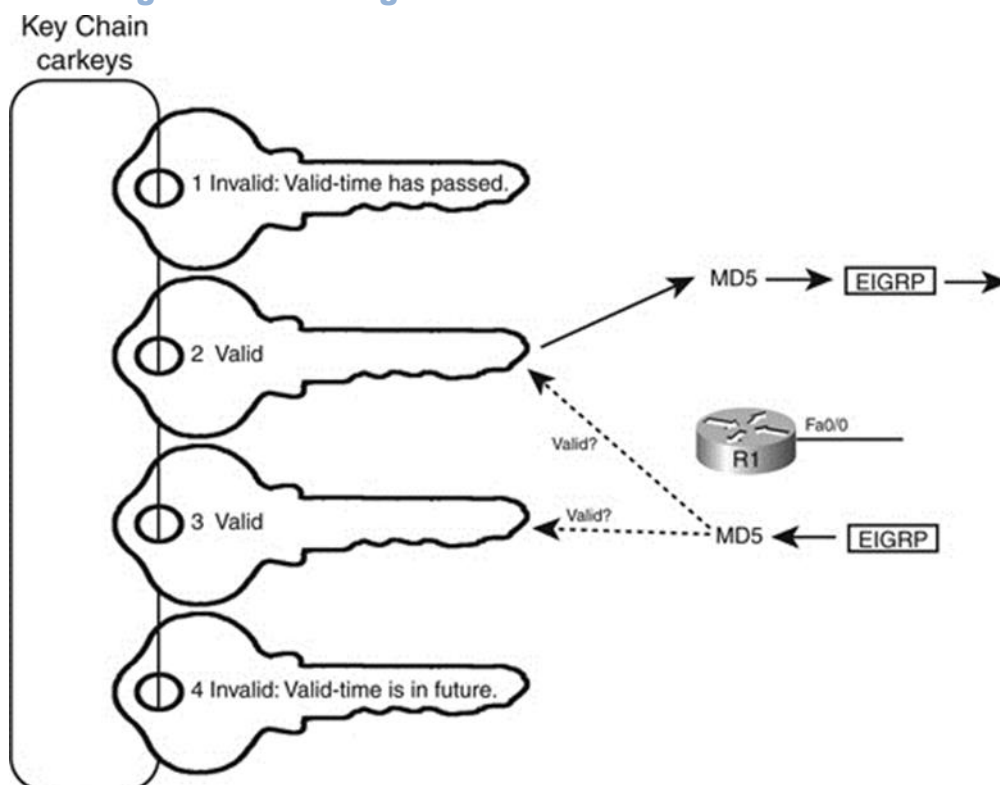
3.7.2.2.1 Passive Interfaces

(config-router)#**passive-interface** *type #* Sets that interface as passive : suppresses the exchange of hellos and updates on the interface. The subnet is still advertised on other enabled interfaces.

(config-router)#**passive-interface default** Sets all interfaces as passive by default

(config-router)#**no passive-interface** *type #* Sets that interface as not passive (needed if previous command entered)

3.7.2.2.2 Authenticating EIGRP Messages



Key Chain Time-Based Logic :

Neighboring EIGRP routers that use authentication should be configured to use **NTP to synchronize their time-of-day clocks**.

The existence of **multiple keys in a key chain, and the existence of valid lifetimes for each key**, can cause some confusion about when the keys are used. The rules can be summarized as follows:

- **Sending EIGRP messages:** Use the lowest key number among all currently valid keys.
- **Receiving EIGRP message:** Check the MD5 digest using ALL currently valid keys.

If the last key expires, authentication will continue and an error message will be generated. To disable authentication, you must manually delete the last valid key.

○ **Step 1:** Create an (authentication) key chain:

- (config)#**key chain** *name* Create the chain and give it a name. The name *does not have to match* on the neighboring routers.
- (config-keychain)#**key** *number* Create one or more key numbers. The key numbers *do have to match* on the neighboring routers.
- (config-keychain-key)#**key-string** *value* Define the authentication key's value. The key strings must match on the neighboring routers.
- (O) (config-keychain-key)#**accept-lifetime** *hh:mn:ss month day year { hh:mn:ss month day year | duration s | infinite }*
Define the lifetime (begin {end | seconds | infinite} time period) for accepting each key string of EIGRP messages.
- (O) (config-keychain-key)#**send-lifetime** *hh:mn:ss month day year { hh:mn:ss month day year | duration s | infinite }*
Define the lifetime (begin {end | seconds | infinite} time period) for transmitting each key string of EIGRP messages.

○ **Step 2:** Enable EIGRP MD5 authentication on an interface:

- (config-if)#**ip authentication mode eigrp** *asn md5* Enable EIGRP MD5 authentication on an interface, for a particular EIGRP ASN.

○ **Step 3:** Link a key chain to an interface:

- (config-if)#**ip authentication key-chain eigrp** *asn name-of-chain* Refer to the correct key chain to be used on an interface.

3.7.3 Optimizing Convergence delay

3.7.3.1 Tuning Hello/Hold Timers

- (config-if)#**ip hello-interval eigrp** *AS# value(s)* Sets the hello timer **for all the neighbors reachable on that interface** (default : 5s on Ethernet ; 60s on NBMA ("12 x Ethernet"))

- (config-if)#**ip hold-time eigrp** *AS# value(s)* Sets the hold timer **on (!!!) all the neighbors reachable on that interface** for their respective neighborhood with the current router (default : **3 x hello-interval** : 15s on Ethernet ; 180s on NBMA ("12 x Ethernet"))

3.7.3.2 Making Feasible Successors or Choosing One Route Instead of Another One: Metric Tuning → Cf. Load-Balancing Summary

- (config-router)#**metric weights** *0 k1 k2 k3 k4 k5* Sets 5 variables (k1 through k5, 0-255), each of which weights the metric calculation formula more or less heavily for various parts of the formula ; by default, 1 0 1 0 0. **Cisco recommends to not change the default values to avoid route flapping and CPU overload. Actually, it does not change anything.**

- (config-(sub)if)#**bandwidth** *value (kbits/s)* Sets the bandwidth of the (sub)interface

- (config-(sub)if)#**delay** *value (*10µs)* Sets the delay of the (sub)interface

(config-router)#**offset-list** {*access-list-number* | *access-list-name*} {*in* | *out*} *offset* [*interface-type interface-number*]

Examines all EIGRP Updates *received* or *sent* on [(*sub*)*interface*] or on all interfaces, and if the prefix matched by this *ACL* is found, add this *offset* to the computed FD and RD for that prefix (and possibly to the offset already present in the Update). This *offset* is propagated from router to router within the Updates.

- With well-tuned EIGRP Timers and with Feasible Successor routes (which can be made if necessary by manipulating the metrics), convergence time can be held low.

3.7.3.3 Optimizing the Query Process

3.7.3.3.1 Stub Routers

(config-router)#**eigrp stub** [*connected*] [*summary*] [*static*] [*redistributed*] [*receive-only*]

Do not advertise EIGRP-learned routes from one neighbor to other EIGRP neighbors

Parameters on the eigrp stub Command

Option	This router is allowed to...	
Connected	Advertise connected routes but only for interfaces matched with a network command.	Default
Summary	Advertise auto-summarized or statically configured summary routes.	Default
Static	Advertises static routes, assuming the redistribute static command is configured.	
Redistributed	Advertises redistributed routes, assuming redistribution is configured.	
Receive-only	Does not advertise any routes. This option cannot be used with any other option.	

3.7.3.3.2 Active Timer

(config-router)#**timers active-time** *time* (mn)

Configures the Active Timer for an entire EIGRP process.
Default: 3 mn

3.7.4 Optimizing Bandwidth Utilization

3.7.4.1 Load Sharing → Cf. [Load-Balancing Summary](#)

3.7.4.1.1 Equal Metric Paths: ECMP

(config-router)#**maximum-paths** *number*

Sets the maximum number of allowed routes in the routing table for each prefix/prefix length. The default setting of 4 is often big enough, because most internetworks do not have enough redundancy to have more than four possible routes.

3.7.4.1.2 Unequal Metric Paths: Variance - UCMP

(config-router)#**variance** *multiplier* (1-128)

Considers **only all FS** which meet the following criteria worth being added to the routing table, up to the number of routes defined by the **maximum-paths** command : **$FS's\ FD \leq (S's\ FD * multiplier)$**

3.7.4.2 Bandwidth Percentage Used for EIGRP Messages

(config-if/subif)#**ip bandwidth-percent eigrp** *asn percent*

Limits the rate of sending the EIGRP messages on that (sub)interface up to *percent* % of the bandwidth defined on the (sub)interface with the **bandwidth** command (50 % by default). To accomplish this, IOS queues the EIGRP messages in memory, delaying them briefly.

There are a few caveats to keep in mind when trying to limit the bandwidth consumed by EIGRP:

- The IOS default for bandwidth on serial interfaces and sub-interfaces is 1544 (Kbps).
- EIGRP limits the consumed bandwidth based on the percentage of interface/sub-interface bandwidth.

- This feature relies on the bandwidth of the interface or sub-interface *through which the neighbor is reachable*, so don't set only the physical interface bandwidth but also the sub-interfaces.
- *Recommendation*: Set the bandwidth of point-to-point links to the speed of the Committed Information Rate (CIR) of the single PVC on the sub-interface.
- *General recommendation*: Set the bandwidth of multipoint sub-interfaces to around the total CIR for all VCs assigned to the sub-interface.
- Note that for multipoint sub-interfaces, IOS WAN bandwidth control first *divides the sub-interface bandwidth by the number of configured PVCs* and then determines the EIGRP percentage based on that number.

3.7.5 Optimizing Router Resources

3.7.5.1 Limiting Reachability: Route Filtering

```
(config-router)#distribute-list [[access-list-number | name] | [prefix name] | [route-map name] in [interface-type #]
(config-router)#distribute-list [[access-list-number | name] | [prefix name] | [route-map name] out [interface-type #
| source-routing-process {as-number| process-id}]
```

- Addresses *not specified* in the **distribute-list in** command *will not be inserted* into the topology table.
- Addresses *not specified* in the **distribute-list out** command *will not be advertised* in outgoing routing updates.

Regarding distribute-lists used with an ACL:

- *In all cases, IOS compares the standard ACL source-ip-address field to the prefix of each route.*
- *Within a single IGP, IOS compares the extended ACL source-ip-address field to the Update-Source (in) / Update-Destination (out) and dest-ip-address field to the prefix of each route.*
- *In all other cases – between different IGPs during redistribution and within BGP -, with an extended ACL, IOS compares the source-ip-address parameter to the prefix and the dest-ip-address to the prefix length of each route.*

3.7.5.2 Without Limiting Reachability

3.7.5.2.1 Route Summarization

3.7.5.2.1.1 Manual Summarization

```
(config-if)#ip summary-address eigrp asn prefix subnet-mask [admin-distance]
```

- *asn* : AS in which the announcement is made (related to config-if),
- By default, AD = 5

- The router brings down, and then back up, all EIGRP neighbors reachable *on that interface only*, effectively causing neighbors to forget previous topology information, and listen to new information (when the neighborships recover).
- When the neighborships recover, the router advertises the summary route, assuming the router has at least one route whose address range is inside the range of the summary route.
- The router does *not* advertise the subordinate routes.
- IOS sets the summary route's metric components based on the lowest metric route upon which the summary route is based.
- When EIGRP advertises a summary route, the advertising router considers the summary route to be up and working unless all subordinate routes fail.
- The router *adds a route to its own routing table*, for the summary prefix/prefix-length, with an outgoing interface of *null0*. Note that this summary route with outgoing interface *null0* is often called a *discard route*.
- *To ensure that the router adds this local summary route, the router uses the local administrative distance (AD) setting of 5. With an AD of 5, the advertising router will ignore any identical EIGRP-advertised summary routes because EIGRP default AD for internal routes is 90.*

3.7.5.2.1.2 Auto-summary

(config-router)#**auto-summary** If a router has some working interfaces with IP addresses in *different classful networks*, it automatically advertises a summary route for each classful network on interfaces attached to a different classful network. *Discontiguous networks are not supported.*

3.7.5.2.2 Default Routes

3.7.5.2.2.1 Static Default Route

- **Step 1:** Choose *one* of these options:
 - a) (config)#**ip route** 0.0.0.0 0.0.0.0 interface # [administrative-distance] → *static directly connected route*
 - b) (config)#**ip route** 0.0.0.0 0.0.0.0 next-hop-ip-address [administrative-distance] [interface #] → *static via a.b.c.d route*

The route is in the routing table, but not in the topology table yet, so EIGRP does not yet advertise the route.

- **Step 2:** Inject this route into the EIGRP topology database which will be advertised:
 - a) either using the **network** 0.0.0.0 0.0.0.0 command, *with a static directly connected route defined (Does not work for a BGP-injected default route!)*
 - b) or *by redistributing the static directly connected route or the static via a.b.c.d route.*
- *The first option using the default static directly connected route is not recommended at all on multipoint interfaces:*
 - ⇒ *This command causes the router to consider all the destinations that the router cannot reach through an alternate route, as directly connected to interface #. Hence, the router sends an ARP request for each host for which it needs to send packets onto this network segment. This configuration can cause high processor utilization and a large ARP cache (along with memory allocation failures).*
 - ⇒ *Configuring a default route or other static route that directs the router to forward packets for a large range of destinations to a connected broadcast network segment can cause the router to reload.*

3.7.5.2.2.2 Static Default Classful Network

- **Step 1:** On the router to which all traffic should be directed, identify a classful network in the routing table that can be advertised into the EIGRP domain, *flag this route as a candidate default route* (in Routing Table) and *set the Exterior Flag* (in Topology Table) using the global command :

(config)#**ip default-network** classful-network-number.

Most often, that route is either created off a *loopback interface* for the purpose of making this process work, or an existing route on the Internet side of the router is used.

- **Step 2:** Inject this route into the EIGRP topology database which will be advertised:
 - a) either using the **network** command,
 - b) or *by redistributing the connected route.*

3.7.6 Avoiding External Routes Confusion : Router-ID

(config-router)#**eigrp router-id** a.b.c.d Sets the RID

3.7.7 Summary of Issues related to EIGRP over Multipoint Frame Relay with a 'Hub and Spokes' Topology

Issues	Solutions
No Neighborhood Between Spokes	⇒ Full mesh (might not be applicable)
Dynamic Neighborhood Issue	⇒ broadcast keyword needed on frame-relay map ip command on multipoint sub-interfaces only
Static Neighborhood Issue	⇒ Static neighbor definition
No Topology Exchange Between Spokes	⇒ no ip split-horizon eigrp asn on the multipoint (sub-)interface(s) on the 'Hub'
Unreachability Issue Between Spokes (for interfaces in common subnet)	⇒ Static IP address ⇔ DLCI mapping on each spoke for all other spokes ⇒ or change of interface type (⇒ PTP)

3.8 Verification Commands

#show protocols

Lists for all interfaces:

- Line and protocol status
- IP address/len

#show ip protocols

Lists **network** parameters and more for each routing process :

- AS number Net-id Wildcard-mask
- Passive interfaces
- K-values
- Variance
- Redistribution
- Neighbor IP addresses - Elapsed time since neighborship was formed
- Administrative distance

#show ip route [prefix mask] [longer-prefixes] [summary]

Lists the contents of the routing table, with a D code for EIGRP-learned routes :

- Connected/Static/RIP/D/O/* (candidate for Default Route)
- Gateway of Last Resort
- Subnet number/length ***OR*** All the subordinate routes of that Subnet (with **longer-prefixes**)
- [AD/metric]
- Next-hop router
- hh:mm:ss since last entry update
- Out interface
- **summary**: number of subnets per AS, memory used, ...

#show ip eigrp neighbors [detail]

Lists :

- Known valid static and dynamic neighbors (distinction only with **detail**)
- Out Interface
- *Current* Hold timer
- Elapsed time since neighborship was formed
- Only with **detail**:
 - IOS version
 - Stub routers with the type of routes advertised from them (Connected/Summary/Static/Redistributed/Receive-only)

#show ip eigrp interfaces [detail]

Lists :

- The working *non passive* interfaces on which EIGRP is enabled
- Number of peers
- only with **detail**:
 - Hello timer *and (IOS 15.2) Hold timer*
 - *(IOS 15.2) Split Horizon enabled or not*
 - Authentication information

#show ip eigrp topology [*prefix mask* | *prefix/len*] [**all-links**] | **include** ...

For each route, one Successor and possibly multiple Feasible Successors :

- State (Passive – Active – Update – Query – Reply)
- Prefix / len
- IP address of the neighbor that advertised that route, “via Connected” for Connected routes or “via Redistributed” for Redistributed routes on the Redistributing router - FD / RD - Outgoing interface
- Only with [*prefix mask* | *prefix/len*]:
 - *Vector Metric* components : min bandwidth (in kbits/s), total delay (in microseconds/picoseconds), min reliability and max load
 - Non metric components : min MTU and hop count
 - *External Data* (Redistribution) : IP address of originating (redistributing) router, AS number, external protocol, external metric
- Exterior Flag : Candidate for Default Route
- **all-links**: Lists all possible routes, including those that are neither successor nor feasible successor routes.

Router Default Bandwidth and Delay Values for Common Interfaces

Media	Bandwidth	Delay
Ethernet	10,000 kbits/s	1000 μs
Fast Ethernet	100,000 kbits/s	100 μs
Gigabit Ethernet	1,000,000 kbits/s	10 μs
T1	1544 kbits/s	20,000 μs

#show ip eigrp traffic

Lists:

- Number of sent and received EIGRP messages / type
- AS number

#show ip eigrp accounting

Lists prefix accounting information for EIGRP processes, including :

- *AS number*
- *RID and current neighborhood state (Adjacency – Pending – Down)*

#show key chain

Lists the key chain configuration and also lists which keys are valid right now

#debug ip packet *acl-number*

*Shows contents of IP packets – output can be filtered out with an ACL.
Needs ‘no ip cef’ to display ICMP traffic.*

#debug eigrp packets [*packet-type*]

Shows contents of EIGRP packets (**HELLO, UPDATE, REQUEST, QUERY, REPLY, PROBE, ACK, STUB, SIAQUERY, SIAREPLY**), including all the reasons why the neighborhood failed, like authentication.

#debug eigrp fsm

Shows changes in Successor and Feasible Successor routes (**fsm**: Feasible Successor Metrics)

#debug ip eigrp asn

Shows IP EIGRP route events – shows routes installation into the routing table

#debug ip eigrp notifications

Displays EIGRP events and notifications

4 OSPFv2

It is described in [RFC 2328](#).

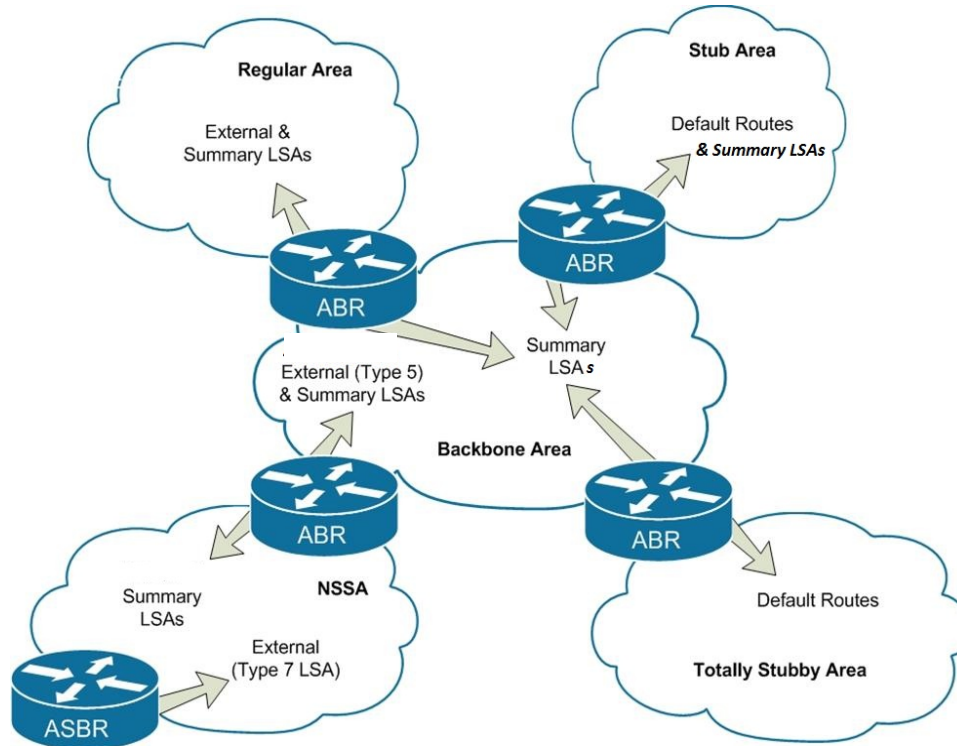
4.1 Hierarchically Segmenting the Topology

4.1.1 LSA Types

LSA Type	Common Name	Description
1	Router	Each router creates its own Type 1 LSA to represent itself for each area to which it connects. The LSDB for one area contains one Type 1 LSA per router per area, listing the RID and all interface IP addresses on that router that are in that area. Represents stub networks as well.
2	Network	One per transit network. Created by the DR on the subnet, and represents the subnet and the router interfaces connected to the subnet.
3	Net Summary	Created by ABRs to represent subnets listed in one area's type 1 and 2 LSAs when being advertised into another area. Defines the links (subnets) in the origin area, and cost, but no topology data.
4	ASBR Summary	Like a type 3 LSA, except it advertises a host route used to reach an ASBR. <i>Not created by (T)NSSA ABRs</i>
5	AS External	Created by ASBRs for external routes injected into OSPF.
6	Group Membership	Defined for MOSPF; not supported by Cisco IOS.
7	NSSA External	Created by ASBRs inside an NSSA area, instead of a type 5 LSA.
8	External Attributes	Not implemented in Cisco routers.
9–11	Opaque	Used as generic LSAs to allow for easy future extension of OSPF; for example, type 10 has been adapted for MPLS traffic engineering.

LSA Type (Number)	LSA Type (Name)	This Type Represents	Display Using show ip ospf database keyword...	LSID Is Equal To	Created By
1	Router	A router	router	RID of router	Each router creates its own
2	Network	A subnet in which a DR exists	network	DR's IP address in the subnet	The DR in that subnet
3	Summary	Subnet in another area	summary	Subnet number	An ABR

4.1.2 Area Types



From <http://www.ccnpguide.com/ccnp-route-642-902-ospf/> by Aaron

Stubby Areas

Even within the realm of stubby areas, four types of stubby areas exist:

- **Stubby,**
- **Totally Stubby,**
- **Not-So-Stubby Areas (NSSA),**
- **Totally NSSA.**

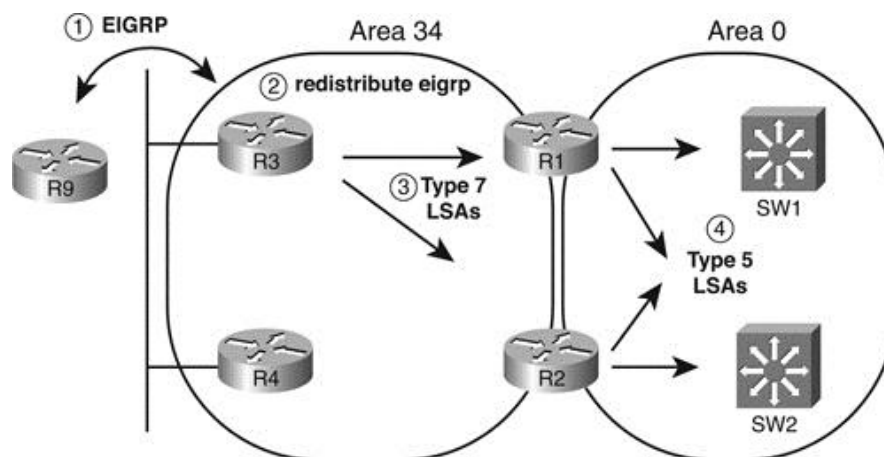
Two types of Stubby areas have the word “Totally” as part of the name, and two do not. The rules are:

- For all types of Stubby areas:
 - the ABR always *filters* Type 4 (ASBR Summary) and Type 5 (External) LSAs,
 - a *default route* (listing subnet 0.0.0.0 and mask 0.0.0.0) is advertised into the areas as:
 - a Type 3 LSA for Stubby and Totally Stubby areas (*automatically*)
 - a Type 7 LSA for NSSA and Totally NSSA (*only with default-information-originate* as keyword in `area area-id nssa` command),
- For Stubby and NSSA areas, the ABRs do not filter Type 3 LSAs, advertising Type 3 LSAs as normal,
- For Totally Stubby and Totally NSSA areas, the ABR also *filters* Type 3 LSAs, except for the one used for the default route, which reduces the size of the LSDB. These areas are *Cisco proprietary*.
- NSSA and Totally NSSA areas can *redistribute external routes* (Type 7 LSAs) into the area,
- Stubby and Totally Stubby areas cannot.

OSPF Stubby Area Types

Area Type	ABRs flood Type 5 External LSAs into the area?	ABRs flood Type 3 Summary LSAs into the area?	Allows redistribution of external LSAs into the stubby area?
Stub	No	Yes	No
Totally stubby	No	No	No
NSSA	No	Yes	Yes
Totally NSSA	No	No	Yes

External Routes into an NSSA Area (34): one direction only



- An ABR connected to a NSSA can be configured to NOT translate Type 7 LSAs into Type 5 LSAs.
- Old recommendations by Cisco (*I would suggest that it all depends on the CPU power and memory resources of all the routers involved in the areas*):
 - maximum of 2 Areas / ABR + Backbone
 - maximum of 50 Routers / Area

4.2 Establishing Neighborhood

4.2.1 The Interface Network Types

Default for LAN

Default for Point-To-Point
Designed for Partial Mesh

Default for Multipoint
Designed for Full Mesh

Designed for Partial Mesh

Designed for Partial Mesh

Interface Type	Uses DR/BDR?	Default Hello Interval	Dynamic Discovery of Neighbors?	More Than Two Routers Allowed in the Subnet?
Broadcast	Yes	10	Yes	Yes
Point-to-point ¹	No	10	Yes	No
Loopback	No	—	—	No
Nonbroadcast ² (NBMA)	Yes	30	No	Yes
Point-to-multi-point	No	30	Yes	Yes
Point-to-multi-point nonbroadcast	No	30	No	Yes

¹Default on Frame Relay point-to-point subinterfaces.

²Default on Frame Relay physical and multipoint subinterfaces.

- OSPF treats Point-To-Multipoint interfaces as a collection of Point-To-Point interfaces.
On the spokes, all the routes point towards the hub(s).
- OSPF treats a collection of Broadcast interfaces as a transit network, with a DR which must be negotiated on the hub.
On the spokes, the routes to networks off other spokes point directly to these spokes.

4.2.2 Neighborhood Requirements

Both the subnet number
and mask must match

Requirement	OSPF	EIGRP
* Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
* Must not be passive on the connected interface.	Yes	Yes
* Must be in same area.	Yes	N/A
* Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	Yes	No
* Router IDs must be unique.	Yes	No
* IP MTU must match.	Yes ¹	No
* Must pass neighbor authentication (if configured).	Yes	Yes
K-values (used in metric calculation) must match.	N/A	Yes
Must use the same ASN (EIGRP) or process-ID (OSPF) on the router configuration command.	No	Yes

¹May allow the other router to be listed in the `show ip ospf neighbor` command, but the MTU mismatch will prevent proper operation of the topology exchange.

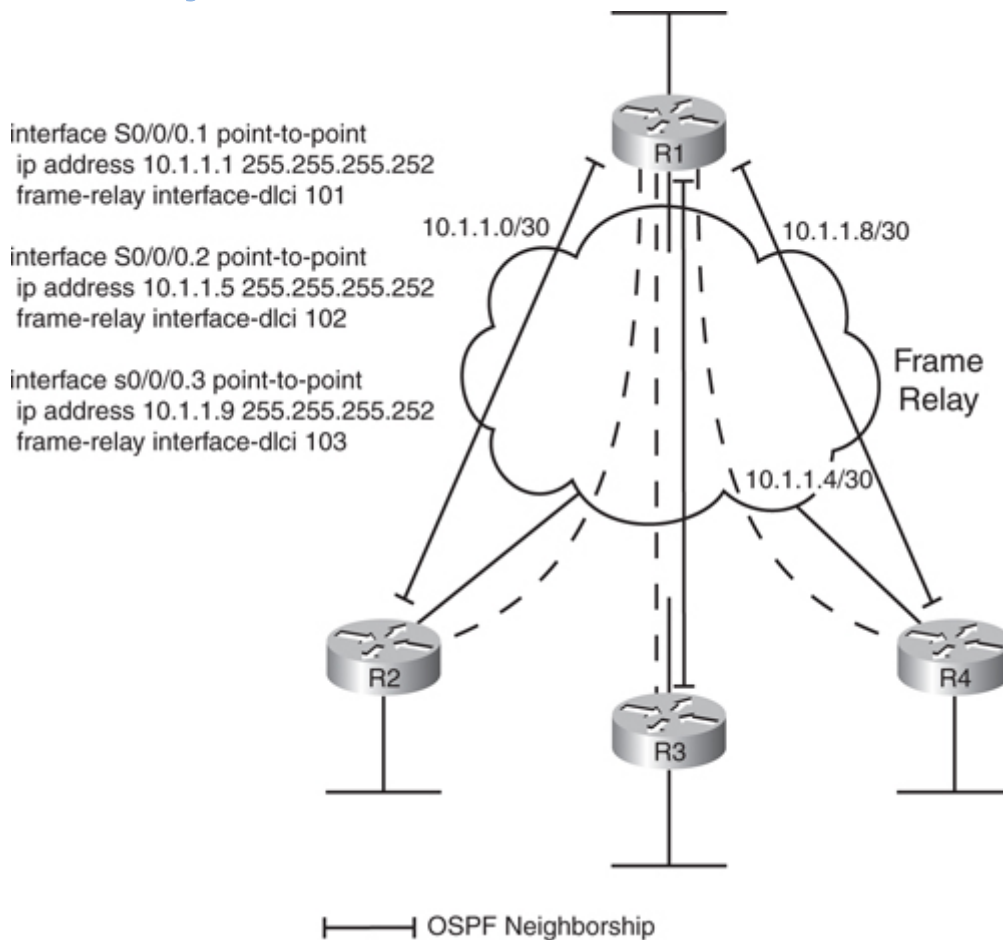
DOWN ⇒ INIT ⇒ EXSTART
🔍 ⚡ ⚡ ⚡ ⚡ ⚡ ⚡ ⚡

* Area Types <u>must</u> be the same ➤ <i>Normal, Stub, NSSA</i>	Yes	N/A
Network Interface Types <u>should</u> be the same ➤ <i>Broadcast, Non Broadcast, Point-To-Point, Point-To-Multipoint, Point-To-Multipoint Non Broadcast</i>	Yes	N/A
* Static Neighborhood for "Non Broadcast" and "Point-To-Multipoint Non Broadcast" Network Interface Types	Yes	N/A
* Broadcast keyword in frame-relay map ip when used instead of dynamic mapping (InARP)	Yes	N/A

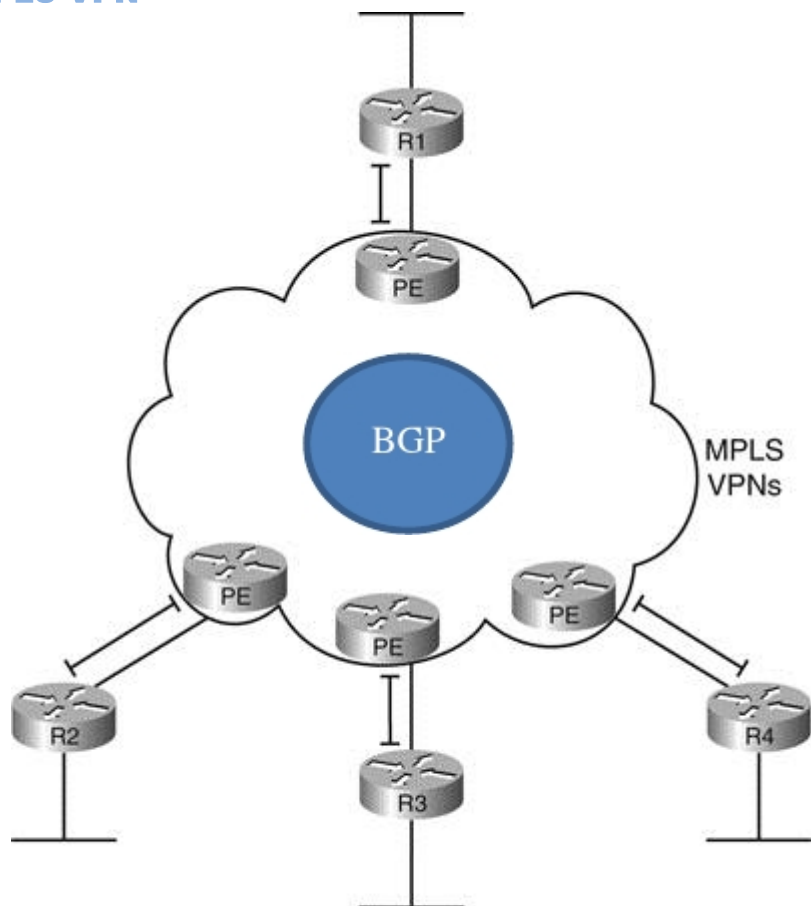
1.1.1

4.2.3 On WAN

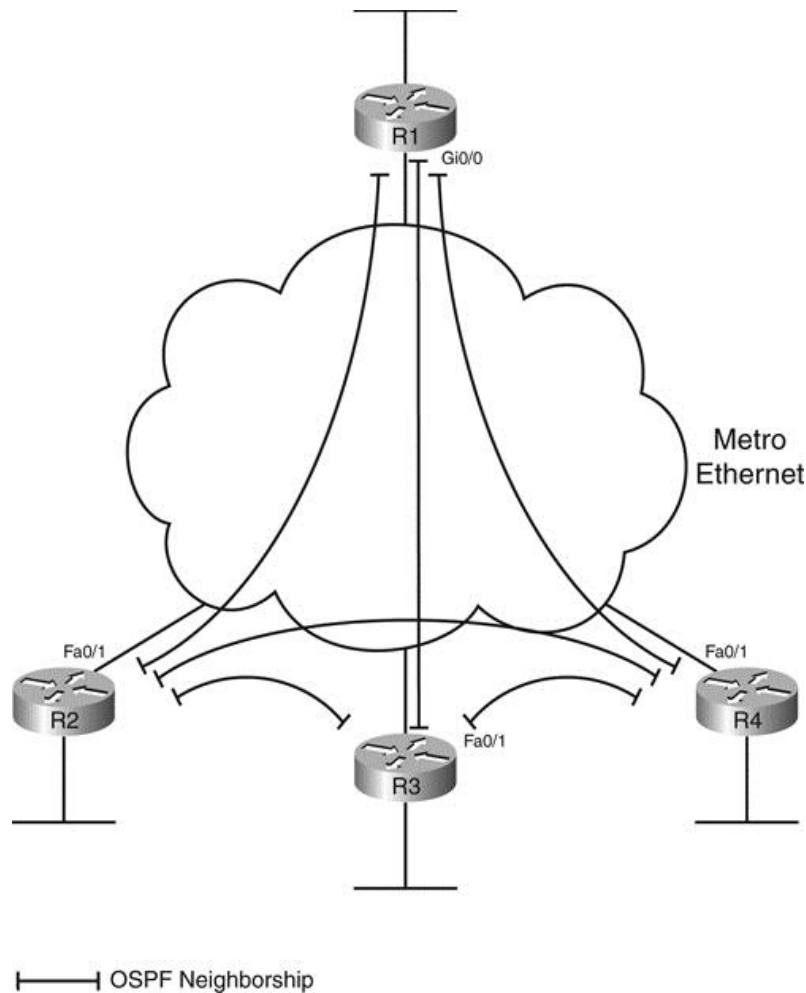
4.2.3.1 Over Frame Relay



4.2.3.2 Over L3 MPLS VPN



4.2.3.3 Over Metro Ethernet



4.3 Exchanging Link State Database

4.3.1 Seeding the LSDB

Before a router can send OSPF link state data to a neighbor, that router needs to add link state information for some interfaces in its LSDB to get the process started, from *two sources* (static configuration):

- The default route and prefixes of connected subnets for interfaces on which OSPF has been enabled on that router using the **network area** and/or **ip ospf area** commands,
- Prefixes learned by the **redistribution** of routes into OSPF from other routing protocols or routing information sources.

4.3.2 Message Types and Functions

Message Name/number	Description
Hello (1)	Used to discover neighbors, supply information used to confirm two routers should be allowed to become neighbors, to bring a neighbor relationship to a 2-way state, and to monitor a neighbor's responsiveness in case it fails
Database Description (DD or DBD) (2)	Used to exchange brief versions of each LSA, typically on initial topology exchange, so that a router knows a list of that neighbor's known LSAs
Link-State Request (LSR) (3)	A packet that lists the LSIDs of LSAs the sender of the LSR would like the receiver of the LSR to supply during database exchange
Link-State Update (LSU) (4)	A packet that contains fully detailed LSAs, typically sent in response to an LSR message
Link-State Acknowledgment (LSAck) (5)	Sent to confirm receipt of an LSU message

4.3.3 Neighbor

<https://t.me/learningnets>

State Reference

*DR/BDR Election
may begin*

*Master/Slave
choice*

*Which unknown and fresher
LSAs does the neighbor have?*

*Needed LSAs
Exchange*

State	Meaning
Down	No Hellos have been received from this neighbor for more than the dead interval.
Attempt	Used when the neighbor is defined with the neighbor command, after sending a Hello, but before receiving a Hello from that neighbor.
Init	A Hello has been received from the neighbor, but it did not have the local router's RID in it or lists parameters that do not pass the neighbor verification checks. This is a permanent state when Hello parameters do not match.
2Way	A Hello has been received from the neighbor, it has the router's RID in it, and all neighbor verification checks passed.
ExStart	Currently negotiating the DD sequence numbers and master/slave logic used for DD packets.
Exchange	Finished negotiating the DD process particulars, and currently exchanging DD packets.
Loading	All DD packets are exchanged, and the routers are currently sending LSR, LSU, and LSAck packets to exchange full LSAs.
Full	Neighbors are fully adjacent, meaning they believe that their LSDBs for that area are identical. Routing table (re)calculations can begin.

4.3.4 DR/BDR Election

Routers **which have attained at least the 2Way state** use the following *election rules* when neither a DR nor BDR yet exists:

- Choose the router with the **highest interface priority** (default 1, max 255),
- If **tied** on priority, choose the router with **highest RID**,
- A router with a router priority set to **0** is **ineligible** to become the DR or BDR,
- Choose a BDR, based on next-best priority, or if a tie, next-best (highest) RID,
- The router which is elected as DR or BDR has its interface state set to **DR or BDR**,
- Other routers have their interface state set to **DROther**.

OSPF uses a DR in a particular subnet for two main purposes:

- To create and flood a Type 2 network LSA for that subnet
- To aid in the detailed process of database exchange over that subnet

4.3.5 LSDB Exchange Process

4.3.5.1 Exchange with a DR/BDR

Database exchange with a DR differs slightly than database exchange when no DR exists. The majority of the process is similar, with the same messages, meanings, and neighbor states. The big difference is the overriding choice of with whom each router chooses to perform database exchange.

Non-DR routers (*DROther*) do not exchange their databases directly with all neighbors on a subnet. Instead, they exchange their database only with the DR/BDR and not any other DROther routers in the subnet. Then, the DR exchanges any new/changed LSAs with the rest of the OSPF routers in the subnet.

The messages differ slightly as well, as follows:

- The non-DR performs database exchange with the same messages, but sends these messages to the **224.0.0.6 All-DR-routers multicast** address. The messages sent to 224.0.0.6 are processed by the DR/BDR only. The DR actively participates, replying to the messages, with the BDR acting as a silent bystander. In effect, this allows the non-DR router to exchange their database directly with the DR and BDR, but with none of the other routers in the subnet.
- The DR performs database exchange with the same messages but sends the messages to the **224.0.0.5 all-SPF-routers multicast** address. All OSPF routers process these messages, so the rest of the routers also learn the newly exchanged LSAs.

4.3.5.2 Classes of Neighborship

With OSPF, even after all the neighbor parameter checks pass, 2/3 classes of neighborships exist:

- **neighbors** ('2WAY') : routers with which topology data was not exchanged directly ; DROthers between them
- **adjacent neighbors** ('EXCHANGE' or 'LOADING') : routers in a state greater or equal to 'Exchange', i.e. all neighbors except DROthers between them.
- **fully adjacent neighbors** ('FULL') : routers with which OSPF topology has been fully exchanged, i.e. all neighbors except DROthers between them.

4.3.5.3 Area Flooding

LSAs need to be flooded throughout an area. To do so, when a router learns new LSAs from one neighbor, that router then knows that its other neighbors in that same area may not know of that LSA. Similarly, when an LSA changes, for example, when an interface changes state, a router may learn the same old LSA but with a new sequence number, and again need to flood the changed LSA to other neighbors in that area.

4.3.5.4 Periodic Flooding

Although OSPF does not send routing updates on a periodic interval, as do distance vector protocols, OSPF does reflood each LSA every 30 minutes based on each LSA's age variable. The router that creates the LSA sets this age to 0 (seconds). Each router then increments the age of its copy of each LSA over time. If 30 minutes pass with no changes to an LSA—meaning no other reason existed in that 30 minutes to cause a reflooding of the LSA—the owning router increments the sequence number, resets the timer to 0, and refloods the LSA.

4.4 Computing Routes

SPF (Shortest Path First) - **Dijkstra's algorithm** - must execute after the LSDB exchange and when the intra-area topology changes, because changes in topology impact the choice of best route. However, *changes to Type 3 LSAs do not drive a recalculation of the SPF algorithm*, because the Type 3 LSAs do not actually describe the topology.

4.4.1 Calculating the Cost of Intra-Area Routes

For **intra-area routes**, OSPF uses pure **link state logic**. When a router analyzes the LSDB to calculate the best route to each subnet, it does the following:

- **Step 1:** Finds all subnets inside the area, based on the stub interfaces listed in the Type 1 LSAs and based on any Type 2 network LSAs
- **Step 2:** Runs SPF to find all possible loop-free paths through the area's topology, from itself to each subnet
- **Step 3:** Calculates the OSPF interface costs for all outgoing interfaces in each route, picking the lowest total cost route for each subnet as the best route

4.4.2 Calculating the Cost of Inter-area Routes

For **inter-area routes**, OSPF uses **distance vector logic**. For routers in one area to calculate the cost of an inter-area route, the process is simple when you realize that the Type 3 LSA lists the ABR's best cost to reach that inter-area subnet. To calculate the cost for each possible inter-area route:

- **Step 1:** Calculate the intra-area cost from that router to the ABR listed in the type 3 LSA.
- **Step 2:** Add the cost value listed in the Type 3 LSA. (This cost represents the cost from the ABR to the destination subnet.)

4.4.3 Calculating the Cost of External Routes

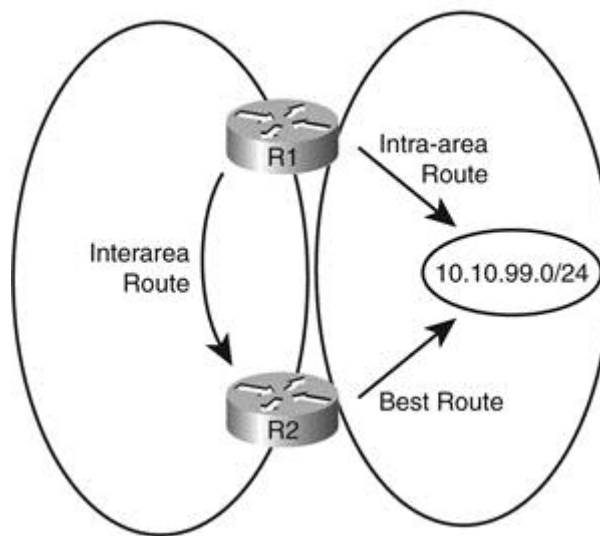
Cf. [Determining the Next-Hop for External Routes](#).

4.4.4 Special Rules Concerning Intra-area and Inter-area Routes on ABRs

OSPF has a couple of rules concerning intra-area and inter-area routes that take precedence over the simple comparison of the cost calculated for the various routes. The issue exists when more than one ABR connects to the same two areas. Many designs use two routers between the backbone and each non-backbone area for redundancy, so this design occurs in many OSPF networks.

The issue relates to the fact that with two or more ABRs, the ABRs themselves, when calculating their own routing tables, can calculate both an intra-area route and inter-area route for subnets in the backbone area.

Intra-Area or Inter-area Route to 10.10.99.0/24



- **Step 1:** When choosing the best route, an intra-area route is always better than a competing inter-area route, regardless of metric.
- **Step 2:** If an ABR learns a Type 3 LSA inside a non-backbone area, the ABR ignores that LSA when calculating its own routes.

4.4.5 Cost Components

$$\text{Cost} = \frac{\text{Reference-bandwidth}}{\text{interface-bandwidth}}$$

with **Reference-bandwidth** = 10^8 and **interface-bandwidth** in bits/s by default

Note : OSPF always rounds down when the calculation results in a fraction.

4.4.6 Backup Routes: LFA Feasible Successors

Out of the scope of CCNP ROUTE: From [Ivan Pepelnjak's Blog](#) :

Every single router knows the whole topology of all locally attached areas and can thus easily calculate which of its neighbors could be feasible successors. That's exactly what LFA (Loop Free Alternate) is doing:

- *OSPF routing process runs SPF, computes its own best paths, and installs them in the IP routing table (RIB).*
- *After the network has converged, OSPF runs SPF algorithm from the perspective of its neighbors. If a neighbor's SPF tree doesn't use current router as the next hop for a specific destination, it's safe to use that neighbor as a feasible successor.*
- *The feasible successor information calculated by OSPF is downloaded in RIB and FIB, where it can be used immediately after the link failure.*

4.5 Optimizing Router Resources

4.5.1 Limiting Reachability: Route Filtering

Route filtering can be used:

- to purposefully prevent hosts in one part of an internetwork from sending packets to another part,
 - to reduce the size of the topology table and IP routing table, reducing both OSPF memory and CPU consumption,
 - to make the packet forwarding process run slightly better.
- OSPF cannot and *does not allow the filtering of LSAs inside an area*, specifically the Type 1 and Type 2 LSAs that describe the intra-area topology.

IOS limits OSPF route filtering to the following:

- Filtering Type 3 LSAs on ABRs,
 - Filtering Type 5/7 LSAs on ASBRs (during redistribution),
 - Filtering the routes OSPF would normally add to the IP routing table on a single router.
- *The only way to filter Type 5/7 LSAs on an ABR is to use a dangerous trick by filtering the Type 3 LSA describing the ASBR connected to a (T)NSSA area: since the ASBR is not accessible from the ABR (and beyond), no external prefixes advertised by this ASBR will be installed into the routing table of this ABR and routers beyond.*

4.5.2 Without Limiting Reachability

4.5.2.1 Route Summarization

Route summarization can:

- optimize resource utilization: by reducing the size of the topology and IP routing tables, reducing both OSPF memory and CPU consumption, and by reducing packet forwarding overhead,
- optimize bandwidth utilization by reducing routing protocol updates,
- optimize network stability: no update is generated when some subordinate subnets state/metric change (except for the least or most costly subnet, but this can be overturned by configuring a cost for the summary route),

But summarization may have a negative effect of creating less-efficient paths through the internetwork.

OSPF allows summarization at both ABRs and ASBRs but not on other OSPF routers.

4.5.2.2 Default Routes and Stub Areas

The default route is announced as IP prefix 0.0.0.0/0 in OSPF.

Enterprises typically use default routes in two different cases:

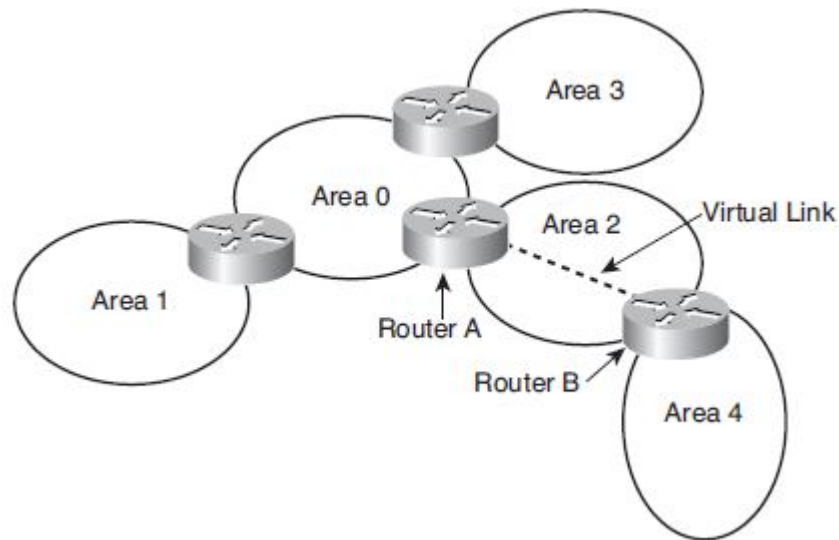
- To direct remote-site routers at the edge of the Enterprise network to send all packets toward the core of the Enterprise, with the core routers knowing all the more specific routes to Enterprise destination addresses.
- To direct traffic on all Enterprise routers toward an Internet-facing router so that all traffic destined for the Internet eventually arrives at the Enterprise Internet-connected routers.

4.6 Extending the Backbone Area with Virtual Links

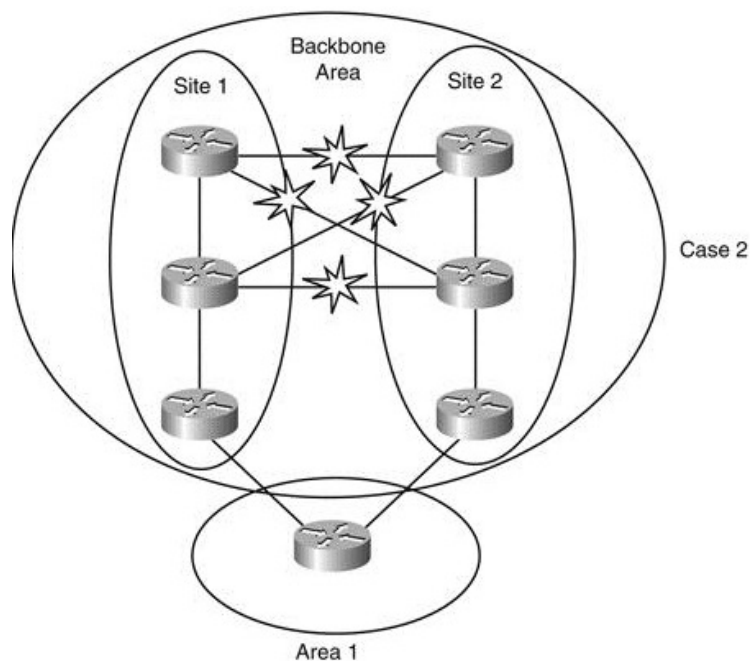
- **Case 1.** An existing internetwork needs to add a new area, with a convenient, low-cost connection point with another non-backbone area. However, that connection does not give the new area any connection to area 0.
- **Case 2.** Even with a well-designed area 0, a combination of link failures might result in a discontinuous backbone area, essentially creating two backbone areas.
- **Case 3.** Two companies could merge, each using OSPF. To merge the OSPF domains, one backbone area must exist. It may be more convenient to connect the two networks using links through an existing non-backbone area, but that design means two backbone areas, which is not allowed.

First Case Solution: Connecting Area 4 to Backbone Area with a Virtual Link (VL)

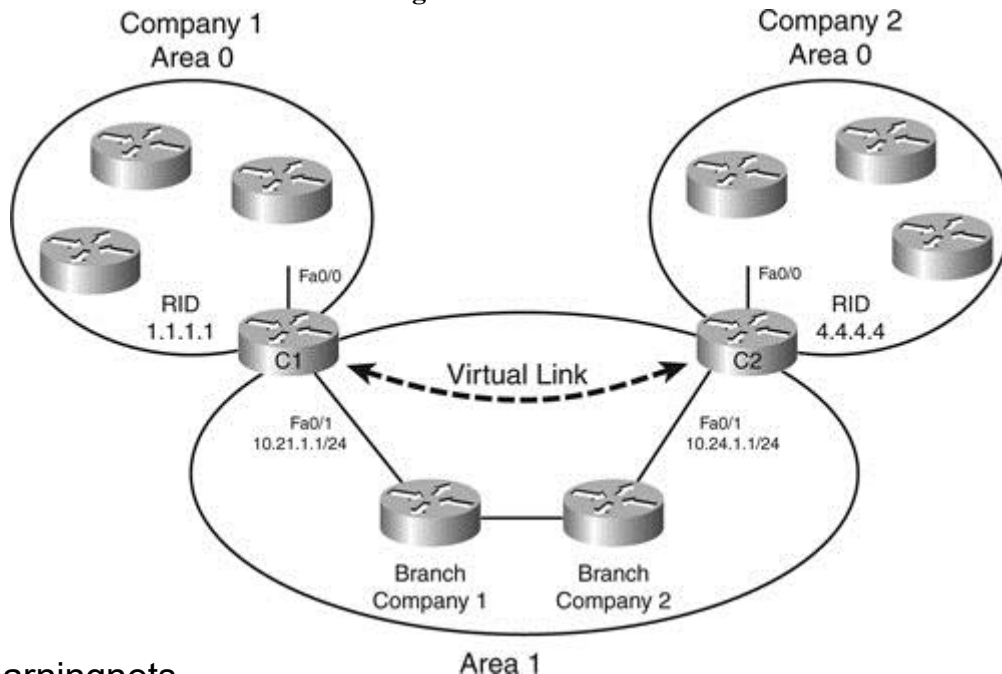
Traffic Between Area 2 ↔ Area 4: Router A ↔ VL ↔ Router B



Second case: Discontiguous Backbone Area



Case 3 solution: Connecting Two Backbone Areas with a Virtual Link



The problems in each case have different symptoms, but the problems all stem from the area design requirements:

- Each area should be contiguous,
- each non-backbone area should connect to the backbone area through an ABR.

When the network does not meet these requirements, engineers could simply redesign the areas. However, OSPF provides an alternative tool called an OSPF **virtual link**:

- An OSPF virtual link allows *two ABRs (or one ABR and one IR) that connect to the same non-backbone non-stubby area* to form a *neighbor relationship* through that non-backbone area, even when separated by many other routers and subnets,
- This virtual link acts like a *virtual point-to-point connection* between the two routers,
- *With that link inside area 0,*
- The routers form a neighbor relationship, inside area 0, and flood **Type 1/2/3** LSAs over that link.
- AS external links are *never* flooded over virtual adjacencies. AS external link advertisements are not summarized over virtual adjacencies during the Database Exchange process.
 - ***Type 4/5 LSAs should be flooded over the virtual link since the area not directly connected to the backbone area has no knowledge of them.***
- The cost of a virtual link is *not* configured. The router calculates the cost as the cost of the intra-area path between the two ABRs defining the virtual link, as calculated using the transit area's LSDB. This cost appears in the virtual link's corresponding routing table entry. When the cost of a virtual link changes, a new router link advertisement should be originated for the backbone area.

The ABRs connected over a virtual link act mostly like any other ABR, with *a few differences*:

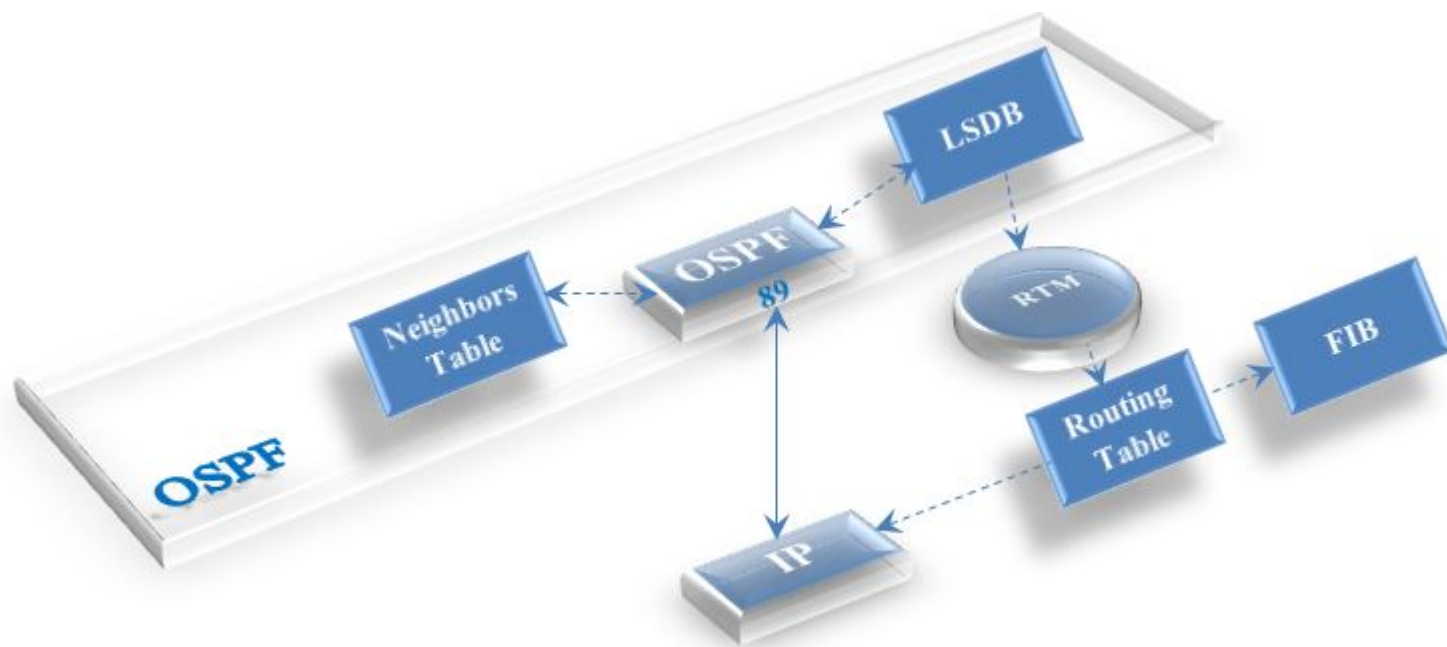
- ABRs send all OSPF messages as *unicasts* to the IP address of the router on the other end of the link.
- the routers also mark the *Do Not Age (DNA)* bit in the LSAs, meaning that all routers on the other side of the virtual link will not expect the LSAs to be re-flooded over the virtual link on the usual 30-minute refresh interval. This helps reduce overhead over the virtual link, which often runs over slower links and less-powerful routers.
- the neighbors choose to not use the usual Hello/Dead interval process over a virtual link. Instead, if all the transit area's routes to reach the router on the other side of the link fail, the virtual link fails. OSPF hellos are "*suppressed*". This means that, *once the virtual link is up, no hellos are exchanged*. OSPF suppresses the hellos because it considers virtual links to be demand circuits. As a result, any changes that you make to the OSPF authentication do not take effect until you clear the OSPF process with the **clear ip ospf process** command.
 - ***So how does a router decide that the virtual link should go DOWN when the connectivity is lost somewhere in between both 'virtual' neighbors?***

4.7 Summary

4.7.1 Topology

Term	Definition
Link state database	The data structure held by an OSPF router for the purpose of storing topology data.
Shortest Path First (SPF)	The name of the algorithm OSPF uses to analyze the LSDB. The analysis determines the best (lowest cost) route for each prefix/length.
Link State Update (LSU)	The name of the OSPF packet that holds the detailed topology information, specifically LSAs.
Link State Advertisement (LSA)	The name of a class of OSPF data structures that hold topology information. LSAs are held in memory in the LSDB and communicated over the network in LSU messages.
Area	A contiguous grouping of routers and router interfaces. Routers in an area strive to learn all topology information about the area, but they do not learn topology information about areas to which they do not connect.
Area Border Router (ABR)	A router that has interfaces connected to at least two different OSPF areas, including the backbone area. ABRs hold topology data for each area, and calculate routes for each area, and advertise about those routes between areas.
Backbone router	Any router that has at least one interface connected to the backbone area.
Internal routers	A router that has interfaces connected to only one area, making the router completely internal to that one area.
Designated Router (DR)	On multiaccess data links like LANs, an OSPF router elected by the routers on that data link to perform special functions. These functions include the generation of LSAs representing the subnet, and playing a key role in the database exchange process.
Backup Designated Router (BDR)	A router on a multiaccess data link that monitors the DR and becomes prepared to take over for the DR, should the DR fail.

4.7.2 Stack



4.7.3 Features

Feature	Description
Routing Protocol	Link-state routing protocol.
Transport	IP, protocol type 89 (does not use UDP or TCP).
Hello interval	Interval at which a router sends OSPF Hello messages on an interface to 224.0.0.5 (All SPF Routers)
Dead interval	Timer used to determine when a neighboring router has failed, based on a router not receiving any OSPF messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.5 (All SPF Routers) and 224.0.0.6 (All Designated Routers).
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used when changes occur and every 30mn
Authentication	Supports MD5 and clear-text authentication.
VLSM/classless	OSPF includes the mask with each route, also allowing it to support discontinuous networks and VLSM.
Route Tags	Allows OSPF to tag routes as they are redistributed into OSPF.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Shortest Path First (SPF)	Uses the Dijkstra algorithm to calculate the SPF tree.
Metric	Based on cumulative cost of all outgoing interfaces in a route. The interface cost defaults to a function of interface bandwidth but can be set explicitly.
Load-Balancing	Equal-cost load-balancing only
Administrative Distance	Default administrative distance is 110 for intra-area, inter-area, external type 1 and external type 2 routes.
Topological Requirements	Two-tier hierarchical design with all areas connected to a central backbone area.
Manual route summarization	Allows route summarization at ABR and ASBR routers only.
Automatic Summarization	Not supported. It can redistribute external classful networks though.
Scalability	High: used in large networks
Diameter	255
Multiprotocol	OSPFv3 over IPv6

4.7.4 Relations With Other Routing Protocols

Cf. [RTM](#)

4.8 Configuration Commands

4.8.1 Defining the Interface Network Types

They can be used to control **static/dynamic neighborships** and **DR/BDR election**.

(config-if)# **ip ospf network** {broadcast | non-broadcast | {point-to-point | point-to-multipoint [non-broadcast]}}

If you use the **non-broadcast** keyword, the **neighbor** command is required.

4.8.2 Dynamic Neighborship

1. (config)#**router ospf** *process-id* Neighboring routers' commands do not have to be configured with the same *process-id* parameter to become neighbors.
2. (config-router)#**network** *net-id wildcard-mask area area-id* Enables OSPF on *matched* interfaces
Or (config-if)#**ip ospf** *process-id area area-id* Enables OSPF on *this* interface

When OSPF is enabled, the router does the following:

- a. Attempts to discover OSPF neighbors on that interface by sending multicast *OSPF Hello* messages
 - b. Includes the connected subnet in future topology database exchanges
3. To match an interface with the **network** command, IOS compares the *net-id* configured in the **network** command with each interface's IP address, while using the configured wildcard-mask as an ACL wildcard mask.
 4. Regardless of the order in which the **network** commands are added to the configuration, IOS puts these commands into the configuration file with the most specific (most binary 0s) wildcard mask first. IOS lists the **network** commands in this sorted order in the configuration.
 5. The first **network** command that matches an interface, per the order shown in the output of the **show running-config** command, determines the OSPF area number associated with the interface.
- From cciethebeginning.wordpress.com: the /32 advertisement limitation of loopback interfaces can be overcome by either changing OSPF network type to point-to-point or creating summary route with the desired mask.

4.8.3 Controlling Neighborships

4.8.3.1 ... and Reducing Multicasts: Static Neighborship

(config-router)#**neighbor** *ip-address* [**cost** *neighbor-cost-value*] [**priority** *neighbor-priority-value*]

- Defines a neighbor on *NBMA, point-to-multipoint and point-to-multipoint non broadcast networks only*.
 - Only the router starting the exchange needs the statement(s) (*unsecure*).
 - On *NBMA and point-to-multipoint non-broadcast* networks, you *must* use the **neighbor** command to identify neighbors. Assigning a **cost** to a neighbor is *optional*, instead of just using the interface cost for all neighbors.
 - On *point-to-multipoint* (broadcast networks), you *can* use the **neighbor** command, and you *must* specify a **cost** to that neighbor, instead of just using the interface cost for all neighbors.
 - The OSPF configuration still needs a **network** command that matches the interface referenced by this command.
- If at least one OSPF static neighbor is defined on an interface, no dynamic neighbors can be either discovered or continue to work if already discovered.

4.8.3.2 Preventing Unwanted Neighbors

4.8.3.2.1 Passive Interfaces

(config-router)#**passive-interface** *type* #

Sets that interface as passive : hellos and updates are not exchanged through the specified router interface. The interface address appears as a stub network in the OSPF domain and is advertised on other enabled interfaces.

(config-router)#**passive-interface default**

Sets all interfaces as passive by default

(config-router)#**no passive-interface** *type* #

Sets that interface as not passive (needed if previous command entered)

<https://t.me/learningnets>

4.8.3.2.2 Authenticating OSPF Messages

○ **Step 1:** Configure key:

a) Clear text authentication

(config-if)#**ip ospf authentication-key** *key-value* For this interface

b) MD5 authentication

(config-if)#**ip ospf message-digest-key** *key-number md5 key-value* For this interface

➤ The same *key-number* on the neighbor router must have the same *key-value*.

➤ The router *uses the most recently added key* for authenticating messages.

○ **Step 2:** Enable OSPF authentication on an interface:

a) No authentication (*default*): Type 0

(config-if)#**ip ospf authentication null** For this interface

Or (config-router)#**area** *area-id authentication null* For all interfaces of this area

b) Clear text authentication: Type 1

(config-if)#**ip ospf authentication** For this interface

Or (config-router)#**area** *area-id authentication* For all interfaces of this area

c) MD5 authentication: Type 2

(config-if)#**ip ospf authentication message-digest** For this interface

Or (config-router)#**area** *area-id authentication message-digest* For all interfaces of this area

- Unlike EIGRP authentication, OSPF authentication does not allow the configuration of a key chain with time-based authentication keys. However, multiple keys can be configured on an interface, each with a different key number.
- The router *uses the most recently added key* for authenticating messages. If a router having several MD5 keys on an interface detects that at least one neighbor has not yet started using the most recent key, it engages in a simple key migration procedure: it sends each OSPF packets multiple times, with each instance of the packet authenticated by a particular MD5 key configured on the interface, one instance for each key. This ensures a smooth, gradual migration.

4.8.4 Optimizing Convergence delay

4.8.4.1 Hello/Dead Timers

(config-if)#**ip ospf hello-interval** *seconds*

Sets the hello timer **for all the neighbors** reachable on that interface (default: 10s on Ethernet ; 30s on NBMA (“3xEthernet”).

Also changes **dead-interval**

Makes the neighborships fail on that router after the **dead timers** expire

(config-if)#**ip ospf dead-interval** {*seconds* | **minimal hello-multiplier multiplier**}

Sets the dead timer **for all the neighbors** reachable on that interface (default: **4xhello-interval** : 40s on Ethernet ; 120s on NBMA (“3xEthernet”))

(“3xEthernet”))

Or sets the dead interval to one second, and the Hello interval to a fraction of a second based on the *multiplier* (3-20) with **minimal hello-multiplier**

4.8.4.2 LFA Feasible Successors

Out of CCNP ROUTE scope

Loop Free Alternate Fast Reroute is available for [OSPF](#) in Cisco IOS 15.1(3)S and XE 3.4S.

The OSPFv2 Loop-Free Alternate Fast Reroute feature is supported only in global Virtual Routing and Forwarding (VRF) OSPF instances.

The OSPF implementation allows you to specify numerous alternate path selection criteria. For example, you might prefer alternate paths that don't go over a shared risk link group, through the same LAN interface, or through the next-hop router.

4.8.5 Controlling DR/BDR Election

4.8.5.1 Interface Priority

(config-if)#**ip ospf priority** *value* Controls the DR/BDR election (default 1, max 255)

4.8.5.2 Router ID

(config-router)#**router-id** *a.b.c.d* Sets the RID

4.8.6 Optimizing Bandwidth Utilization

4.8.6.1 Load Sharing → Cf. [Load-Balancing Summary](#)

4.8.6.1.1 Equal Metric Routes

(config-router)#**maximum-paths** *number* Sets the maximum number of allowed routes in the routing table for each prefix/prefix length. The default setting of 4 is often big enough, because most internetworks do not have enough redundancy to have more than four possible routes.

4.8.6.1.2 Making Equal Metric Routes ... or choosing One Route Instead of Another: Cost Components

(config-if)#**ip ospf cost** *value* Changes **Cost** in the formula

(config-if)#**bandwidth** *value (kbits/s)* Changes **Bandwidth** in the formula (1544 by default on serial links and sub-interfaces, actual speed on Ethernet)

(config-router)#**auto-cost reference-bandwidth** *value (Mbits/s)* Changes **Reference Bandwidth** in the formula.

4.8.6.2 Setting the MTU

(config-if)#**ip mtu** *value* Sets the IP MTU on an interface.

(config-if)#**mtu** *value* Sets the MTU on an interface for all Layer 3 protocols.

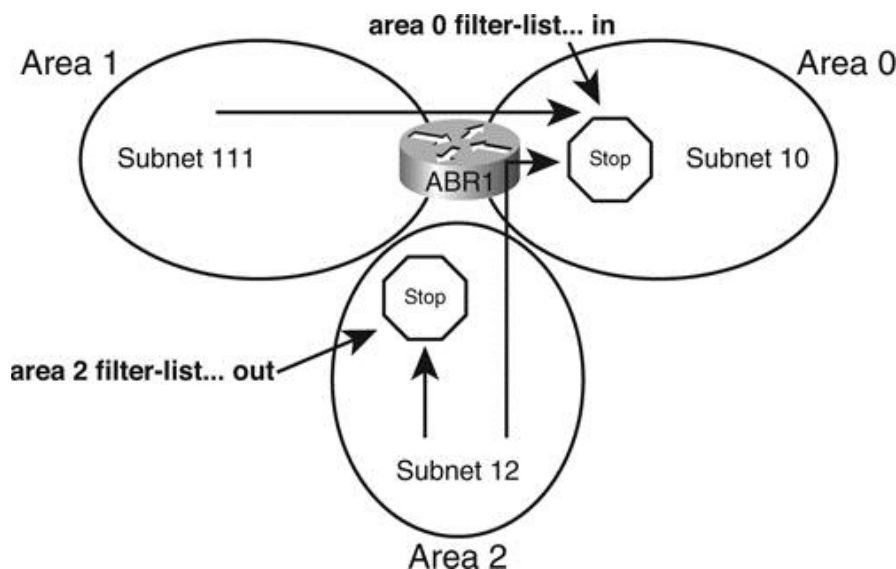
4.8.7 Optimizing Router Resources

4.8.7.1 Limiting Reachability: Route Filtering

4.8.7.1.1 Filtering Type 3 LSA on ABRs

(config-router)#**area** *number* **filter-list** *prefix name* **in** | **out**

- When **in** is configured, IOS filters prefixes *being created and flooded by the ABR into the configured area, (not including the ones received from other ABRs – describing subnets from another area - and ‘relayed’ by the ABR).*
- When **out** is configured, IOS filters prefixes coming out of the configured area.



- *Only one filtering per area per direction.*
- *Caution: if the Type 3 LSA describing the ASBR subnet is filtered out from its (T)NSSA area, on its attached ABR, then all the external routes and default route originating from the ASBR will disappear from all routing tables outside the (T)NSSA area!*
- *You cannot filter subnets from a non-backbone area into another non-backbone area with this command!*

4.8.7.1.2 Filtering Routes Added to the Routing Table on Each Intended Router

(config-router)#**distribute-list** [[*access-list-number* | *name*] | [**prefix** *list-name*] | [**route-map** *name*] **in** [*interface-type* #]

- The command requires either an **in** or **out** direction. Only the **in** direction works for filtering routes as described in this section.
- The command must refer to either a numbered ACL, named ACL, prefix list, or route map. Regardless, routes matched with a **permit** action are allowed into the *routing table*, and routes matched with a **deny** action are filtered.
- Optionally, the command can include the *interface-type* # parameters. The router compares these parameters to the route's *outgoing* interface.

4.8.7.1.3 Filtering Type 5/7 LSA on ASBRs

- *Either with a route-map during redistribution into OSPF with redistribute route-map (cf. [§Redistribution into OSPF](#))*
- *Or with the distribute-list out command (cf. [§distribute-list out](#))*

4.8.7.2 Without Limiting Reachability

4.8.7.2.1 Route Summarization

4.8.7.2.1.1 Manual Inter-Area Summarization on ABRs

(config-router)#**area** *area-id* **range** *prefix mask* [**cost** *cost*] [**not-advertise**]

- The configured area number refers to the area where the subnets exist; *the summary will be advertised into all other areas connected to the ABR.*
- The ABR compares the summary route's range of addresses with all intra-area OSPF routes, in the *origin* area, for which the ABR is creating Type 3 LSAs. If at least one *subordinate subnet* exists (subnets that sit inside the range), then the ABR *advertises the summary route as a Type 3 LSA.*
- The ABR does not advertise the subordinate subnet's Type 3 LSAs.
- The ABR assigns a metric for the summary route's Type 3 LSA, by default, to match the lowest metric among all subordinate subnets, *except if the "(config)#no compatible rfc1583" command is issued, in which case the summary route's metric is the biggest metric of all subordinate subnets.*
- **cost** explicitly sets the cost of the summary.
- If no subordinate subnets exist, the ABR does not advertise the summary.
- The optional **not-advertise** option on this command tells the ABR to not advertise the Type 3 LSA for the summary route, making it possible to do the *equivalent of Type 3 LSA filtering.*

4.8.7.2.1.2 Manual External Summarization on ASBRs

(config-router)#**summary-address** *prefix mask* [**not-advertise**]

- The ASBR compares the summary route's range of addresses with all routes redistributed into OSPF from all outside route sources on that ASBR to find any subordinate subnets (subnets that sit inside the summary route range). If at least one subordinate subnet exists, the ASBR advertises the summary route.
- The ASBR does not advertise the subordinate subnets.
- To create the summary, the ASBR actually creates a Type 5 *or* Type 7 LSA for the summary route.
- The ASBR assigns the summary route the same metric as the lowest metric route amongst all subordinate subnets, *The "(config)#no compatible rfc1583" command has no effect for external summarization!*
- If no subordinate subnets exist, the ASBR does not advertise the summary.
- The optional **not-advertise** option on this command tells the ABR to not advertise the Type 5/7 LSA for the summary route, making it possible to do the *equivalent of Type 5/7 LSA filtering.*

4.8.7.2.2 Default Routes and Stub Areas

4.8.7.2.2.1 Domain-wide on ASBRs

(config-router)#**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*]

- On *ASBRs connected to non-stubby area(s)*.
- With all default parameters, this command injects an *external default route -- 0.0.0.0/0 --* into OSPF, as an *External Type 2* route, using a Type 5 LSA, with *metric 1*, throughout the OSPF domain, but *only if* a default route exists in that router's IP routing table. This logic relies on the fact that the ASBRs typically either have a static default route pointing to the connected *ISP* router, or they learn a default route from the ISP using BGP (each ISP is advertising a default route in this case). All the routers then learn a default route, based on the Type 5 LSAs for 0.0.0.0/0 as flooded by the ASBRs.
- With the **always** parameter, the default route is advertised even if there is no default route in the router's routing table.
- The **metric** keyword defines the metric listed for the default route (default 1).
- The **metric-type** keyword defines whether the LSA is listed as external type 1 or external type 2 (default).
- The decision of when to advertise, and when to withdraw, the default route is based on matching the referenced **route-map** (*applied on the routing table*) with a permit action, i.e. only if the defined prefixes are present in the routing table (a default route in the routing table is not necessary anymore in this case).

4.8.7.2.2.2 Area-wide on ABRs/ASBRs

4.8.7.2.2.2.1 Stubby and Totally Stubby Areas

(config-router)# **area** *area-id* **stub** [**no-summary**]

Action	Configuration Steps
Stubby	Configure area <i>area-id</i> stub on each router in the area
Totally stubby	Configure area <i>area-id</i> stub no-summary command on the ABRs Configure area <i>area-id</i> stub , without the no-summary keyword, on all other routers in the area
Set the metric of the default route	Configure area <i>area-id</i> default-cost <i>cost</i> on an ABR; can differ from ABR to ABR. Default value is 1.

4.8.7.2.2.2.2 NSSA and Totally NSSA Areas

(config-router)#**area** *area-id* **nssa** [**no-summary**] [**no-redistribution**]
[**default-information-originate** [**metric** *metric-value*] [**metric-type** *type-value*] [**nssa-only**]]

- For normal NSSA areas, omit the **no-summary** keyword,
 - For Totally NSSA areas, use the **no-summary** (no Type 3 LSA) keyword, *only on the ABRs, not on the ASBR*,
 - **no-redistribution** is used when the router is a normal and NSSA ASBR and you want the **redistribute** command to import routes only into the normal areas, but not into the NSSA area,
 - **default-information-originate** on *ABRs or ASBRs connected to a NSSA or TNSSA area* generates an external Type 7 (N2) default route into the (T)NSSA area, *without the need for the ABR to have a default route in its routing table*. The ASBR *does need* a non-OSPF default route in its routing table,
 - **nssa-only** limits the *default* Type 7 LSA to this NSSA area only.
- Default route *does not appear* in the (T)NSSA Area if **default-information-originate** is not specified in previous command.

(config-router)#**area** *area-id* **default-cost** *cost* Another way to specify a cost on a ABR or ASBR for the default summary route that is sent into a NSSA or Totally NSSA (same command as for Stub area), and default *cost* is 1.

4.8.8 Extending the Backbone Area with Virtual Links

(config-router)#**area** *transit-area-id* **virtual-link** *remote-router-rid* [**hello-interval** *seconds*] [**dead-interval** *seconds*]

- The *transit-area-id* refers to the transit area –which must not be a Stubby area– over which the packets flow between the two routers.
- The optional configuration of Hello and Dead intervals, normally configured as interface subcommands, must be configured as additional parameters.
- The optional configuration of OSPF neighbor *authentication parameters*, normally configured as interface subcommands, must be configured as additional parameters.

Type (Name)	Type (Number)	Command Syntax for Virtual Links
none	0	<code>area num virtual-link router-id authentication null</code>
clear text	1	<code>area num virtual-link router-id authentication authentication-key key-value</code>
MD5	2	<code>area num virtual-link router-id authentication message-digest message-digest-key key-num md5 key-value</code>

4.8.9 Summary of Issues related to OSPF over Multipoint Frame Relay with a ‘Hub and Spokes’ Topology

Multipoint interfaces include serial physical interfaces and multipoint sub-interfaces.

Network Types	Dynamic Neighborhood	DR/BDR
Broadcast	✓ ¹	✓ ^{3/4}
Point-to-Point (PTP)	✓	–
Non Broadcast	– ²	✓ ^{3/4}
Point-to-Multipoint (PTM)	✓ ¹	–
Point-to-Multipoint Non Broadcast (PTMNB)	– ²	–

Issues	Solutions
No Neighborhood Between Spokes	⇒ Full mesh (might not be applicable), depends on the interfaces network type
¹ Dynamic Neighborhood Issue	⇒ broadcast keyword in frame-relay map ip command
² Static Neighborhood Issue	⇒ Static neighbor definition
³ DR/BDR Issue	⇒ DR/BDR on hubs and broadcast keyword in frame-relay map ip command
⁴ Unreachability Issue between Spokes (for interfaces in common subnet)	⇒ Static IP address ⇔ DLCI mapping on each spoke for all other spokes ⇒ or change of network type (⇒ PTM/PTMNB) ⇒ or change of interface type (⇒ PTP)

4.9 Verification Commands

#show ip protocols

Lists **network** parameters for each routing process :

- RID - Net-id Wildcard-mask - Area ID,
- Type of router (ABR, ASBR...),
- Passive interfaces,
- Reference bandwidth unit,
- Routing Information Sources, Administrative distances, Elapsed time since Last Update.

- #show ip ospf** [*process-id*] Displays general information about this OSPF *process-id*:
- RID
 - Total number of areas, number of normal areas, stub areas and nssa
 - Per Area:
 - number of interfaces in this area
 - type of area
 - whether a default route is injected or not
 - whether authentication is enabled or not
 - number of times SPF executed and when was the last execution
- #show ip ospf neighbor** [*type number*] [*neighbor-id*] [**detail**] Lists Known valid dynamic and static neighbors:
- Neighbor RID – Neighbor IP address - Neighbor Priority
 - Neighbor State (DOWN-ATTEMPT-INIT-2WAY-EXSTART-EXCHANGE-LOADING-FULL / DR-BDR-DROTHER)
 - *Local* Current Dead Timer (due in ...) / Neighbor - *Local Out Interface*
 - Only with *neighbor-id* or **detail** :
 - Area ID,
 - DR and BDR in the subnet,
 - Elapsed time since neighborhood was formed
- #show ip ospf interface** [*type number*] [**brief**] Lists working *non passive* interfaces on which OSPF is enabled :
- Area ID - IP address/len
 - Cost - State (DR, BDR, DROTHER, P2P, P2MP)
 - Not with **brief** :
 - RID - Priority
 - Network Type (BROADCAST, NON BROADCAST, PTP, PTM, PTM NON BROADCAST)
 - DR and BDR in the subnet: RID – IP address
 - Hello and Dead Timers, current Hello Timer (due in ...)
 - Neighbors count, Adjacent Neighbors count and RID
 - Authentication information
- #show ip ospf** [*process-id area-id*] **database** [**keyword**] Displays information about different LSAs, with detailed information depending on **keyword** :
- **router** [*link-state-id*] [**adv-router** [*rid*]]: Type 1 LSA [of this router] [advertised by this router]
 - **network** [*link-state-id*] [**adv-router** [*rid*]]: Type 2 LSA [advertised by this router]
 - **summary** [*link-state-id*] [**adv-router** [*rid*]]: Type 3 LSA [advertised by this router]
 - **asbr-summary** [*link-state-id*] [**adv-router** [*rid*]]: Type 4 LSA [advertised by this router]
 - **external** [*link-state-id*] [**adv-router** [*rid*]]: Type 5 LSA [advertised by this router]
 - **nssa-external** [*link-state-id*] [**adv-router** [*rid*]]: Type 7 LSA [advertised by this router]
 - **adv-router** [*rid*]: All the LSAs of the specified router. If no RID is included, the information is about the local router itself (in this case, the same as **self-originate**).
 - **self-originate** [*link-state-id*]: LSAs from only the local router.
 - **database-summary**: How many of each type of LSA for each area and for each process there are in the database, and the total.

#show ip ospf database router [*router-RID*]

Type 1 Router link states

- LSID: Router RID
- Advertising router: Router RID
- Number of links: n
 - Links connected to a *transit* network:
 - Link ID: DR IP address
 - Link Data: Router IP address
 - Metric: cost of the interface
 - Links connected to a *stub* network:
 - Link ID: subnet number
 - Link Data: subnet mask
 - Metric: cost of the interface

#show ip ospf database network [*DR-IP-address*]

Type 2 Network link states

- LSID: interface IP address of the DR that created this LSA
- Advertising router: DR RID
- Mask: subnet mask
- Attached routers: all RIDs of attached routers on this subnet

#show ip ospf database summary [*subnet-IP-Address*]

Type 3 Summary link states

- LSID: subnet prefix
- Advertising router: ABR RID
- Mask: subnet mask
- Metric: cost to reach the subnet from the ABR

#show ip ospf database asbr-summary [*ASBR-RID*]

Type 4 ASBR Summary link states

- LSID: ASBR RID
- Advertising router: RID of the ABR that created and flooded this LSA
- Metric: Cost to reach the ASBR from the ABR

➤ *Not created by ABRs for (T)NSSA ASBRs***#show ip ospf database external** [*subnet-IP-Address*]

Type 5 AS External link states

- LSID: External subnet number
- Advertising router: ASBR or “first” ABR RID (*when Type 7 LSA becomes Type 5 LSA on that ABR*)
- Mask: External subnet mask
- Metric type: E2 or E1 as set by the ASBR
- Metric: External Cost as set by the ASBR
- Forward address : ASBR IP address (*only when Type 7 LSA becomes Type 5 LSA*):
 - *Used by routers outside (T)NSSA areas to calculate the costs of external E1/E2 routes and default E1/E2 routes originating from the ASBR with the help of the appropriate Type 3 LSA.*
 - *Except if “(config-router)#area area-id nssa translate type-7 suppress-fa” has been configured, in which case “Forward Address” is set to 0.0.0.0. The E1 routes metrics start from the ABR, so we lose the costs from the NSSA ASBR to the ABR.*

#show ip ospf database nssa-external [*subnet-IP-Address*]

Type 7 NSSA External

- LSID: External subnet number
- Advertising router: ASBR RID (*is changed by the first ABR when converted into the corresponding Type 5 LSA during the flooding process and “forward address” points to the original ASBR*)
- Mask: External subnet mask
- Metric type: N2 or N1 as set by the ASBR
- Metric: External Cost as set by the ASBR
- Forward address : ASBR IP address

(config-router)#**max-lsa** *maximum-number* Limits the maximum number of LSAs learned from other routers

#**show ip ospf border-routers** Displays the Routing Table entries to an ABR and ASBR:

- Type of route (intra-Area, Inter-Area)
- RID – Cost to the ABR/ASBR – Next hop IP Address – Outgoing Interface
- Router type of the destination (ABR, ASBR) - Area ID of the area from which this route is learned.

This command can be misleading, depending on the type of Area which the local router is linked to:

- On routers connected to a Non Stubby Area:

The information listed by the command is an image of the reality, with two exceptions:

- *ABRs linked to NSSA areas are seen as "ABR/ASBR", despite the fact that they are not directly linked to an AS, even though they transform Type 7 LSAs received from true ASBR into Type 5 LSAs for flooding into the Backbone Area. These ABRs are even seen as ASBRs from IRs inside Non Stubby Areas.*
- *ASBRs within NSSA Areas are not seen from outside of these areas, except from ABRs connected to the same NSSA.*
- *The reason why all this is because **no Type 4 LSAs** are generated for the (T)NSSA ASBRs.*

- On routers inside a Stubby Area (Stubby, Totally Stubby, NSSA or Totally NSSA):

They view only "local" ABRs(why ???) and ASBRs (no Type 4 LSA allowed in). When these local ASBRs are connected to the Backbone, they are seen as ABRs.

➤ To join in the CLN thread: "[Tricky views of OSPF border routers from Stubby and Non Stubby Areas](#) "

#**show ip route** [*prefix mask*] [*longer-prefixes*] [*summary*] Lists the contents of the routing table, with :

- O : OSPF Intra-Area routes
- O IA : Inter-Area routes
- O E1 : External Type 1 routes
- O E2 : External Type 2 routes
- O N1 : NSSA External Type 1 routes
- O N2 : NSSA External Type 2 routes
- Connected/Static/RIP/D/O/IA/E1/E2/N1/N2/* (candidate for Default Route)
- Gateway of Last Resort
- Subnet number/length **OR** All the subordinate routes of that Subnet (with **longer-prefixes**) - [AD/metric] - Next-hop router - hh:mm:ss since last entry update - Out interface
- **summary**: number of subnets/C/S/O/IA/E1/E2/N1/N2, memory used, ...

#**show ip ospf virtual-links** Shows some items unique to virtual links:

- State of special interface Virtual Link OSPF_VLn: UP/DOWN
- Bit 'DoNotAge' (DNA) - Transit Area number
- Cost of that special interface: Σ intra-area costs on the path between the two endpoints of the virtual link
- Type of that special interface: POINT-TO-POINT
- Adjacency state: TWO-WAY, FULL
- Authentication information: type and youngest key

#**debug ip ospf hello** Displays Hellos exchanges

#**debug ip ospf events** Displays events, such as flooding information, designated router selection, adjacencies, and shortest path first (SPF) calculation.

#**debug ip ospf adj** Displays information on adjacency events, such as DR/BDR election, authentication information, packets being dropped due to a Time-to-Live (TTL) security check

#**debug ip ospf packet** Displays information about each OSPF packet received.

<https://t.me/learningnets>

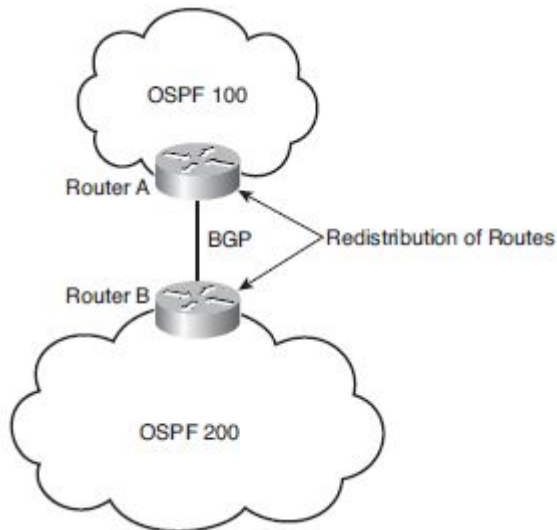
5 Path Control (Traffic Engineering)

5.1 Route Redistribution

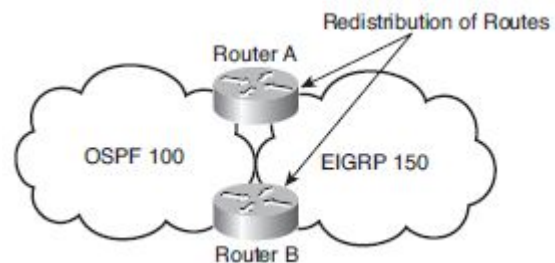
5.1.1 Reasons

- Migration from an older routing protocol to a new routing protocol.
- Mixed-vendor environment in which Cisco routers might be using EIGRP and other vendor routers might be using OSPF.
- Different administrative domain between company departments using different routing protocols.
- Mergers and acquisitions in which the networks initially need to communicate. In this scenario, two different EIGRP processes might exist.

Redistribution of BGP Routes



Redistribution Between IGP



5.1.2 Route Feedback

Route feedback occurs when a routing protocol learns routes from another routing protocol and then announces the routes back to the other routing protocol. In past figure, OSPF should not advertise the routes it learned from EIGRP on Router A back to EIGRP on Router B. And EIGRP should not announce the routes it learned from OSPF on Router B back to OSPF on Router A.

5.1.3 Tools

You can use **access lists**, **distribution lists**, **prefix-lists** and **route maps** when redistributing routes. You can use these methods to specify (select) routes for redistribution, to set metrics, or to set other policies for the routes. They are used to prevent loops in the redistribution. They are also used to control routes redistribution direction.

5.1.4 Methods

Redistribution can be accomplished by two methods:

- **Two-way redistribution:** In two-way redistribution, routing information is exchanged between both routing protocols. No static routes are used in this exchange. Route filters are used to prevent routing loops. Routing loops can be caused by one route protocol redistributing routes that were learned from a second route protocol back to that second routing protocol.
 - **One-way redistribution:** One-way redistribution only allows redistribution from one routing protocol to another. Normally, it is used in conjunction with a default or static route at the edge of a network.
- **Redistribution is not necessary between IGRP and EIGRP when they are defined on the same ASN.** Two-way redistribution between these processes occurs automatically.
- It must be manually enabled when IGRP and EIGRP processes are on different ASNs.

5.1.5 Seed/External Metrics

5.1.5.1 Configured Values

You should configure the metric of the redistributed routes to a metric other than 0. You can configure the metric during the redistribution or configure a default seed metric, **except for IS-IS**.

5.1.5.2 Default Values

From PacketLife.net with some additions:

Source	Into				
	RIP	EIGRP	OSPF	IS-IS	BGP (MED)
Connected	1	Internal metric	20 (E2)	0	0
Static	1	Internal metric	20 (E2)	0	0
RIP	Source Metric (IPv6)	Infinite*	20 (E2)	0	IGP metric
EIGRP	Infinite*	Source Metric	20 (E2)	0	IGP metric
OSPF	Infinite*	Infinite*	Source Metric	0	IGP metric
IS-IS	Infinite*	Infinite*	20 (E2)	?	IGP metric
BGP	Infinite*	Infinite*	1 (E2)	0	N/A

Infinite*: The seed metric value should be set during redistribution (**redistribute**) or as default value (**default-metric**).

5.1.6 Route Filtering at a Redistribution Point

- Avoids routing loops
- Avoids suboptimal routing
- Prevents certain routes from entering the domain

One redistribution point where filtering is important is in a **multihomed BGP connection** with multiple Internet providers. When BGP routes get exchanged with multiple Internet Service Providers (ISP), route filtering is used to prevent advertisement of private addresses and addresses that are out of scope of the domain. Route filtering is also used to filter routes at the redistribution of BGP into IGP.

5.2 Redistribution into OSPF

Unlike EIGRP:

- OSPF *does* have useful default metrics for redistributed routes, but OSPF does use the same general methods to configure metrics for these routes,
- OSPF creates LSAs to represent each external route, and OSPF must then apply some much different logic than EIGRP to calculate the best route to each external subnet.
- **Redistribution turns the OSPF router into an ASBR, even if only Connected/Static routes are injected into the domain by an internal router**, instead of a router at the edge of the AS.

Redistribution into OSPF uses the following **seed default metrics**:

- When taking from **BGP**, use a default metric of **1**.
- When taking from **another OSPF process**, take the **source route metric**.
- When taking from **all other sources**, use a default metric of **20**.
- Create a **Type 5 LSA** for each redistributed route **not inside an (T)NSSA area**:
 - A Type 4 LSA is also generated by the ABR receiving this type 5 LSA to point to the ASBR
 - Forward Address is 0.0.0.0, or the router IP address only if:
 - the type 5 LSA comes from a type 7 LSA
 - the type 5 LSA represents a redistributed static route– *which does not make sense since a type 4 LSA has also been generated*,
- Create a **Type 7 LSA** for each redistributed route if **inside an (T)NSSA area**:
 - A Type 4 LSA is NOT generated by the ABR to point to the ASBR, because this ABR – forwarding the type 7 LSA as type 5 LSA - is now an ASBR, so other ABRs do create type 4 LSAs in their own areas when they send this newly created type 5 LSA,
 - Forward Address is the first ASBR IP address,
- Use **External metric type 2: E2**, and **internal OSPF costs are not added to the cost of the route**,
- **Redistribute only routes of classful networks** from the routing table, and *not* routes for subnets.

When redistributing routes into OSPF, use the **subnets** keyword to permit subnetted routes to be received. If you do not use it, only the classful network route is redistributed, without any subnetworks. In other words, OSPF performs automatic summarization to IP classful network values. Also, unlike EIGRP and RIPv2, OSPF does not need a metric to be specified during redistribution, neither does it need a default seed metric to be specified because it uses a default metric for redistributed routes.

By default, redistributed routes are classified as External Type 2 (E2) in OSPF. You can use the **metric-type** keyword to change the external route to an External Type 1 (E1). The network design can take into account the after-redistribution seed cost (Type 2) or the after-redistribution seed cost plus the path cost (Type 1).

5.2.1 Determining the Next-Hop for External Routes

5.2.1.1 Intra-Area

5.2.1.1.1 E2/N2

The following list spells out the mechanics of the calculation used to **break the tie** when multiple equal external cost (as set by the ASBR(s)) E2/N2 routes exist for a particular subnet, **when the ASBR is located in the same area as the local router**:

- **Step 1:** Find the ASBR(s) as listed in the Type 5/7 LSA(s) advertising the lowest external cost for that external route.
- **Step 2:** Calculate the lowest cost route to reach any of the ASBR(s) based on the intra-area LSDB topology.
- **Step 3:** Use the outgoing interface and next hop based on the best route to reach the ASBR (as chosen at Step 2).
- **Step 4:** The route metric is unchanged—it is still simply the value listed in the Type 5/7 LSA.

5.2.1.1.2 E1/N1

OSPF routers calculate the metrics of E1 routes by adding:

- The best intra-area cost to reach the ASBR (per that area LSDB)
- The external cost for the route (per Type 5/7 LSA) as set by the ASBR.

5.2.1.2 Inter-Area

5.2.1.2.1 E2

The following list spells out the mechanics of the calculation used to **break the tie** when multiple equal external cost E2 routes exist for a particular subnet, **when the ASBR is located in another area than the local router**:

- **Step 1:** Find the ASBR(s) as listed in the Type 5 LSA(s) advertising the lowest cost for that external route.
- **Step 2:** Calculate the lowest cost route to reach any of the ASBR(s) based on 2 sub-steps:
 - **Step 2.1:** Calculate the cost to reach the ABR, based on the local area's topology database.
 - **Step 2.2:** Add the cost from the ABR to the ASBR, as listed in a Type 4 LSA, *except for NSSA ABR in which case the Type 3 LSA describing the NSSA ASBR subnet is used instead, cf. [§Special Case Involving an NSSA or TNSSA ASBR for Inter-Area E2/E1](#)*.
- **Step 3:** Use the outgoing interface and next hop based on the best route to reach the ASBR (as chosen at Step 2).
- **Step 4:** The route metric is unchanged—it is still simply the value listed in the Type 5 LSA.

5.2.1.2.2 E1

OSPF routers calculate the metrics of E1 routes by adding:

- The best intra-area cost to reach the ABR per that area LSDB,
- The cost from that ABR to the ASBR per Type 4 LSA, *except for NSSA ABR in which case the Type 3 LSA describing the NSSA ASBR subnet is used instead, cf. [§Special Case Involving an NSSA or TNSSA ASBR for Inter-Area E2/E1](#)*,
- The external cost for the route per Type 5 LSA.

5.2.1.2.3 Special Case Involving a (T)NSSA ASBR for Inter-Area E2/E1

ABRs connected to an (T)NSSA area do not create Type 4 LSA into the backbone area concerning the ASBRs connected to the same (T)NSSA area, which means OSPF routers must use another way to calculate the metrics of inter-area E2/E1 routes...

The routers outside the (T)NSSA area receive Type 5 LSAs originating from the (T)NSSA ABR, which received them as Type 7 LSAs from the (T)NSSA ASBR. These Type 5 LSAs have a field called 'Forward Address' which in this case contains the (T)NSSA ASBR IP address.

The routers outside the (T)NSSA area also receive Type 3 LSAs describing the subnets from this (T)NSSA area, including the subnet 'containing' the ASBR IP address. All they need to do is to search within their own database the correct Type 3 LSA (the one with the subnet/mask which is the home of the NSSA ASBR), and they use the cost described in that LSA to add it to the cost to reach the ABR, and the job is done!

- But there is one major pitfall: on the (T)NSSA ABR, if we filter out the Type 3 LSA describing the ASBR subnet, then all the external routes and default route originating from the ASBR will disappear from all routing tables outside the (T)NSSA area!

So now we know we should be cautious with the "area id filter-list prefix name in | out" command!

- To join in CLN thread: "[No Type 4 LSA for NSSA ASBRs](#)"

5.2.2 A Brief Comparison of E1 and E2 Routes:

- For E1 routes, both the external cost and internal OSPF cost matters to the choice of best route
- For E2 routes, only the external cost matters to the choice of best route (unless a tie needs to be broken.)

The benefits of the different external route types apply mostly to when *multiple* ASBRs advertise the same subnet.

For example, imagine two ASBRs, ASBR1 and ASBR2, between OSPF and another routing domain:

- If the goal is to **always send traffic through ASBR1**, you could:
 - use E2 routes and set the metric for ASBR1 redistributed routes to a *lower metric* than ASBR2. Because routers ignore the internal metrics when calculating the E2 metrics, then every router will choose ASBR1 as the better ASBR.
 - Or use E1 routes on ASBR1 and E2 routes on ASBR2 (for a given prefix/length, OSPF always prefers an E1 route over an E2 route).
- Conversely, if the goal were to **balance the traffic, and make each router pick the closest ASBR**:
 - both ASBRs could set the *same external metric* to their redistributed routes, and make all the routes either *Type E1 or E2*. As a result, routers closer to each ASBR choose best routes based on the lower OSPF internal costs.
- **If the goal is to send the traffic to the appropriate ASBR depending on the external Routing Protocol (RP), then:**
 - **Use E2 routes on both ASBR1 and ASBR2 with default metric; order of choice: BGP → EIGRP/RIP/IS-IS → OSPF (if metric > 20)**
 - **Or use E2 routes on both ASBR1 and ASBR2 with different external metrics if you need to favor other RPs.**

5.2.3 External LSAs with Overlapping Subnets

It is possible to have same network number with different masks imported into OSPF from different routing protocols.

The first one is installed normally into the LSDB.

The newly received LSA which overlaps with the first one is installed as its **subnet broadcast number** (LSID) instead of its subnet number into the LSDB.

5.3 Inter-Domain Routing Issues with Multiple Redistribution Points

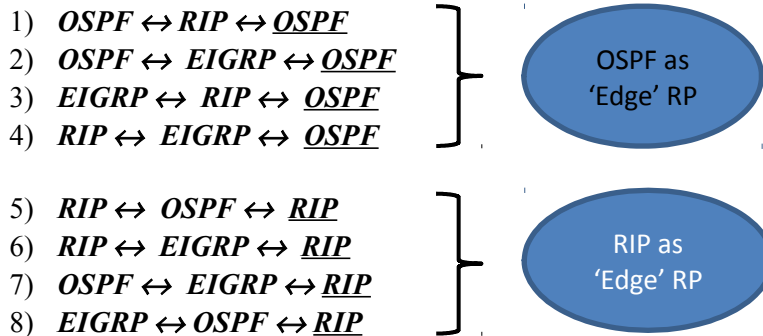
5.3.1 Between 2 Domains

- *Inter-Domain Routing Issues (IDRI) only between RIP ↔ OSPF, except if at least one internal router injects a route by means of redistribution (Connected, Static), in which case we would have an issue with different kind of RPs.*
- *To solve this issue, for each of the two routing protocols, the AD (Administrative Distance) used for internal routes for one routing protocol must be better than the AD used for external routes by the other routing protocol, to be applied on all redistributing routers:*

RP1 internal AD < RP2 external AD

5.3.2 Between 3 Domains

Considering the last 2 domains, there's an IDRI only for:



4) and 7) represent the same topology.

1), 3), 5) and 8) have already been solved by the [Between 2 Domains](#) section.

Several solutions exist for the topologies 2), 4) and 6) involving EIGRP and OSPF or RIP:

5.3.2.1 Setting AD Values for All Routes

Changing external AD for all routes fixes IDRI in the 2) and 4) cases (not for 6) nor 7)) on all redistributing routers between the last 2 domains, so that :

"middle" RP external AD < "edge" RP external AD

This means in our situation:

"middle" RP external AD < OSPF external AD

We cannot apply the previous tool for all routes in the last 2 cases (6) and 7)) because we also have to make sure in both "directions" between the last 2 domains that:

"middle" RP internal AD < "edge" RP external AD

So, in cases 6) and 7) (RIP as edge RP), a filtering tool must be used instead, as described in "Setting Per-Route AD Values"; the "Filtering routes during redistribution" and "Using route tags" solutions are incomplete : the long route stays on the redistributing routers.

- *The only complete solutions to solve these IDRI require AD tweaking, either using all routes AD settings or using a per-route AD settings for the routes coming from the first routing domain (more complicated solution), or filtering the reception of tagged routes updates.*

5.3.2.2 Setting Per-Route AD Values

Complete solution.

5.3.2.3 Filtering Routes During Redistribution

Incomplete solution: the long route stays on the redistributing routers.

<https://t.me/learningnets>

5.3.2.4 Filtering Tagged Routes During Redistribution

Incomplete solution: the long route stays on the redistributing routers.

5.3.2.5 Filtering Tagged Routing Updates During Reception

Complete solution.

5.4 Load-Balancing Summary

5.4.1 With Routing Protocols

Some tools to implement load-sharing are available for:

5.4.1.1 EIGRP

- [equal-cost load-balancing](#),
- [unequal-cost load-sharing](#),
- [metric tuning](#).

5.4.1.2 OSPF

- [equal-cost load-balancing and metric tuning](#),
- *unequal-cost load-sharing is NOT possible.*

5.4.1.3 BGP

- [equal-path load-balancing](#)

5.4.2 With First Hop Redundancy Protocols

All the details are in my [CCNP SWITCH Summary](#).

Here's an interesting article about [FHRP in IPv6 networks on ipSpace](#).

5.4.2.1 HSRP/VRRP

HSRP (Hot Standby Redundancy Protocol) and VRRP (Virtual Router Redundancy Protocol): multiple groups/VLAN with a vIP/group where the Active/Master roles are distributed among different routers.

5.4.2.2 GLBP

GLBP (Gateway Load-Balancing Protocol): designed with that purpose (along with high-availability) in mind.

5.4.3 Caveats

There are some tricks to be aware of depending on the topology (from [PERFECT LOAD-BALANCING: HOW CLOSE CAN YOU GET?, NIL](#)).

If you need a refresh regarding all different forwarding architectures, you can check out my [CCNP SWITCH Summary](#).

5.4.3.1 Fast Switching

Load-balancing is performed based on destination prefix only (unless you can afford configuring CPU-intensive process switching with the “no ip route-cache” configuration command, where the load-balancing is done on a packet-by-packet basis). If there's only one destination, destination-only load-balancing is obviously useless.

5.4.3.2 CEF

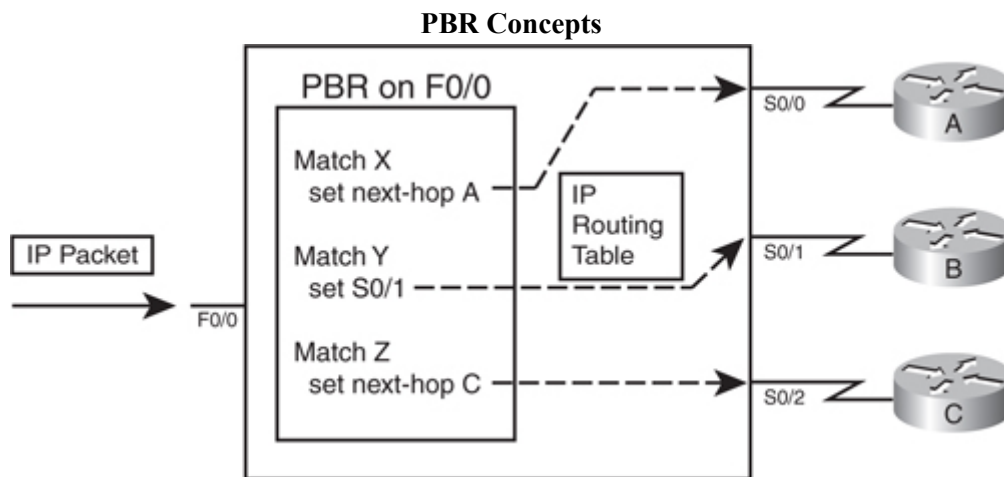
By default, CEF (enabled with “ip cef” global configuration command) performs source-destination pair-based load-balancing. Source and destination IP addresses are hashed together in a 4-bit value that is used to select one of the potential outgoing paths. This guarantees that all the traffic between a pair of IP hosts will flow over the same path. Layer 4 information (source/destination UDP/TCP ports) can also be included in the hashing and the decision process.

5.5 PBR

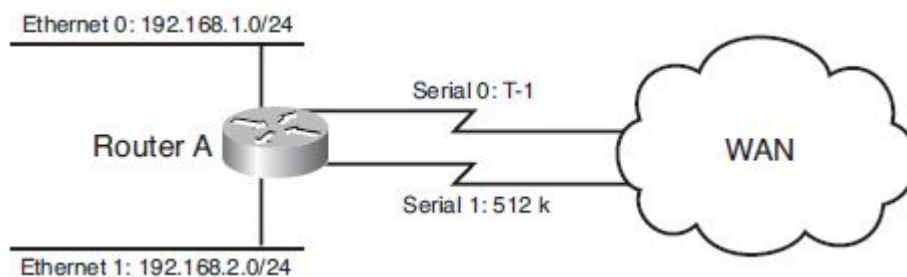
You can use PBR to modify the next-hop address of packets or to mark packets to receive differential service.

Policy-Based Routing defines static routes for some packets received on some particular interface(s) or originated from the local router, and matching some filter(s) applied on their IP header, corresponding to:

- ✓ source and destination IP addresses,
 - ✓ IP protocols,
 - ✓ services (source and destination ports).
- These static routes do not appear in the IP routing table,
 - By default, PBR is CEF-switched (as long as CEF is enabled),
 - If PBR is unable to route a particular packet, then the normal routing process takes over with the routing table.

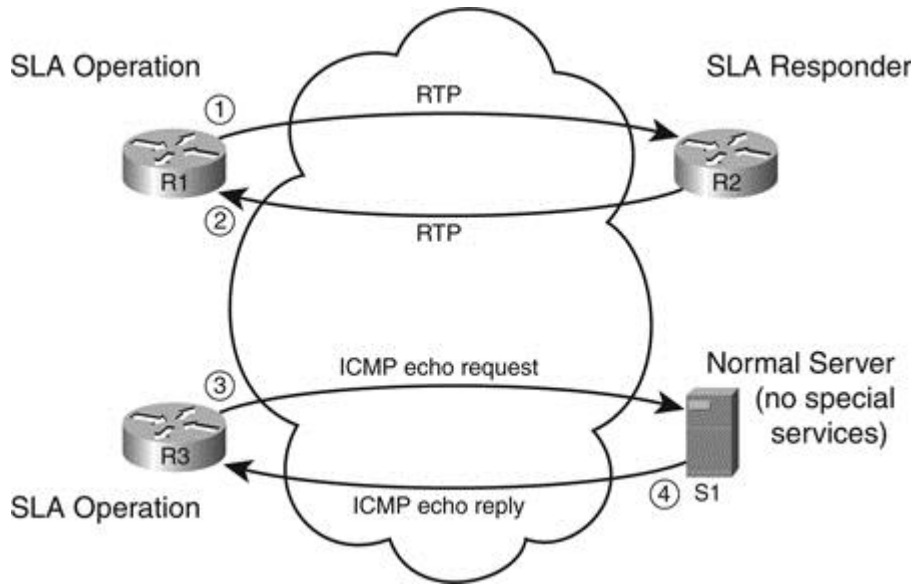


Example: in the following figure, the routing protocol might select Serial 0 as the preferred path for all traffic because of the higher bandwidth. The company might have business-critical systems that use the T1 but does not want systems on Ethernet 1 to affect WAN performance. You can configure PBR on Router A to force traffic from Ethernet 1 out on Serial 1.



5.6 SLA

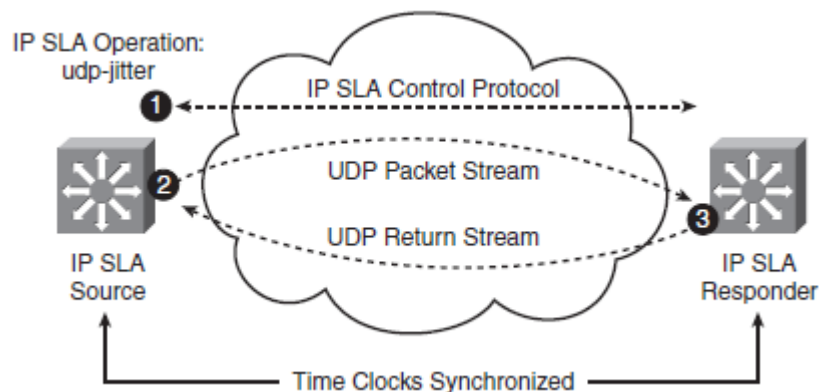
The IOS *IP Service-Level Agreement* feature measures the ongoing behavior of the network:



The following list summarizes the majority of the available operation types, just for perspective:

Operation Type	Description	SLA Responder on target?
ICMP echo	End-to-end response times	NO
ICMP jitter	End-to-end jitter	YES
Path echo	Hop-by-hop and end-to-end response times over path discovered from ICMP Echo	NO
Path jitter	Hop-by-hop jitter over ICMP Echo path	YES
UDP echo	End-to-end response time of UDP echo	NO
UDP jitter	Round trip delay, one-way delay, one-way jitter, one way packet loss, and connectivity using UDP packets Can be used to test VOIP quality with a specific codec	YES
TCP connection	Response time to build a TCP connection with a host	NO
DNS	DNS query response time	NO
DHCP	DHCP IP address request response time	NO
HTTP	Web page retrieval response time	NO
FTP	FTP file retrieval response time	NO

IP SLA UDP Jitter Test Operation



By leveraging IP SLA test operations, you can take advantage of some fancy features:

- Generate SNMP traps/syslog messages when certain test thresholds are exceeded
- Schedule further IP SLA tests automatically when test thresholds are crossed
- Track an IP SLA test to trigger a next-hop gateway redundancy protocol, such as HSRP
- Gather voice quality measurements from all over a network

<https://t.me/learningnets>

5.7 Configuration Commands

5.7.1 Redistribution into EIGRP

(config-router)#**redistribute** *source-protocol* [*process-id* | *as-number*] [**metric** *bandwidth* *delay* *reliability* *load* *mtu*] [**match** {**internal** | **external 1** | **external 2** | **nssa-external**}] [**tag** *tag-value*] [**route-map** *name*]

<i>kbits/s</i>	<i>*10μs</i>	<i>100%</i> <i>0-255</i>	<i>100%</i> <i>1-255</i>	<i>1-65535</i>
----------------	--------------	-----------------------------	-----------------------------	----------------

Parameters of the EIGRP *redistribute* Command

Option	Description
protocol	The source of routing information. Includes RIP, OSPF, EIGRP, IS-IS, BGP, connected, and static.
process-id, as-number	If redistributing a routing protocol that uses a process ID or ASN on the router global config command, use this parameter to refer to that process or ASN value.
metric	A keyword after which follows the four metric components (bandwidth, delay, reliability, link load), plus the MTU associated with the route.
match	If redistributing from OSPF, this keyword lets you match internal OSPF routes, external (by type), and NSSA external routes, essentially filtering which routes are redistributed.
tag	Assigns a unitless integer value to the routes redistributed by this command—tags which can be later matched by other routers using a route-map.
route-map	Apply the logic in the referenced route-map to filter routes, set metrics, and set route tags.

Necessary from
other IGPs →

When the **redistribute** command refers to another IGP as the routing source, it tells the router to redistribute the following:

- All routes in the routing table learned by that routing protocol
- All connected routes of interfaces on which that routing protocol is enabled

Methods of Setting EIGRP Metrics When Redistributing into EIGRP

Function	Command	Priority
Setting the default for all redistribute commands	The default-metric <i>bw delay reliability load mtu</i> EIGRP subcommand	3
Setting the component metrics applied to all routes redistributed by a single redistribute command	The metric <i>bw delay reliability load mtu</i> parameters on the redistribute command	2
Setting different component metrics to different routes from a single route source	Use the route-map parameter on the redistribute command, matching routes and setting metric components.	1

5.7.2 Redistribution into OSPF

```
(config-router)#redistribute source-protocol [process-id | as-number] [metric metric-value] [metric-type type-value]
[match {internal | external 1 | external 2 | nssa-external}] [tag tag-value] [route-map name] [subnets]
```

Parameters on the OSPF *redistribute* Command

Option	Description
protocol	The source of routing information. Includes RIP, OSPF, EIGRP, IS-IS, BGP, connected , and static .
process-id, as-number	If redistributing a routing protocol that uses a process-id or ASN on the router global config command, use this parameter to refer to that process or ASN value.
metric	Defines the cost metric assigned to routes redistributed by this command, unless overridden by a referenced route map.
metric-type {1 2}	Defines the external metric type for the routes redistributed by this command: 1 (E1 routes) or 2 (E2 routes), <i>or N1/N2</i>
match	If redistributing from another OSPF process, this keyword lets you match internal OSPF routes, external (by type), and NSSA external routes, essentially filtering which routes are redistributed.
tag	Assigns a unitless integer value to the routes redistributed by this command—a tag that can be later matched by other routers using a route-map.
route-map	Apply the logic in the referenced route-map to filter routes, set metrics, and set route tags.
subnets	Redistribute subnets of classful networks. Without this parameter, only routes for classful networks are redistributed. (This behavior is unique to the OSPF redistribute command.)

Summary of Metric Values When Redistributing into OSPF

Function	Command or Metric Values	Priority
Default if no metric configuration exists	Cost 1 for routes learned from BGP. If redistributed from another OSPF process, use the source route's OSPF cost. Cost 20 for all other route sources.	
Setting the default for all redistribute commands	The default-metric cost OSPF subcommand.	3
Setting the metric for one route source	The metric cost parameters on the redistribute command.	2
Setting different metrics for routes learned from a single source	Use the route-map parameter on the redistribute command.	1

5.7.3 Redistribution Default Seed Metrics

For all redistributed routes, and all routing protocols, *except IS-IS*:

(config-router)#**default-metric** *value*

- EIGRP *value: bandwidth delay reliability load mtu*
 - bandwidth*: from 1 to $2^{32}-1$, in *kbits/s*
 - delay*: in tens of μ s
 - reliability*: from 0 through 255. The value 255 means 100 percent reliability; 0 means no reliability.
 - load*: from 1 to 255. The value 255 means 100 percent load. 1 is minimum load.
 - mtu*: from 1 to 65535.
- RIPv2, OSPF and BGP: from 1 to $2^{32}-1$
- A metric value of 0 is applied to redistributed *connected* routes.
- The **default-metric** command *does not override* metric values that are applied with the **redistribute** command.

5.7.4 Inter-Domain Routing with Multiple Redistribution Points

5.7.4.1 Between 2 domains

5.7.4.1.1 Setting AD Values for all routes

Routing Protocol	Command
RIP	distance <i>ad-value</i>
EIGRP	distance <i>eigrp internal-ad external-ad</i>
OSPF	distance <i>ospf {external ad-value} {intra-area ad-value} {inter-area ad-value}</i>
BGP	distance <i>bgp external-ad internal-ad local-ad</i>

5.7.4.2 Between 3 domains

5.7.4.2.1 Setting AD Values for all routes

Cf. [§Setting AD Values for all routes](#)

5.7.4.2.2 Setting Per-Route AD Values

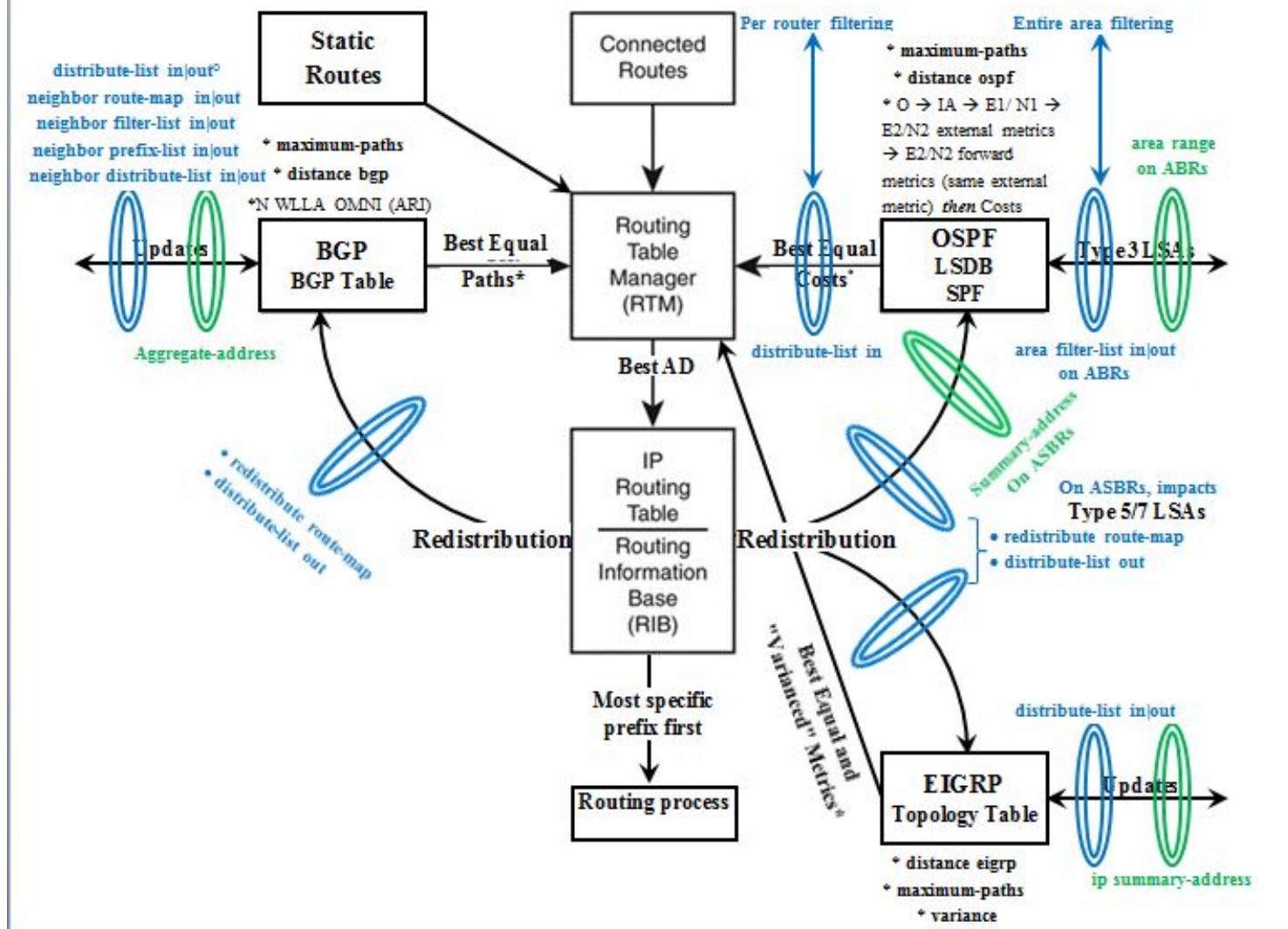
(config-router)#**distance** *distance advertising-router wildcard-mask [acl-number-on-prefix-len | acl-name-on-prefix-len]*

Defines an administrative distance *for ACL-matched routes that are inserted into the routing table when advertised from wildcard-matched neighboring routers*.

- The required parameters match the neighboring router(s) that advertise(s) some route(s). The router with the **distance** command configured compares the advertising router's IP address (or **OSPF RID of the OSPF router that owns the LSA**) to the range of addresses implied by the *advertising-router* and *wildcard-mask* parameters of the command, as if these were parameters in an ACL. For routes advertised by a matching neighbor, that router then applies the AD listed in the command.
- Optionally, the **distance** command can also refer to a standard or extended ACL. If included, that router compares the ACL to the *prefix/length of each route* learned from any matched neighbors and uses the listed AD only for routes permitted by the ACL.
- There can be multiple distance commands, which are all executed. So always set the AD from the least to the most specific network.*

5.7.5 RTM

Routing Table Manager Concept



° BGP distribute-list configuration will be removed from CLI at a future date (BGP Command Reference, March 2011), because prefix lists provide improved performance and are simpler to configure

In case two routing protocols with the same AD (2 EIGRP or 2 OSPF processes on the same router) injects same routes to RTM:

- EIGRP installs the route with the lowest metric in the RIB; in case of a tie (same metric), the route coming from the lowest AS number is installed in the RIB;
- OSPF prefers to install in the RIB its oldest route whatever its cost is (strange choice).

5.7.6 Default Administrative Distances

Route Type	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5 → Local (Advertised with AD 90 and redistributed with AD 170)
eBGP	20
EIGRP (internal)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
On-Demand Routing (ODR)	160
EIGRP (external)	170
iBGP	200
Unreachable	255

5.7.7 PBR

(config-if)#**ip policy route-map** *name* Enables PBR for packets entering that interface.

- PBR processes packets that match a permit clause using the defined **set** command.
- For packets matched by a deny clause, PBR lets the packet go through to the normal IP routing process.

(config)#**ip local policy route-map** *name* Makes IOS process locally created packets using PBR logic.

(config)#**route-map** *map-name* [**permit** | **deny**] [*sequence-number*]

(config-route-map)#**match ip address** {*access-list-number* [*access-list-number* ... | *access-list-name* ...] | *access-list-name*}

- *standard ACL : matches source IP address of packet;*
- *extended ACL : matches protocol, source and destination IP addresses and ports of the packet*

(config-route-map)#**match length** min max

Choosing Routes Using the PBR *set* Command

Command	Comments
set ip next-hop <i>ip-address</i> [...] <i>ip-address</i>]	Next-hop addresses must be in a connected subnet; forwards to the first address in the list for which the associated interface is up.
set ip default next-hop <i>ip-address</i> [...] <i>ip-address</i>]	Same logic as previous command, except policy routing first attempts to route based on the routing table.
set interface <i>interface-type</i> <i>interface-number</i> [...] <i>interface-type</i> <i>interface-number</i>]	Forwards packets using the first interface in the list that is up.
set default interface <i>interface-type</i> <i>interface-number</i> [...] <i>interface-type</i> <i>interface-number</i>]	Same logic as previous command, except policy routing first attempts to route based on the routing table.

⇒ *Virtually replaces IP routing table's default route*

⇒ *Virtually replaces IP routing table's default route*

The set clauses can be used in conjunction with one another. They are evaluated in the following order:

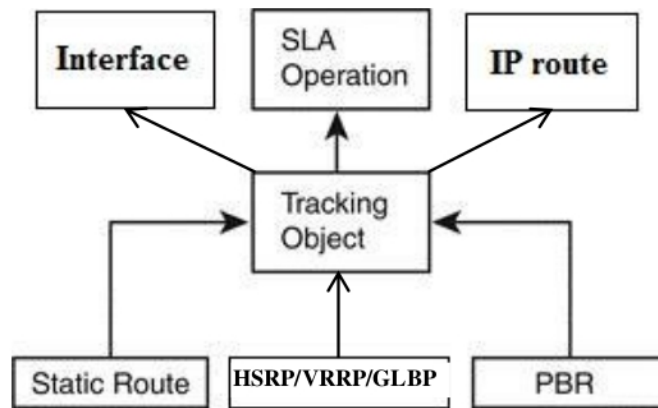
- 1) *set ip next-hop*
- 2) *set interface* (not recommended at all on multipoint interfaces, cf. [§Static Default Route](#))
- 3) *set ip default next-hop*
- 4) *set default interface* (not recommended at all on multipoint interfaces, cf. [§Static Default Route](#))

5.7.8 SLA

5.7.8.1 SLA Operation

- **Step 0:** Enable the IP SLA *responder* on the target device **if necessary**: **jitter** operations.
(config)#**ip sla responder** [{**udp-echo** | **tcp-connect**} **ipaddress** *ip-address* **port** *port-number*]
- **Step 1:** Create the IP SLA Operation, and assign it an integer operation number:
(config)#**ip sla** *sla-ops-number*
- **Step 2:** Define the operation type and the parameters for that operation type.
For ICMP echo, you define the destination IP address or hostname, and optionally, the source IP address or hostname:
(config-ip-sla)#**icmp-echo** *destination-ip-address* [**source-ip** {*ip-address* | **source-interface** *interface-name*}]
Or
(config-ip-sla)#**udp-jitter** *destination-ip-address* *destination-port-number* [**source-ip** *ip-address*]
[**source-port** *port-number*]
There are also icmp-jitter, path-echo, path-jitter, udp-echo, tcp-connect, voip delay, voip rtp, dns, dhcp, http get, ftp get... commands.
- **Step 3:** (Optional) Define a frequency that the operation should send the packets, in seconds (60s by default):
(config-ip-sla-echo)#**frequency** *seconds*
(Optional) Define a Length of time the operation waits to receive a response from its request packet, in ms (5000 ms by default for **icmp-echo**).
(config-ip-sla-echo)#**timeout** *milliseconds*
(Optional) Define a length of time required for a rising threshold to be declared, in ms. (5000 ms by default).
(config-ip-sla-echo)#**threshold** *milliseconds*
➤ **For all IP SLAs operations (except UDP jitter), the following configuration guideline is recommended:**
(threshold milliseconds) < (timeout milliseconds) < (frequency seconds)
- **Step 4:** Schedule when the SLA will run:
(config)#**ip sla schedule** *sla-ops-number* [**start-time** {*hh:mm[:ss]* [*month day* | *day month*]} | **pending** | **now** | **after** *hh:mm:ss*}] [**life** {**forever** | *seconds*}] [**ageout** *seconds*] [**recurring**]
 - Name of the *month*, number of the *day*
 - **pending**: no information is collected (default)
 - Default **life** *seconds* is 1 hour (3600s)
 - **ageout**: waiting period when not actively collecting information before it dies (default 0: never ages out)
 - **recurring**: start automatically at the specified time and for the specified duration every day.

5.7.8.2 Tracking Object



5.7.8.2.1 Tracking IP SLA

- **Step 1:** Track the state of an SLA operation:

(config)#**track** object-number ip sla sla-ops-number [reachability | state]

- *object-number* The range is from 1 to 1000.
- **reachability:** tracks the operation return code: **OK when the destination Protocol/Service is reachable and when the RTT/jitter is below or over Threshold.**
- **state:** tracks the operation return code: **OK when the destination Protocol/Service is reachable and when the RTT/jitter is below Threshold.**

Track keyword	SLA Operation Return Code	Tracking Object State
• reachability	OK (reachable and below or over Threshold)	UP
	All other codes	DOWN
• State	OK (reachable and below threshold)	UP
	All other codes	DOWN

5.7.8.2.2 Tracking an Interface

- **Step 1:** Track the state of an interface:

(config)#**track** object-number interface type number {line-protocol | ip routing}

- **line-protocol:** Tracks the state of the interface line protocol
- **ip routing:** Tracks whether IP routing is enabled, whether an IP address is configured on the interface, and whether the interface state is up before reporting to the tracking client that the interface is up.

5.7.8.2.3 Tracking an IP Route

A tracked IP route object is considered UP and reachable when a routing table entry exists for that route and the route is not inaccessible.

- **Step 1:** Track the state of an IP route:

(config)#**track** object-number ip route subnet-number subnet-mask { reachability | metric threshold }

- **reachability:** Tracks whether the route is reachable.
- **metric threshold:** Tracks the threshold metric. The default up threshold is 254 and the default down threshold is 255.

- **Step 1 bis:** Specify the UP / DOWN metric thresholds

(config-track)#**threshold metric** {up up-to-that-metric [down equal-or-above-that-metric]}

- **up:** Specifies the up threshold. The state is up if the scaled metric for that route is less than or equal to the *up-to-that-metric* threshold. The default up threshold is 254.
- **down:** Specifies the down threshold. The state is down if the scaled metric for that route is greater than or equal to the *equal-or-above-that-metric* threshold. The default down threshold is 255.

5.7.8.2.4 Regulating Flapping of the Tracking Object State

- **Step 2:** (Optional) Configure the delay to regulate flapping of the tracking object state by:


```
(config-track)#delay {down waiting-seconds | up waiting-seconds}
```

 - The **down** timer defines how long to wait, when in an UP state, after seeing the operation return code move to a non OK state, before moving the tracking object to a DOWN state. Default is 0.
 - The **up** timer on the tracking object defines how long to wait, when in a DOWN state, after seeing the operation return code transition to an OK state, before moving the tracking object to an UP state. Default is 0.

5.7.8.2.5 Static Route to Track an IP SLA Operation / IP Route / Interface

- **Step 3:** Configure the static route with:


```
(config)#ip route destination mask {interface | next-hop} track object-number
```
- If the operation fails (Tracking object state DOWN), the static route is removed from the routing table and replaced by a dynamic one if available.

5.7.8.2.6 PBR to Track an IP SLA Operation / IP Route / Interface

- **Step 3:** Configure the PBR with:


```
(config-route-map)#set ip next-hop [verify-availability] next-hop-address track object-number
```

Configures PBR to verify the reachability/*state* of the **Protocol/Service of a router/host** before the router performs PBR to that next hop:

- **verify-availability:** verifies next hop reachability using Cisco Discovery Protocol (CDP) before the router tries to policy route to it if **track** is *not specified*:
 - It causes some performance degradation.
 - CDP must be configured on the interface.
 - Some media or encapsulations do not support CDP.
 - The next hop must be a Cisco device with CDP enabled.
 - If you want to selectively verify availability of only some next hops, you can configure different route map entries (under the same route map name) with different criteria (using access list matching or packet size matching), and then use the **set ip next-hop verify-availability** command selectively
 - **track:** allows PBR to track an operation to make decisions based on the following criteria:
 - ICMP ping reachability to a remote device,
 - Service running on a remote device (for example, the device responds to an HTTP GET request),
 - Availability of an IP route or interface,
 - ...
 - *next-hop-address* : IP address of the next hop to which packets will be forwarded
 - *object-number* : Object number that the tracking subsystem is tracking
- When the tracking object is UP, PBR works as configured. When the tracking object is DOWN, PBR acts as if the **set** command does not exist. That means that the router will still attempt to route the packet per the normal destination-based routing process.

5.7.8.2.7 FHRP to Track an IP SLA Operation / IP Route / Interface

FHRP (First hop Redundancy Protocol – HSRP/VRRP/GLBP) can also track the state of a tracking object. Cf. the [CCNP SWITCH Summary](#) for details.

5.8 Verification Commands

#show ip route [*prefix mask*]

- EIGRP External: D EX
- OSPF External Type 1 and 2: O E1 / E2
- OSPF NSSA External Type 1 and 2: O N1 / N2
- Administrative Distance – Costs
- With *prefix mask* : “*forward metric*” shows the cost from the local router to the ASBR for the E2/N2 redistributed subnets.

#traceroute *IP Address* Verifies the results of the policy routing

#show ip policy Shows the interfaces on which PBR is enabled and the route map used.

#show ip sla configuration [*sla-ops-number*] Confirms all the configuration settings for the operation

#show ip sla statistics [**aggregated**] [*sla-ops-number*]

Lists the current statistics for the operation:

- Number of passed intervals
- Round trip time (RTT)
- Return code of the most recent operation, a key value used by SLA tracking

#show ip sla application Displays global information about SLAs

#show ip sla responder Displays global information about SLAs responder

#show track Lists:

- the tracking object’s configuration,
- the current tracking state,
- the time in this state,
- the number of state transitions,
- the entities that track the object (STATIC-IP-ROUTING, ROUTE-MAP).

#show processes cpu sorted time-frame

CPU utilization per process

#show processes cpu history

Overall CPU utilization graph

#debug ip policy [*acl*]

The only way to verify the types of packets that are policy routed, which can be dangerous on production routers, given its multiple lines of output per packet.

#debug ip sla trace

Outputs IP SLA trace messages

#debug ip routing

To dynamically view IP routing table changes (used with IP SLA for example).

6 BGP

BGP-4 has been standardized in [RFC 4271](#) and **MP-BGP** (Multiprotocol Extensions for BGP-4) in [RFC 4760](#).

6.1 Topological Designs

6.1.1 Single-Homed designs

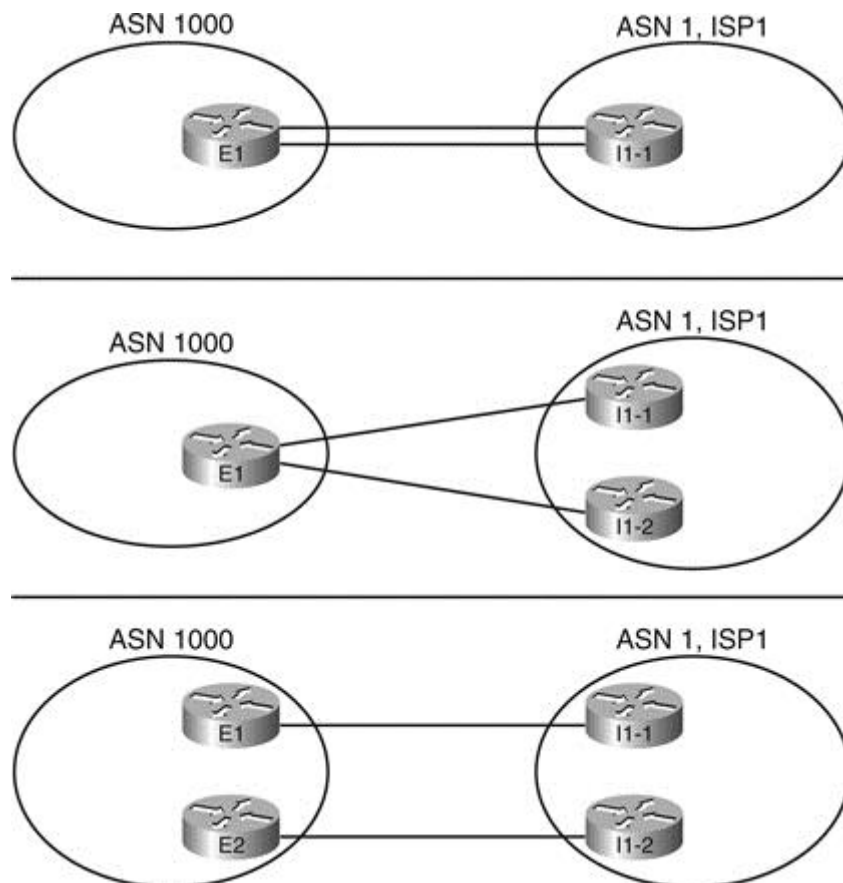
- 1 link per ISP, 1 ISP

They often use one of two options for routing to and from the Internet:

- Use static routes (default in the Enterprise, and a static for the Enterprise's public address range at the ISP).
- Use BGP, but only to exchange a default (ISP to Enterprise) and a route for the Enterprise's public prefix (Enterprise to ISP).
 - The ISP router uses BGP to advertise a default route to the Enterprise.
 - You must configure BGP on the Enterprise router and advertise the Enterprise's public prefix toward the ISP.
 - You must configure the IGP on the Enterprise's Internet-facing router to flood a default route (typically only if the default route exists in that router's routing table).

6.1.2 Dual-Homed Designs

- 2+ links per ISP, 1 ISP

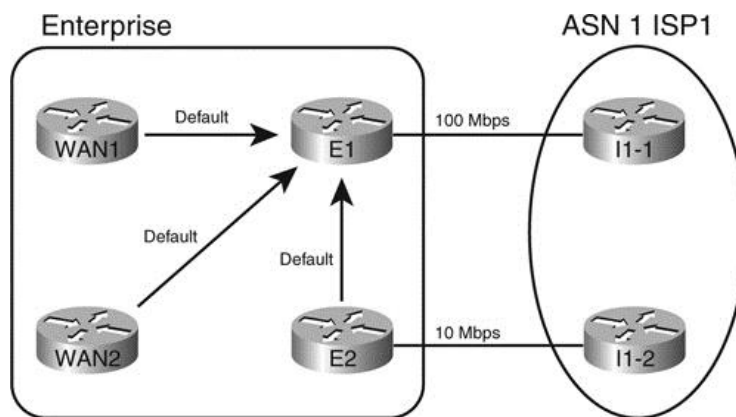


The common cases when using defaults works well, ignoring BGP, are:

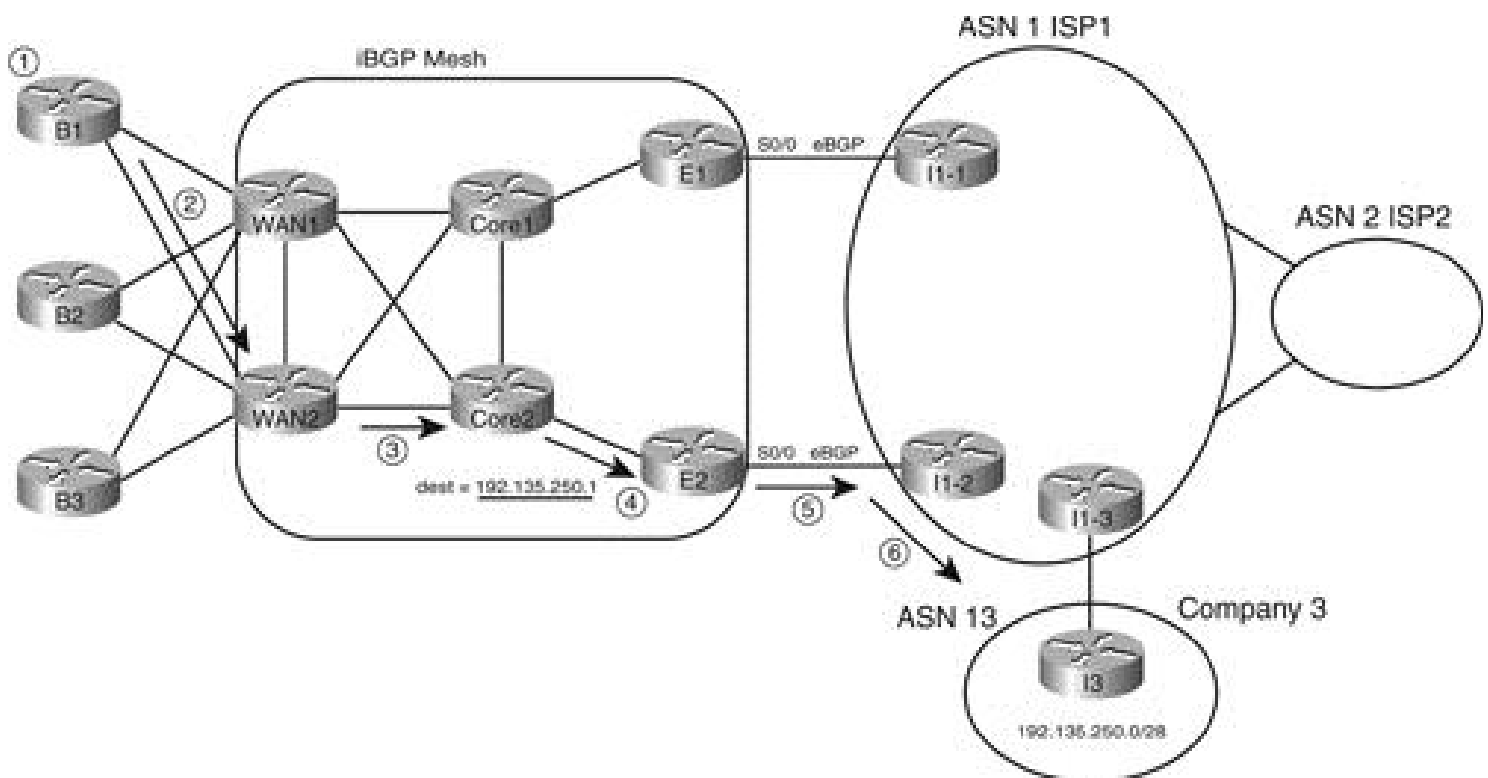
- **Active + Backup links:** preferring one Internet connection over another for all destinations, but when the better ISP connection fails, all traffic re-routes over the secondary connection.
- **2 active links:** treating both Internet connections as equal (*load-balancing*), sending packets for some destinations out each path. However, when one fails, all traffic re-routes over the one still-working path.

6.1.2.1 Choosing One Path over Another for All Destinations Using Default Routes

Dual-Homed Design, Using Defaults to Favor One Link



6.1.2.2 Choosing One Path over Another for Some Destinations Using iBGP

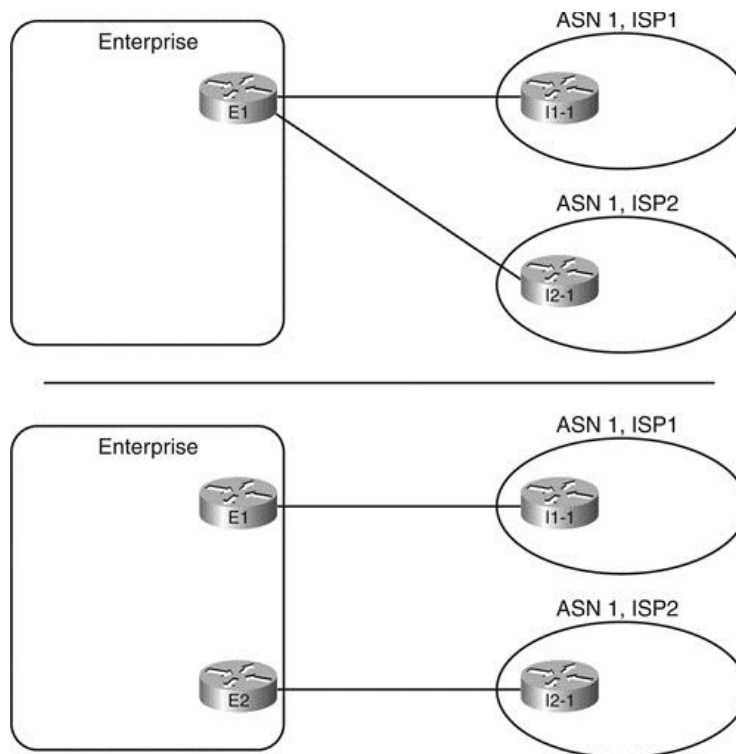


ISPs give you three basic options for what routes the ISP advertises:

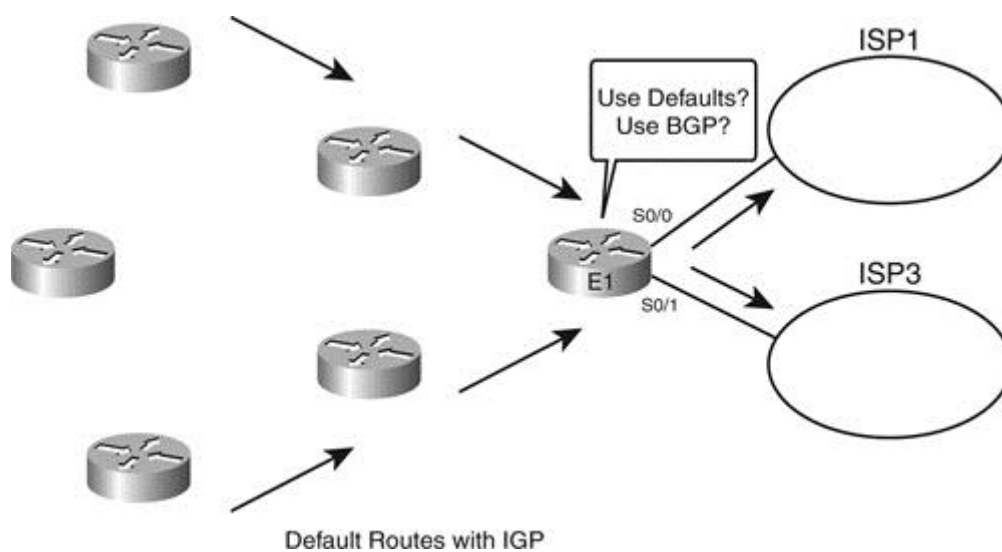
- **Default route only:** The ISP advertises a default route with BGP, but no other routes.
- **Full updates:** The ISP sends you the entire BGP table.
- **Partial updates:** The ISP sends you routes for prefixes that might be better reached through that ISP, but not all routes, plus a default route (to use instead of the purposefully omitted routes as needed).

6.1.3 Single Multi-Homed Designs

- 1 link per ISP, 2+ ISPs

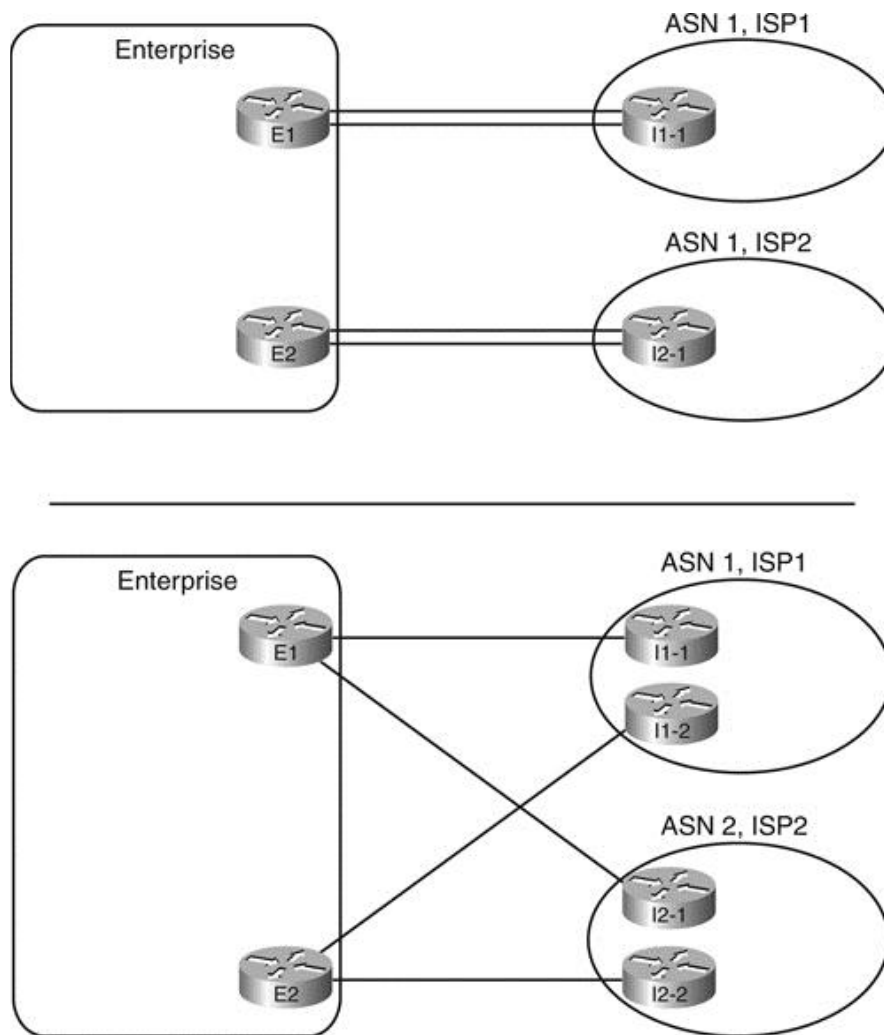


Outbound Routing with a Single multi-homed Design



6.1.4 Dual Multi-homed Designs

- 2+ links per ISP, 2+ ISPs



6.2 Reserved Values

6.2.1 Private IPv4 Address Blocks

- Not Allocated In Public Internet ([RFC 1918](#))

Number of Classful Networks	Range of Classful Networks	Prefix for Entire Range
(1) Class A:	10.0.0.0	10.0.0.0/8
(16) Class B:	172.16.0.0 through 172.31.0.0	172.16.0.0/12
(256) Class C:	192.168.0.0 through 192.168.255.0	192.168.0.0/16

6.2.2 Reserved IPv4 Address Registries

- Not Allocated In Public Internet ([RFC 6890](#))

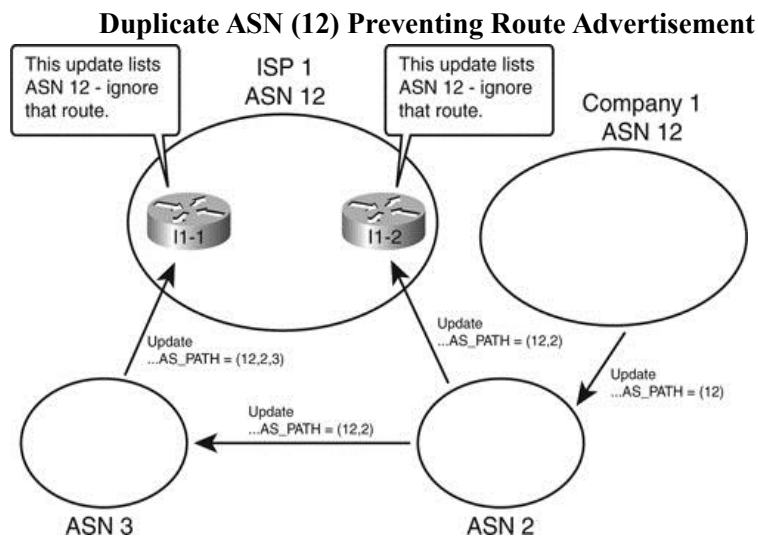
UPnP
(Auto-IP)



Value or Range	Reason
0.0.0.0/8	Used for self-identification on a local subnet.
127.0.0.0/8	Loopback testing.
169.254.0.0/16	This "link local" block is used for default IPv4 address assignment when DHCP process fails.
192.0.2.0/24	Reserved for use in documentation and example code.
192.88.99.0/24	Used for IPv6 to IPv4 relay (6to4 relay) (RFC 3068).
198.18.0.0/15	Benchmark testing for Internet devices (RFC 2544).

6.2.3 16-Bit ASN Assignment Categories

- From [IANA](#)



Value or Range	Purpose
0	Reserved
1 through 64,495	Assignable by IANA for public use
64,496 through 64,511	Reserved for use in documentation
64,512 through 65,534	Private use
65,535	Reserved

- [RFC 6793](#) introduced 32-bit AS numbers, which IANA has begun to allocate. These numbers are written either as simple integers, or in the form x.y, where x and y are 16-bit numbers. Numbers of the form 0.y are exactly the old 16-bit AS numbers, 1.y numbers and 65535.65535 are reserved, and the remainder of the space is available for allocation (from [Wikipedia](#)).

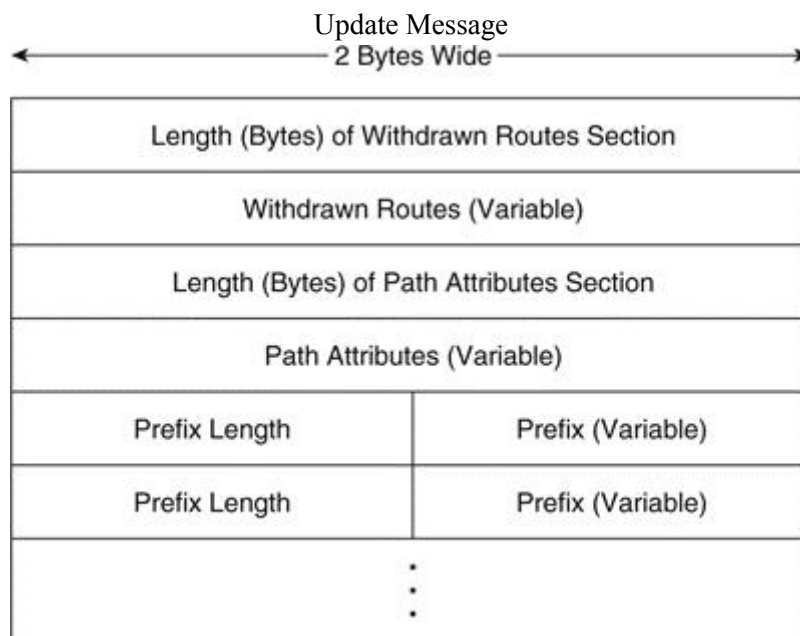
6.3 Establishing Peering

Peering is another word for *Neighborhood*.

6.3.1 Message Types Summary

Message	Purpose	Similarity with EIGRP
Open	Used to establish a neighbor relationship and exchange basic parameters, including ASN and MD5 authentication values.	Hello
Keepalive	Sent on a periodic basis to maintain the neighbor relationship. The lack of receipt of a Keepalive message within the negotiated Hold timer causes BGP to bring down the neighbor connection.	Hello
Update	Used to exchange PAs and the associated prefix/length (NLRI) that use those attributes.	Update
Notification	Used to signal a BGP error; typically results in a reset to the neighbor relationship.	No direct equivalent

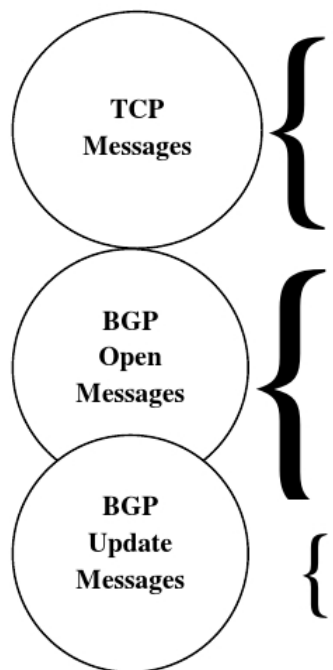
Route Refresh: Allows BGP peers to synchronize without a hard reset by re-advertising the respective outbound routing tables



6.3.2 Peer States

Although the Finite State Machine (FSM) for BGP neighbor states has many twists and turns, particularly for handling exceptions, retries, and failures, the overall process works as follows:

- **Step 1:** A router tries to establish a TCP connection with the IP address listed on a **neighbor** command, using **well-known destination port 179 and a random source port**.
- **Step 2:** When the three-way TCP connection completes, the router sends its first BGP message, the BGP **Open** message, which generally performs the same function as the EIGRP and OSPF Hello messages. The Open message contains several **BGP parameters**, including those that must be **verified** before allowing the routers to become neighbors. The information exchanged before the neighbors are established includes:
 - **BGP version number,**
 - **ASN,**
 - **BGP router ID,**
 - **BGP capabilities.**
- **Step 3:** After an Open message has been sent and received and the neighbor parameters match, the neighbor relationship is formed, and the neighbors reach **established** state.

		State	Typical Reasons
	TCP Messages	Idle	The BGP process is either administratively down or awaiting the next retry attempt. <i>(admin)</i>
		Connect(<i>ing</i>)	The BGP process is waiting for the TCP connection to be completed. You cannot determine from this state information whether the TCP connection can complete.
		Active	The TCP connection has been completed, but no BGP messages have been sent to the peer yet.
	BGP Open Messages	Opensent	The TCP connection exists, and a BGP Open message has been sent to the peer, but the matching Open message has not yet been received from the other router.
		Openconfirm(<i>ing</i>)	An Open message has been both sent to and received from the other router. The next step is to receive a BGP Keepalive message (to confirm all neighbor-related parameters matched) or BGP Notification message (to learn there is some mismatch in neighbor parameters).
	BGP Update Messages	Established	All neighbor parameters match, the neighbor relationship works, and the peers can now exchange Update messages.

6.3.3 Peering Requirements

- Each router must be part of a TCP connection with the other router, with:
 - the remote router's *IP address* used in that TCP connection matching the **Update Source** local router configuration in that same TCP connection,
 - the **eBGP multihop** configured if necessary (multi-links with the use of loopback interfaces).
- A **local router's** ASN must match the **neighboring router's reference to that ASN**.
- The **BGP router IDs** of the two routers must not be the same.
- If configured, **MD5 authentication** must pass.

6.3.4 eBGP

eBGP peers must share a common subnet, and Cisco does allow some flexibility to avoid doing so.

6.3.5 iBGP

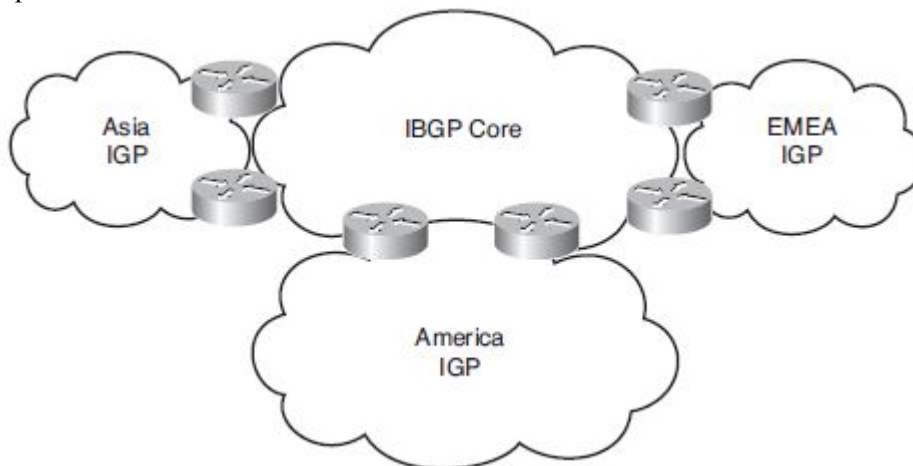
6.3.5.1 In Transit AS

iBGP is used primarily *in transit ASes*. If transit ASes did not use iBGP, the eBGP-learned routes would have to be redistributed into an IGP and then redistributed into the BGP process in another eBGP router with 2 major downsides:

- Normally, the number of eBGP routes is too large for an IGP to handle.
- Most of the BGP path attributes would be lost during the transit.

6.3.5.2 In Large Corporations

Another use of iBGP is *in large corporations* where the IGP networks are in smaller independent routing domains along organizational or geographic boundaries:



6.3.5.3 Policing Internal AS

Another use of iBGP is applying policies in the internal AS with the help of BGP path attributes.

6.3.5.4 QoS Policy Propagation on BGP

QPPB uses iBGP to spread common QoS parameters from one router to other routers in the network. It classifies packets using IP DSCP bits based on BGP community lists, BGP AS paths, and ACLs. After packets are classified, QoS features can enforce policies.

6.3.5.5 Carrying MPLS VPN Information

The multiprotocol version of BGP is used to carry MPLS VPN information between all Provider Edge (PE) routers within a VPN community.

MP-BGP introduces a new BGP capabilities advertisement to determine whether a BGP peer supports MP-BGP.

6.4 Exchanging the BGP Table

6.4.1 Seeding the BGP Table

- BGP **network** command.

When a router **knows routes only for subsets of the prefix** that needs to be advertised, an **additional step** is needed when using the **network** command. So, either :

- use a *static (discard) route for the entire range, with outgoing interface null0*,
- or use *IGP route summarization* to create a summary route for the entire prefix.
- or use *auto-summarization with a classful network* when using the **network** command.

- **Redistribution** from an IGP / Connected / Static.

However, to reach the goal of **advertising only a single route for an entire public prefix**, another step must be taken, typically one of the following:

- use a *static (discard) route for the entire range, with outgoing interface null0*,
- or use *IGP route summarization* to create a summary route for the entire prefix.
- or use *BGP route summarization* to make BGP advertise only the entire prefix.

6.4.2 Exchange Process: Advertising Rules

- **Only advertise the best route(s)** in any BGP Update (*up to the maximum number of equal paths – 1 by default*).
 - A router will **not advertise the best route back to the same neighbor** that advertised that route.
 - When advertising to an **eBGP peer**, a BGP router **updates the AS_Path PA with its own AS number**, but it does not do so when advertising to an iBGP peer.
 - **Do not advertise iBGP-learned routes to iBGP peers.**
 - *By default*, when a router advertises a route using **eBGP**, the advertising router lists its **own update-source IP address as the next-hop address of the route**. In other words, the next-hop IP address is the IP address of the eBGP neighbor.
- However, when advertising a route to an **iBGP peer**, the advertising router (*by default*) **does not change the next-hop**.

➤ A few options exist to **ensure reachability to these next-hop addresses**:

- Create **static routes** so that each router can reach **these next-hop addresses that exist in other ASNs**,
- Use an **IGP between both ASes** (the Enterprise and the ISPs for example) for the sole purpose of advertising these next-hop addresses,
- **Redistribute the Connected subnet into the IGP (if the eBGP neighbor's update-source is in that subnet)**,
- **Advertise the Connected subnet into BGP by network command or redistribution**,
- Change the default iBGP behavior.

6.4.3 Convergence

- BGP convergence is slow.
- BGP does not use periodic updates. Instead, as soon as a change occurs, i.e a path is lost or some of its PAs are changed, a new Update message is sent to its peers signaling the change(s).

6.4.4 New Multi-Paths Capability: ADD-PATH

*A **new capability** is being added to BGP to allow BGP peers to advertise multiple paths for the same address prefix in one UPDATE message without the new paths implicitly replacing any previous ones. The essence of the extension is that each path is identified by a path identifier in addition to the address prefix.*

This new extension will improve the stability of the overall BGP network by significantly reducing the convergence time: when a path becomes unavailable, each BGP router will have other paths already available at hand to change the route instead of needing another exchange process.

6.5 Computing Routes

6.5.1 Path Attributes

BGP uses path attributes to **communicate routing policies**.

	PA	Description	Enterprise Route Direction (Typical)
Mandatory →	NEXT_HOP	Lists the next-hop IP address used to reach a prefix.	N/A
	Weight ¹	A numeric value, range 0 through $2^{16} - 1$, set by a router when receiving Updates, influencing that one router's route for a prefix. Not advertised to any BGP peers.	Outbound
Mandatory for iBGP →	Local Preference (LOCAL_PREF)	A numeric value, range 0 through $2^{32} - 1$, set and communicated throughout a single AS for the purpose of influencing the choice of <u>best route for all routers out of that AS</u> .	Outbound
Mandatory →	AS_PATH (length)	The number of ASNs in the AS_Path PA.	Outbound, Inbound
Mandatory →	ORIGIN	Value implying the route was injected into BGP; I (IGP), E (EGP), or ? (incomplete information). ↓ "network" command → "redistribute" command	Outbound
Non Transitive →	Multi Exit Discriminator (MED)	Set and advertised by routers in one AS, impacting the BGP decision of routers in the other AS <u>into the first AS</u> . Smaller is better.	Inbound
Transitive →	Community	Groups routes and applies policies or decisions (accept, prefer) to those routes. It is a group of destinations that share some common property. This PA is not used in best path algorithm.	Outbound, Inbound

¹Weight is not a BGP PA; it is a Cisco-proprietary feature that acts somewhat like a PA.

BGP uses the AS_Path to perform two key functions:

- One of many criteria to choose the best route for a prefix based on the shortest AS_Path (fewest number of ASNs listed).
- Prevent routing loops between ASes.

Path attributes fall into four separate categories:

1. **Well-known mandatory:** Recognized by all BGP implementations and always present in the Updates
2. **Well-known discretionary:** Recognized by all BGP implementations and might be included in the Updates
3. **Optional transitive:** Might be recognized by all BGP implementations and must be advertised by a router to its peers even if it does not support the path attribute locally
4. **Optional non-transitive:** Might be recognized by all BGP implementations and might be advertised by a router to its peers even if it does not support the path attribute locally.

6.5.2 Best Path Algorithm

The BGP best path algorithm is used to discriminate between several paths to the same prefix/len received from different sources and it follows the steps shown in shorthand form in next table, after making sure that :

- **the path is not a loop**, by looking at AS_PATH (eBGP): if the local AS number is found, the update is ignored,
- **the path is not marked as “not synchronized”** if synchronization is enabled, and *OSPF neighbor's RID must equal BGP neighbor's RID*,
- **the path is not marked as “received-only”** in the `show ip bgp longer-prefixes` output.

BGP Decision Process + Mnemonic: N WLLA OMNI (ARI)

Step	Mnemonic Letter	Short Phrase	Which Is Better?
0	N	Next hop: reachable?	If no route to reach Next_Hop, router cannot use this route. <i>Not passed to BGP peers</i>
1	W	Weight	Bigger, <i>defaults to 32768 for locally injected, 0 for BGP routes</i>
2	L	LOCAL_PREF	Bigger, <i>defaults to 100, passed only to iBGP peers</i>
3	L	Locally injected routes	Locally injected is better than iBGP/eBGP learned, <i>using network, redistribute or aggregate-address commands</i>
4	A	AS_PATH length	Smaller.
5	O	ORIGIN	Prefer I over E. <i>IGP better than EGP better than Incomplete</i> Prefer E over ?
6	M	MED	Smaller, <i>defaults to 0 or IGP Metric for IGP redistributed routes</i>
7	N	Neighbor Type	Prefer eBGP over iBGP.
8	I	IGP metric to Next_Hop	Smaller.
9	A	Age of eBGP route	Oldest (longest-known) eBGP route
10	R	RID	Lowest neighbor BGP RID
11	I	IP address	Lowest neighbor IP address

Next-Hop
0.0.0.0

- If the BGP best path algorithm does not choose a best path by Step 8, the routes which still tie for being best path will be allowed into the IP routing table, up to the number defined by the BGP **maximum-paths number-of-paths** router subcommand.
- To join in CLN thread: “[Proposal for a BGP Enhanced Best Path Algorithm](#)”

6.5.3 RIB Failures

When BGP tries to install the best path prefix into Routing Information Base (RIB) (for example, the IP Routing table), [RTM](#) might reject the BGP route due to any of these reasons:

- Route with better administrative distance already present in the RIB.
- Memory failure...

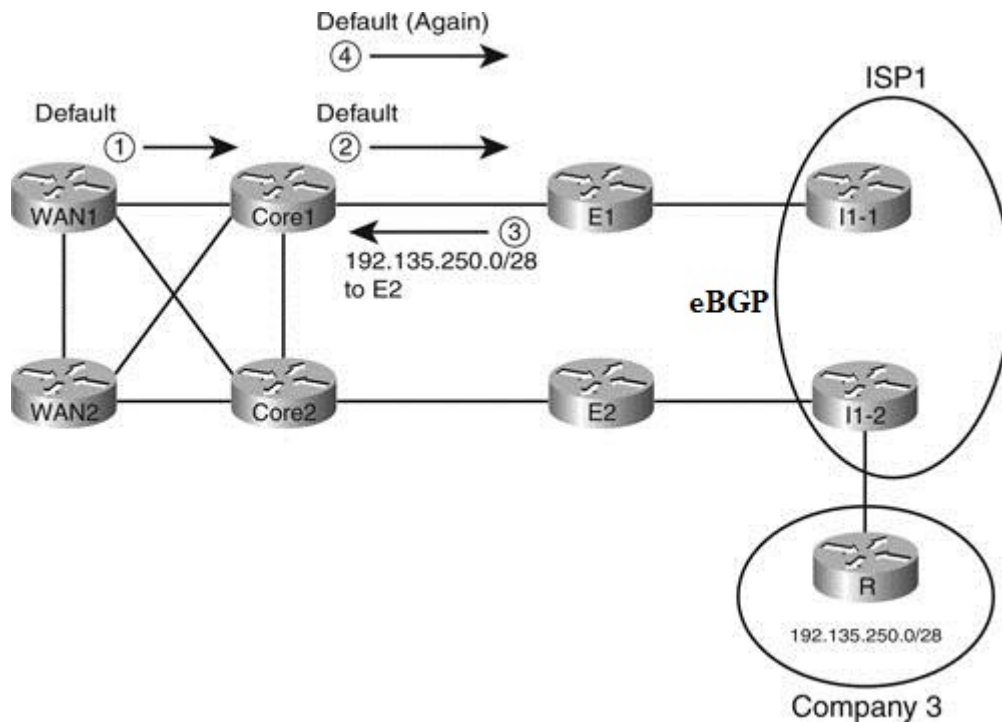
In such cases, the prefixes that are rejected for these reasons are **identified by “r”** (RIB Failure) in the `show ip bgp` command output and *are advertised to the peers (contrary to what most other distance-vector routing protocols do) with exceptions:*

- “*The BGP routes that are not used due to higher administrative distance are still advertised to all BGP peers, unless you configure ‘bgp suppress-inactive’*”, from [Ivan Pepelnjak's blog](#),
- “*It suppresses the rib-failure routes [in updates] if the next-hop of BGP rib-failure route is different from the next-hop of same route currently installed in the routing table*”, from [Zakir Khan's blog](#).

6.5.4 Avoiding Routing Loops when Forwarding Toward the Internet

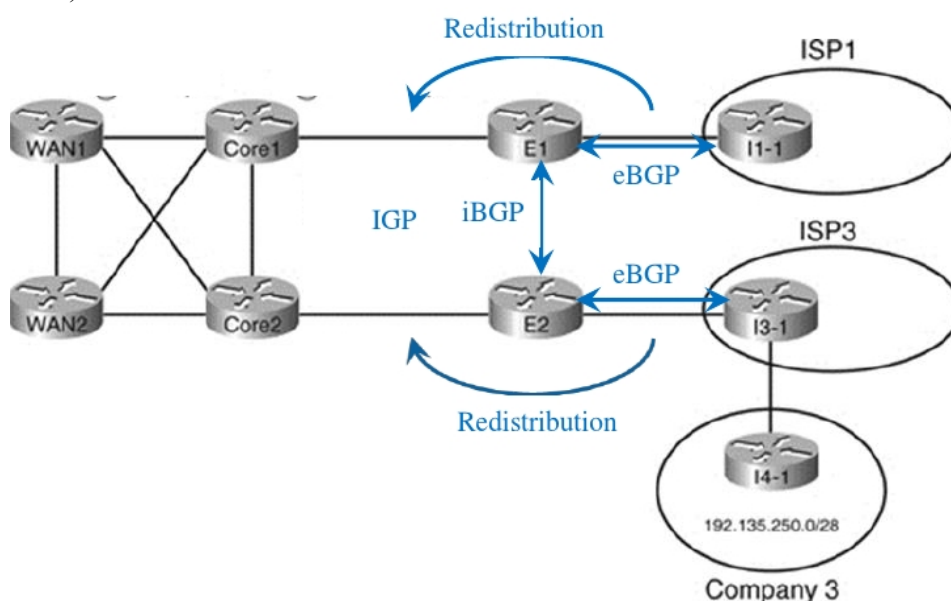
Routing loops can occur when the Internet-connected routers do not have a direct connection to each other, but are iBGP peers.

A Routing Loop for Packets Destined to 192.135.250.0/28 Without BGP in the Core



Some solutions exist to help these internal routers learn the routes:

- Run iBGP on at least some of the routers internal to the Enterprise (such as Core1 and Core2) with a full mesh of iBGP peers.
 - *In fact, a full mesh of BGP peers is a little too much: the peering between internal routers Core1 and Core2 is not necessary because useless; by default, they will not exchange any BGP route!*
- The BGP Confederations or BGP Route Reflector features can be used instead of a full mesh of iBGP peers.
- iBGP Peering, Redistribution of eBGP into the IGP and BGP Synchronization (between iBGP-learned routes and IGP-learned routes): *not recommended*.



The **synchronization** feature tells a BGP router the following:

- Do not consider an iBGP-learned route as “best” unless the exact prefix was learned via an IGP and is currently in the IP routing table.

6.6 Clearing Peering

A reset must occur after having introduced new prefix filters or after having modified some PAs for the changes to be considered by the neighbor(s).

When a **hard reset** occurs, the local and neighboring router:

- First :
 - removes all BGP table entries learned from that neighbor, just like it does for any failed BGP neighborhood,
 - brings down the neighborhood then the underlying TCP connection,
- Then :
 - brings back up the underlying TCP connection then the neighborhood,
 - exchange all BGP table entries with that neighbor,

With a **soft reset**, the local router :

- does *not* bring down the BGP neighborhood,
- does *not* bring down the underlying TCP connection,
- re-sends outgoing Updates, adjusted per the outbound filter and/or reprocesses incoming Updates per the inbound filter, which adjusts the BGP tables based on the then-current configuration.

6.7 Summary

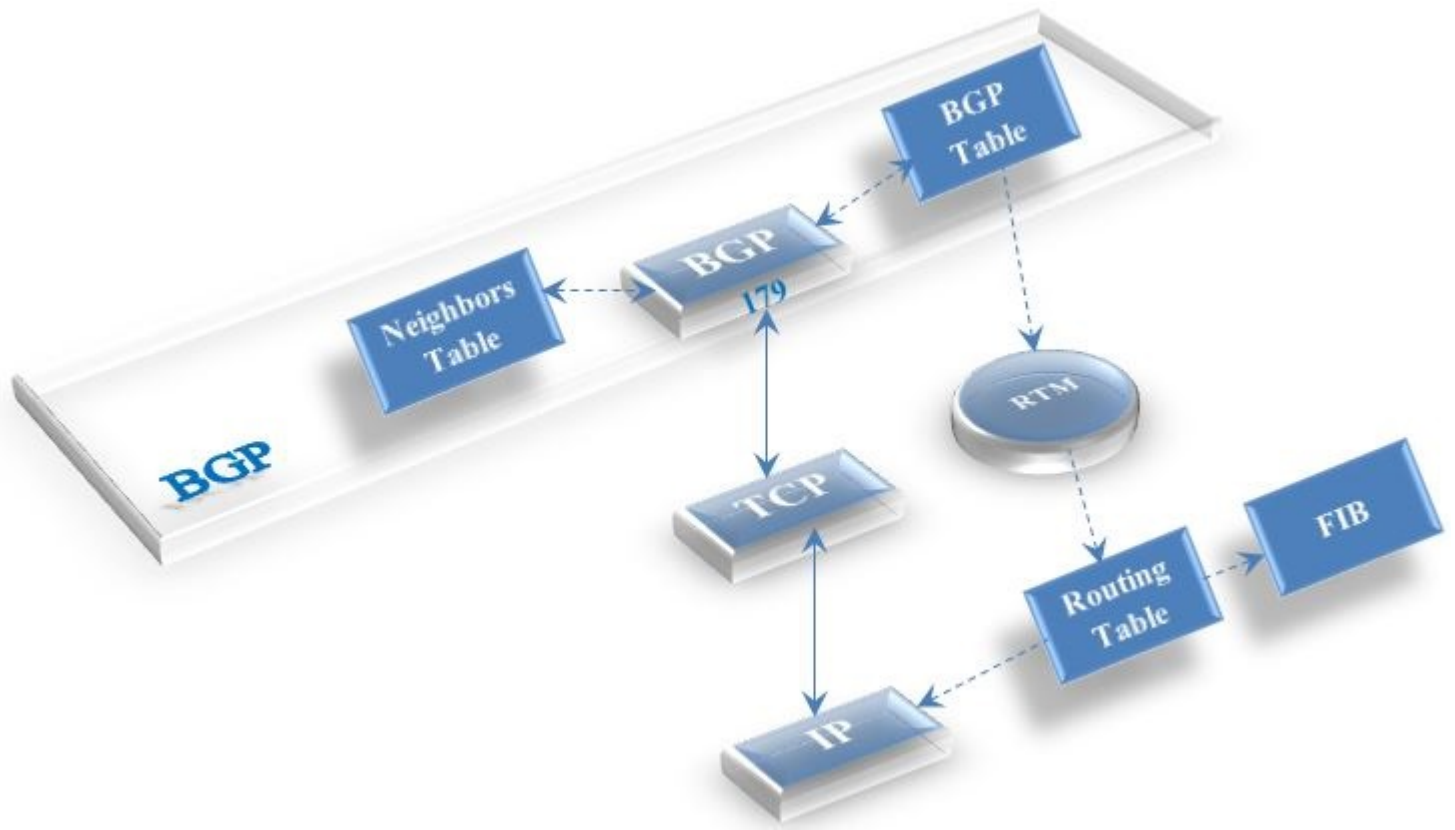
6.7.1 Features

OSPF/EIGRP	BGP
Forms neighbor relationship before sending routing information	Same.
Neighbors typically discovered using multicast packets on the connected subnets	Neighbor IP address is explicitly configured and may not be on common subnet.
Does not use TCP	Uses a TCP connection between neighbors (port 179).
Advertises prefix/length	Advertises prefix/length, called <i>Network Layer Reachability Information (NLRI)</i> .
Advertises metric information	Advertises a variety of path attributes (PA) that BGP uses instead of a metric to choose the best path.
Emphasis on fast convergence to the truly most efficient route	Emphasis on scalability; may not always choose the most efficient route.
Link state (OSPF) or distance vector (EIGRP) logic	Path vector logic (similar to distance vector).

Enterprises typically have two major classes of options for outbound routing toward the Internet:

- default routing,
- BGP:
 - First, it makes most sense to use BGP when you have at least two Internet connections.
 - Second, BGP becomes most useful when you want to choose one outbound path over another path for particular destinations in the Internet. In short, when you have multiple Internet connections, and you want to influence some packets to take one path and some packets to take another, consider BGP.

6.7.2 Stack



6.7.3 Relations With Other Routing Protocols

Cf. [RTM](#)

6.8 Configuration Commands

6.8.1 Controlling eBGP/iBGP Peerings

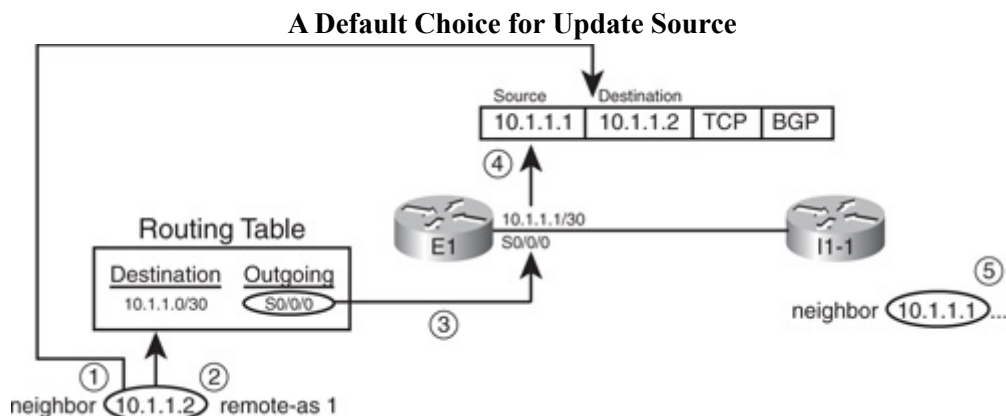
6.8.1.1 Static Peering

(config)#**router bgp** *asn*

The router's own ASN

(config-router)#**neighbor** *neighbor-ip* **remote-as** *remote-asn*

The IP address of each neighbor and that neighbor's ASN



Configuring eBGP/iBGP peers to use a loopback interface IP address with BGP requires several steps, as follows:

- **Step 1:** Configure an IP address on a loopback interface on each router.
- **Step 2:** Tell BGP on each router to use the loopback IP address as the source IP address using:
(config-router)#**neighbor... update-source** *interface-id*
- **Step 3:** Configure BGP on each router to refer to the other router's loopback IP address using:
(config-router)#**neighbor** *neighbor-ip* **remote-as** *remote-asn*.
- **Step 4:** Make sure each router has IP routes so that they can forward packets to the loopback interface IP address of the remote router.
- **Step 5:** Configure eBGP multihop using:
(config-router)#**neighbor... ebgp-multihop** [*hops*] *hops*=255 if not specified, command not necessary for iBGP.
- **Step 6:** Configure iBGP next-hop-self if necessary using:
(config-router)#**neighbor** *neighbor-ip* **next-hop-self** The router advertises iBGP routes *with its own update-source* IP address as the next-hop IP address. And because the iBGP neighborhood already relies on a working route for these update source IP addresses, if the neighborhood is up, then IP routes already exist for these next-hop addresses.

6.8.1.2 Shutting down the Peering

(config-router)#**neighbor** *neighbor-ip* **shutdown**

Disables a BGP neighborhood and removes all associated routing information and *stay that way* (different from **clear ip bgp** command). Those disabled neighbors have an Idle status and the *Admin* entry in the **show ip bgp summary**.

(config-router)#**no neighbor** *neighbor-ip* **shutdown**

Enables a BGP neighborhood

6.8.1.3 Authenticating BGP Messages

(config-router)#**neighbor** *neighbor-ip* **password** *key*

MD5 Authentication of BGP messages

6.8.2 Seeding the BGP Table

6.8.2.1 Using network command

6.8.2.1.1 No Auto-summary by Default

(config-router)#**network** *network-number* [**mask** *network-mask*] [**route-map** *map-name*]

- Look for a route in the router current IP routing table that *exactly* matches the parameters of the **network** command ; if a route for that *exact* prefix/length exists, put the equivalent prefix/length into the local BGP table.
- The route map should be examined to filter the networks to be advertised. If not specified, all networks are advertised. If the keyword is specified, but no route map tags are listed, no networks will be advertised.

6.8.2.1.2 The Effect of Auto-summary

The change in logic occurs *only* when the **network** command *omits* its **mask** parameter; there is no difference in logic if the **mask** parameter is explicitly configured. When the **network** command refers to a Class A, B, or C network, with no **mask** parameter configured, and with **auto-summary** configured, the router adds a route for that *classful* network to the BGP table:

- if the exact classful route is in the IP routing table
- or if any subset routes of that classful network are in the routing table

In summary, of the two actions in the list, the first occurs regardless of the **auto-summary** setting, and the second occurs only if **auto-summary** is configured.

6.8.2.2 Using Redistribution

(config-router)#**redistribute** *source-protocol* [*process-id* | *as-number*] [**match** {**internal** | **external 1** | **external 2** | **nssa-external**}] [**tag** *tag-value*] [**route-map** *name*]

With a **route-map** (*for example to filter out all private and external IP address ranges [no transit AS]*), all the subordinate **public** subnets could be advertised by BGP into the ISP.

6.8.3 Optimizing Convergence Delay

6.8.3.1 Keepalive/Hold Timers

Per neighbor:

(config-router)#**neighbor** [*neighbor-ip*] **timers** *keepalive holdtime*

- *keepalive*: **frequency (in seconds) with which the router sends keepalive messages to its peer. The default is 60 seconds. The range is from 0 to 65535.**
- *holdtime*: **Interval (in seconds) after not receiving a keepalive message that the router declares a peer dead. The default is 180 seconds. The range is from 0 to 65535.**

For all neighbors:

(config-router)#**timers** *bgp keepalive holdtime*

6.8.4 Optimizing Bandwidth Utilization

6.8.4.1 Equal Path Load Sharing → Cf. [Load-Balancing Summary](#)

(config-router)#**maximum-paths** *number-of-paths*

Sets the maximum number of parallel routes that BGP can support (1 by default)

6.8.5 Choosing One Path Over Another

➤ The neighborhood must be cleared (cf. [§Clearing Neighbors](#)) in all the following cases.

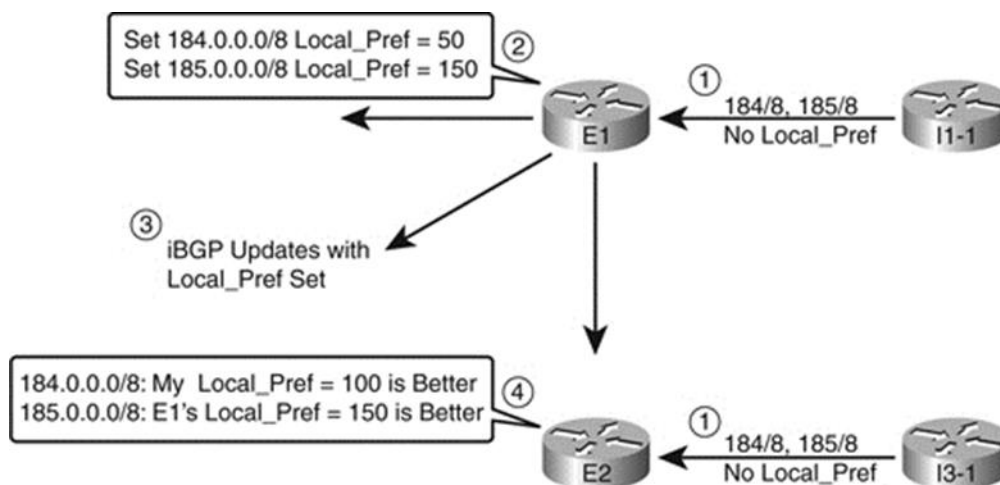
6.8.5.1 Influencing Enterprise's Outbound Routes

6.8.5.1.1 Using Administrative Weight

Feature	Description
Is it a PA?	No; Cisco proprietary feature
Purpose	Identifies a single router's best route
Scope	Set on inbound route Updates; influences only that one router's choice
Range	0 through 65,535 ($2^{16} - 1$)
Which is best?	Bigger values are better
Default	0 for learned routes, 32,768 for locally injected routes
Defining a new default	Not supported
Configuration	neighbor route-map (per prefix) in / set weight neighbor weight (all routes learned from this neighbor)

6.8.5.1.2 Using Local_Pref

Feature	Description
PA?	Yes
Purpose	Identifies the best exit point from the AS to reach a given prefix
Scope	Throughout the AS in which it was set; not advertised to eBGP peers
Range	0 through 4,294,967,295 ($2^{32} - 1$)
Which is best?	Higher values are better
Default	100
Changing the default	Using the bgp default local-preference <0-4294967295> BGP sub-command
Configuration	Via neighbor route-map command; in option is required for updates from an eBGP peer - in out with iBGP peers / set local-preference



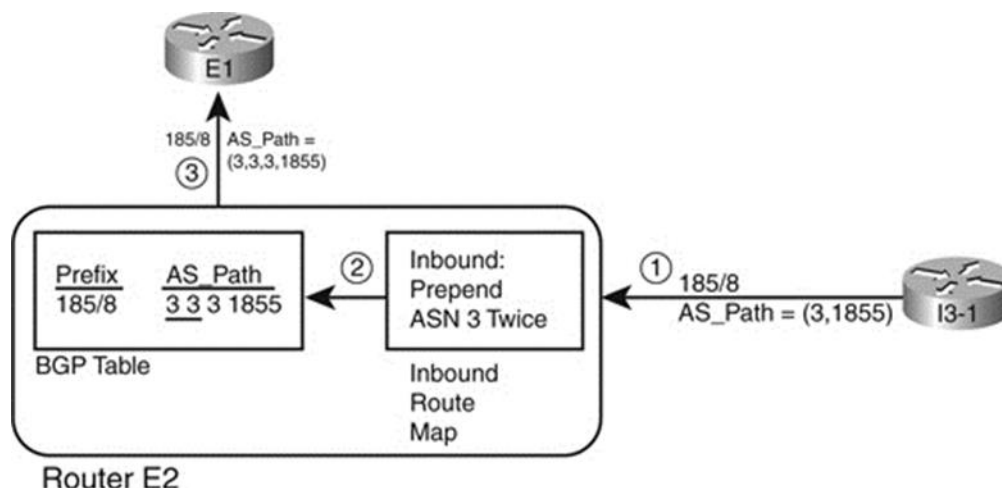
6.8.5.1.3 Using AS_Path

(config-router)#**neighbor** *neighbor-ip* **route-map** {in | out}

- applies the route map to all BGP Updates received/sent from/to the listed neighbor. The router allows routes first matched in a permit clause and discards routes first matched with a deny clause.
- changes the PAs of routes by using the **set** command within permit clauses (Although you can configure a **set** command in a route map deny clause, the set has no effect because the deny clause filters the route) :
 - **set as-path prepend** *as-number as-number* ...

➤ *Does not work on iBGP sessions*

➤ *Can be used by a multi-homed enterprise to implement a backup internet link by advertising a longer AS_PATH to the backup ISP for its prefix(es).*



6.8.5.2 Influencing Enterprise's Inbound Routes

6.8.5.2.1 Using AS_Path

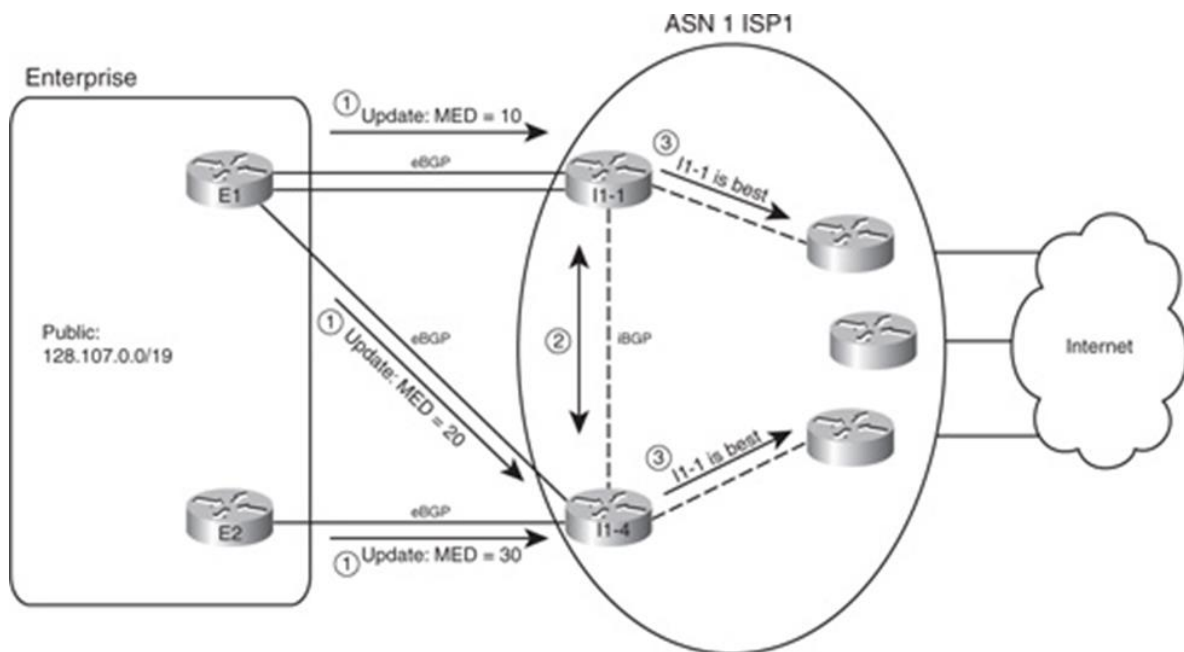
Although presented above as a tool for influencing outbound routes, AS_Path prepending can *also* be used to *influence the inbound routes as well*.

6.8.5.2.2 Using MED

Feature	Description
Is it a PA?	Yes.
Purpose	Allows an AS to tell a neighboring AS the best way to forward packets into the first AS.
Scope	Advertised by one AS into another, propagated inside the AS, but not sent to any other autonomous systems. <i>(where the metric returns to 0)</i>
Range	0 through 4,294,967,295 ($2^{32} - 1$).
Which is best?	Smaller is better.
Default	0
Configuration	Via neighbor route-map out command, using the set metric command inside the route map.
Changing the default	(config-router)#default-metric <i>metric</i>

(config-router)#**bgp always-compare-med**

Enables the comparison of the Multi Exit Discriminator (MED) for paths from neighbors in different autonomous systems. By default (without this command), the MED is compared only if the AS_Path for the compared routes is identical.



6.8.6 Optimizing Router Resources

6.8.6.1 Limiting Reachability: Route Filtering

6.8.6.1.1 Filtering Tools

BGP Subcommand	Commands Referenced by neighbor Command	What Can Be Matched	
neighbor distribute-list (standard ACL)	access-list, ip access-list	Prefix, with WC mask	➔ Range of Prefixes
neighbor distribute-list (extended ACL)	access-list, ip access-list	Prefix and prefix length, with WC mask for each	➔ Range of Prefixes Range of Lengths
neighbor prefix-list	ip prefix-list	Exact or "first N" bits of prefix, plus range of prefix lengths	➔ Range of Prefixes Range of Lengths
neighbor filter-list	ip as-path access-list	AS_PATH contents; all NLRI whose AS_PATHs are matched considered to be a match	➔ Range of ASes
neighbor route-map	route-map	Prefix, prefix length, AS_PATH, and/or any other PA matchable within a BGP route map	➔ Range of Prefixes Range of Lengths Range of ASes Range of metrics

(config-router)#**distribute-list** {acl-number | **prefix list-name**} {in | out}

- For all neighbors.

(config-router)#**neighbor neighbor-ip distribute-list** {access-list-number | access-list-name} {in | out}

- With an *standard* ACL, IOS compares the source IP address parameter to the subnet number of the route :
 - Filtering prefix: ... **host prefix**
 - Filtering *range* of prefixes: ... **prefix wildcard-mask**
- With an *extended* ACL, IOS compares the source IP address parameter to the subnet number of the route and the destination IP address to the subnet mask of the route :
 - Filtering prefix/len : ... **ip host prefix host subnet-mask**
 - Filtering *range* of prefixes/lengths : ... **ip prefix wildcard-mask subnet-mask wildcard-mask**
- **Either neighbor distribute-list or neighbor prefix-list** at the same time *in any given direction*.

(config-router)#**neighbor neighbor-ip prefix-list prefix-list-name**{in | out}

<https://t.me/learningnets>

(config-router)#neighbor neighbor-ip filter-list as-path-access-list-number {in | out}

➤ Cf. [§AS_Path ACLs](#).

(config-router)#neighbor neighbor-ip route-map map-name {in | out}

(config-router)#neighbor neighbor-ip remove-private-as *To remove private AS numbers from eBGP outbound routing updates*

6.8.6.1.2 Clearing Neighbors

#clear ip bgp {* | neighbor-ip}

Hard Reset: clears BGP peer connection, causing the filter to take effect.

#clear ip bgp {* | neighbor-ip} {in | out}

Soft Reset:

- **in**: the router uses *route refresh* feature to ask the neighbor to re-send its full BGP Update, and then the router can apply its current inbound BGP filters, updating its BGP table
- **out**: the router *builds a new Update based on the existing configuration*, so any new or changed outbound filters affect the contents of the Update.

#clear ip bgp {* | neighbor-ip} soft [in | out]

(Old-fashioned) Soft Reset:

- **in**: the router reapplies the inbound filter to the cached received Update, updating the local router's BGP table.
 - Needs **neighbor neighbor-id soft-reconfiguration inbound** to cause the router to retain the received BGP Updates from that neighbor.
- **out**: the router *builds a new Update based on the existing configuration*, so any new or changed outbound filters affect the contents of the Update.
- without **in | out**: both directions

6.8.6.2 Without Limiting Reachability

6.8.6.2.1 Route Summarization

(config-router)#aggregate-address address mask [summary-only]

- Advertises an entire summary IP address prefix **when any subset routes of address mask exists in the BGP and RIB tables** (IP routing table),
- Creates an entry in the IP routing table for this Prefix, "via 0.0.0.0" pointing to Null0
- The **summary-only** keyword tells IOS to suppress the advertisement of the subset routes, as noted by the code "s" beside the subset routes listed in the BGP table.

6.8.6.2.2 Default Route

(config-router)#neighbor neighbor-ip default-originate [route-map map-name]

Allows a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route:

- *This command does not require the presence of 0.0.0.0 in the local router.*
- *When used with a route map, the default route 0.0.0.0 is injected if the route map contains a match ip address clause and there is a route that matches the IP standard or extended access list exactly.*

➤ *It is better to implement this on the ISP side to send a default route to a customer; if his CPE router had configured a static default route towards the ISP PE router and this one dies and the interface stays UP on the CE router, the traffic from the CE router would be blackholed. If the customer is multi-homed and receives default routes from both ISPs through BGP, it can reroute the traffic towards the backup ISP in case of a remote failure in the primary ISP.*

6.8.7 Synchronizing Routes Between iBGP and IGP

(config-router)#synchronization

Allows routers within an AS to have the route before BGP makes it available to other ASs.

(config-router)#no synchronization

Disables it.

➤ *Use only one IGP in the BGP AS core to avoid inefficient routing issues (cf. [BGP/IGP Problem](#)).*

<https://t.me/learningnets>

6.9 Verification Commands

#show ip bgp [prefix/len longer-prefixes]

Lists the entirety of the BGP routing table:

- ‘>’: denotes the chosen best route,
- ‘i’: learned using iBGP / the absence of the letter ‘i’: learned using eBGP,
- NLRI (prefix/len) - Next hop – MED – Local_Pref - Weight - AS_Path – Origin,
- The *oldest* route is the last one per prefix/len,
- With *prefix/len*, displays a more detailed view of the BGP table entry:
 - ‘internal’: iBGP-learned,
 - ‘external’: eBGP-learned,
 - ‘best’: “best” route,
 - the *next hop address* of the route, the *neighbor* from which the route was learned and the *neighbor’s BGP RID*.

Note : BGP **show** commands list the AS_Path with the *first-added ASN on the right* and the last-added ASN on the left. BGP uses this convention because when BGP adds an ASN to the AS_Path, BGP prepends the ASN to the list, causing the new ASN to show up as the leftmost ASN in the AS_Path.

Finding PA Settings in the Output of the show ip bgp Command

R3 #show ip bgp

BGP table version is 12, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path	Origin
* 11.0.0.0	10.1.36.6			0	65000 1 33333 10 200 44 i	
* 12.0.0.0	10.1.35.5			0	5 1 33333 10 200 44 i	
* i ← Neighbor Type	10.1.14.4	0	100	0	(111) 4 1 33333 10 200 44 i	
* 10.1.34.4	10.1.34.4			0	4 1 33333 10 200 44 i	
* 10.1.36.6	10.1.36.6			0	65000 1 33333 10 200 44 i	
* 10.1.35.5	10.1.35.5			0	5 1 33333 10 200 44 i	
* i	10.1.14.4	0	100	0	(111) 4 1 33333 10 200 44 i	
* 10.1.34.4	10.1.34.4			0	4 1 33333 10 200 44 i	
* i 16.0.0.0/4	10.1.14.4	0	100	0	(111) 4 {1, 404, 303, 202} i	

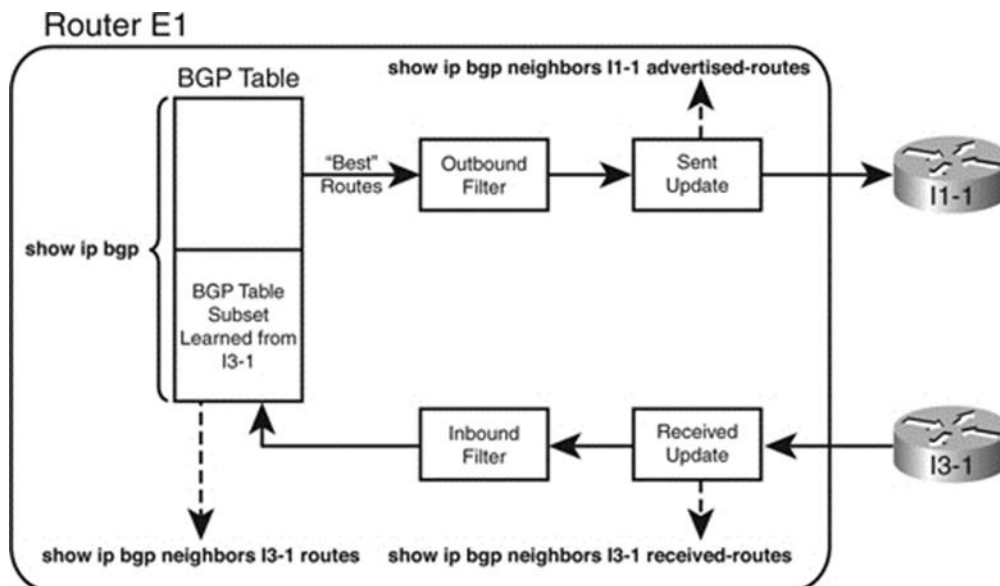
Comments: To Discover Other Details...
Neighbor Type: No Letter Means "EBGP"
IGP Metric: **show ip route next-hop-address**
RID: **show ip bgp nlri**

#show ip bgp summary

- The local router’s BGP RID - the local router’s ASN
- Statistics for the BGP table entries:
 - The neighbor IP address as defined on the local router’s **neighbor** command and not the neighbor’s BGP RID
 - the neighbor’s ASN (as configured on the local router’s **neighbor remote-as** command),
 - the time spent in the current state,
 - State/PfxRcd : either lists the BGP neighbor state, or the number of prefixes received from that neighbor. A numeric value under this heading implies a neighbor state of *Established*, because the peers must be in *Established* state before Updates can be sent. If the peer is not in an *Established* state, the value in this heading lists the text name of the current BGP state

Verification Commands for eBGP-Learned Routes

Verification Step	Command
List possible default routes.	<code>show ip bgp 0.0.0.0 0.0.0.0</code>
List possible routes, per prefix.	<code>show ip bgp prefix {[subnet-mask] [/len]} [longer-prefixes]</code>
List routes learned from one neighbor, before any inbound filtering is applied.	<code>show ip bgp neighbors ip-address received-routes</code> <i>Needs 'neighbor neighbor-ip soft-reconfiguration inbound'</i>
List routes learned from a specific neighbor that passed any inbound filters.	<code>show ip bgp neighbors ip-address routes</code>
Lists routes advertised to a neighbor after applying outbound filtering.	<code>show ip bgp neighbors ip-address advertised-routes</code>
List the number of prefixes learned per neighbor.	<code>show ip bgp summary</code>



#show ip bgp neighbors [neighbor-id]

- The neighbor is an eBGP/iBGP neighbor (external/internal link)
- The neighbor's BGP RID and ASN
- The current state.
- Route refresh capability.
- Prefix activity (# sent / # received)
- The eBGP multihop setting (# hops).
- Local and remote ('foreign') TCP socket information (IP addresses and port numbers).

#show ip bgp regexp regexp *Shows all matches in the BGP table with this regular expression*

#show ip bgp rib-failure *Lists one line for each best BGP route that the RTM does not place into the IP routing table.*

#show tcp brief *Confirms the TCP socket details of the underlying TCP connection*

Extended Ping with record option *Records outbound interfaces on the way to the destination and the way back.*

#debug ip bgp Lists the somewhat-readable BGP messages. These messages list the BGP states and shows the information inside the Open messages.

<https://t.me/learningnets>

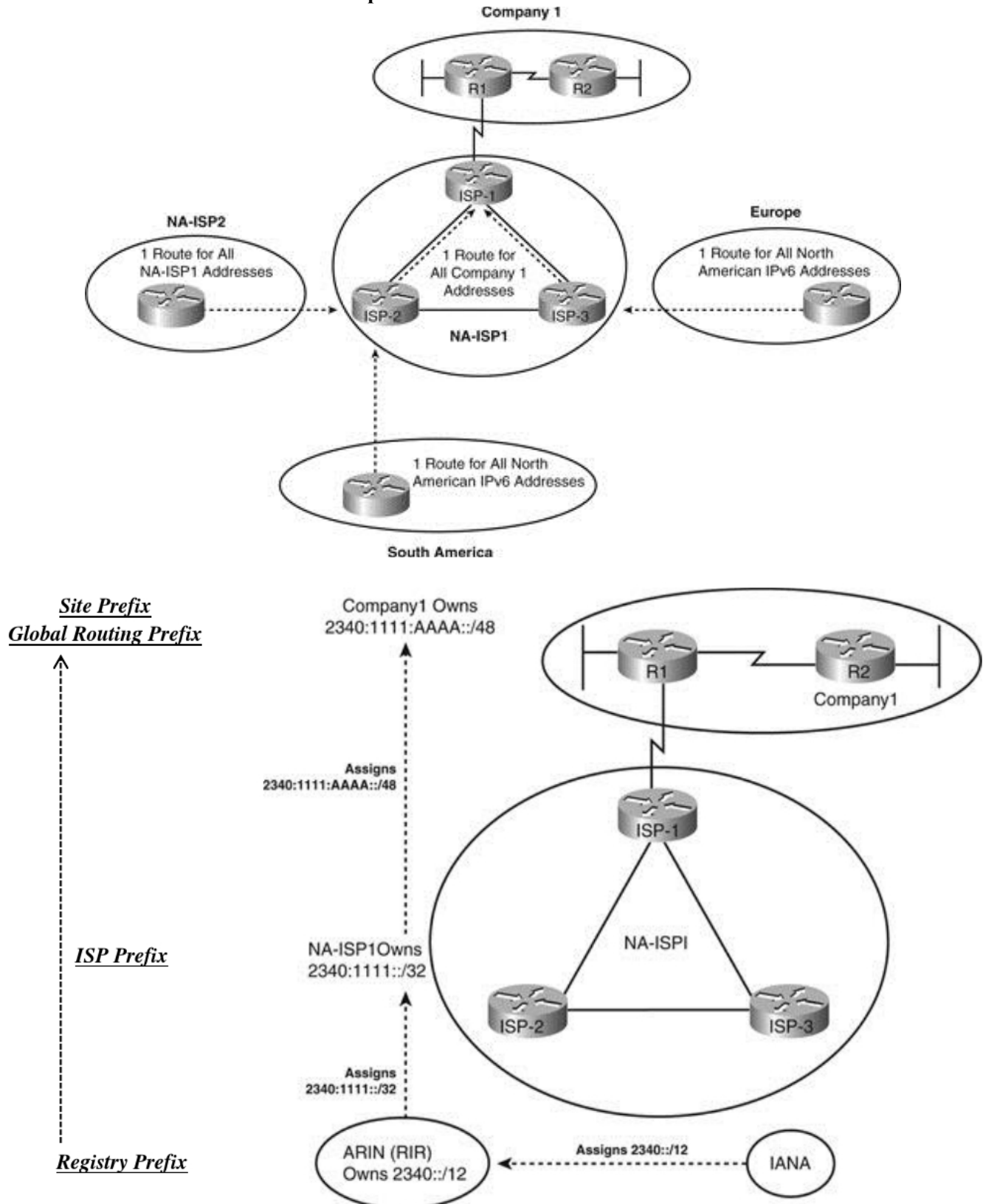
7 IPv6

7.1 Hierarchical Prefix Assignment in the Internet

IANA delegates regional assignments to Regional Internet Registries (RIR). The five RIRs are:

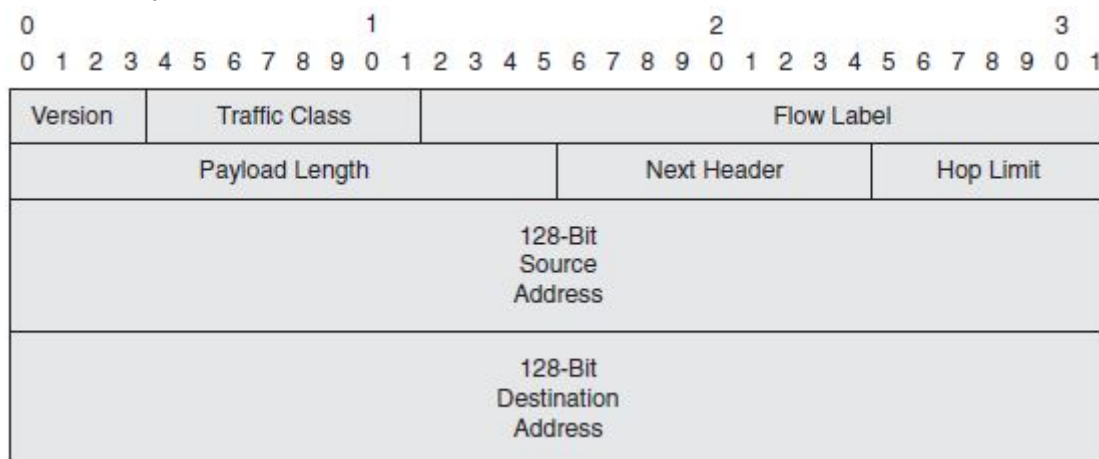
- ARIN (American Registry for Internet Numbers)
- RIPE NCC (Réseaux IP Européens Network Control Center)
- APNIC (Asia Pacific Network Information Center)
- LACNIC (Latin America and Caribbean Network Information Center)
- AfriNIC (African Network Information Centre)

Conceptual View of IPv6 Global Routes



7.2 Header

The IPv6 header is simpler than the IPv4 header. Some IPv4 fields have been eliminated or changed to optional fields. The IPv6 header size is *40 bytes*. The **fragment offset fields and flags in IPv4 have been eliminated** from the header. IPv6 adds a **flow label** field for QoS mechanisms to use.



- **Traffic Class:** This field is 8 bits in length. It describes the class or priority of the IPv6 packet and provides functionality **similar to the IPv4 ToS/DS** field.
- **Flow Label:** This field is 20 bits in length. It indicates a specific sequence of packets between a source and destination that requires special handling, such as real-time data (voice and video) and **service chaining**.
- **Next Header:** This field is 8 bits in length. It identifies the upper-layer protocol. It uses the values defined by the Internet Assigned Numbers Authority (IANA); they are the **same as the ones used for the 'Protocol' IPv4 field**.

Protocol Number	Protocol
6	Transmission Control Protocol (TCP)
17	User Datagram Protocol (UDP)
50	Encapsulating Security Payload (ESP)
51	Authentication Header (AH)
58	ICMP for IPv6
59	No Next Header for IPv6
60	Destination Options for IPv6
88	Enhanced IGRP (EIGRP)
89	Open Shortest Path First (OSPF)

- **Extended Headers:**
 - *Authentication Header (AH),*
 - *Encapsulating Security Payload (ESP) header,*
 - *Routing header: used for source routing,*
 - *Fragment header: included in fragmented datagrams to provide information to allow the fragments to be reassembled,*
 - *Hop-by-hop header: used to support jumbo-grams.*

7.3 Addresses

7.3.1 IANA Address Allocation

7.3.1.1 Assigned and Unassigned Prefixes

Hexadecimal/Prefix	Allocation
0000::/8	Unspecified, loopback, IPv4-compatible
0100::/8	Unassigned
0200:/7	Unassigned
0400::/7	Reserved for Internetwork Packet Exchange (IPX) allocation
0800::/5	Unassigned
1000::/4	Unassigned
2000::/3	Global unicast address
4000::/3	Unassigned
6000::/3	Unassigned
8000::/3	Reserved for geographic-based unicast addresses
A000::/3	Unassigned
C000::/3	Unassigned
E000::/3	Unassigned
F000::/5	Unassigned
F800::/6	Unassigned
FC00::/7	Unique Local Unicast
FE00::/9	Unassigned
FE80:/10	Link-local unicast addresses
FEC0::/10	Unassigned; was site-local unicast addresses (deprecated)
FF00::/8	Multicast addresses

Geographic-based unicast addresses: their allocation have been dropped in favor of developing improved automatic and dynamic host configuration under IPv6—including the ability to automatically renumber entire networks when providers are changed (from [IPv6: Theory, Protocol, and Practice, 2nd Edition, Edition 2](#), by [Peter Loshin](#)).

7.3.1.2 PA Address Space

*A **Provider-Assigned** address space is a block of IP addresses assigned by an ISP directly to an end-user organization.*

7.3.1.3 PI Address Space

*A **Provider-Independent** address space is a block of IP addresses assigned by a Regional Internet Registry (RIR) directly to an end-user organization. The user must contract with an Internet service provider to obtain routing of the address block within the Internet.*

Provider-independent addresses offer end-users the opportunity to change service providers without renumbering of their networks and to use multiple access providers in a multi-homed configuration. However, provider-independent blocks may increase the burden on global routers, as the opportunity for efficient route aggregation through Classless Inter-Domain Routing (CIDR) may not exist.

*Assignments are taken from the address range **2001:678::/29** and have a minimum size of a **/48** prefix.*

The anycasting TLD (Top-Level Domains) Name servers prefixes are taken from the same address space.

7.3.1.4 PI vs PA Address Spaces

[PA, PI or ULA IPv6 Address Space? It depends, 01/2014, ipSpace](#)

PI/48 and PA/32 are the longest prefixes currently accepted by all ISPs.

7.3.2 Unicast Address Types

Like IPv4, hosts and routers assign these IP addresses to a single interface for the purpose of allowing that one host or interface to send and receive IP packets.

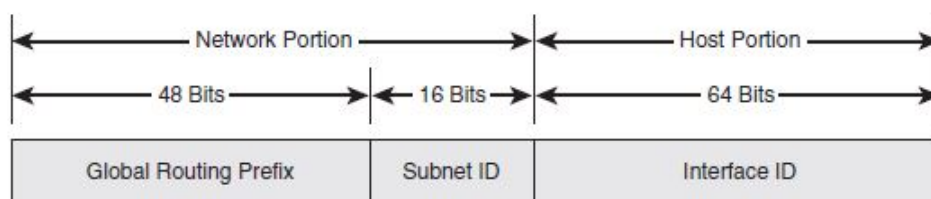
Type of Address	Purpose IPv4 Equivalent	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet: 1.0.0.0/8 up to 223.0.0.0/8 except following IPv4	2000::/3	2 or 3
Unique local	Unicast packets inside one organization 10.0.0.0/8 - 172.16.0.0/12 192.168.0.0/16	FC00::/7	FC
Link local	Packets sent in the local subnet 169.254.0.0/16	FE80::/10	FE8*
Site local	Deprecated; originally meant to be used like private IPv4 addresses	FECO::/10	FEC, FED, FEE, FEF
Unspecified	An address used when a host has no usable IPv6 address 0.0.0.0	::/128	N/A
Loopback	Used for software testing, like IPv4's 127.0.0.1	::1/128	N/A

- *Used as the source address for RS/RA messages for Router Discovery*
- *Used as the source address for NS/NA messages and as the destination address for NA messages for Neighbor Discovery (the equivalent of ARP for IPv6)*
- *As the next-hop IPv6 address for IP routes*

*IPv6 RFCs define the FE80::/10 prefix, which technically means that the first three hex digits could be FE8, FE9, FEA, or FEB. However, bit positions 11-64 of link local addresses should be 0, so in practice, link local addresses should always begin with FE80.

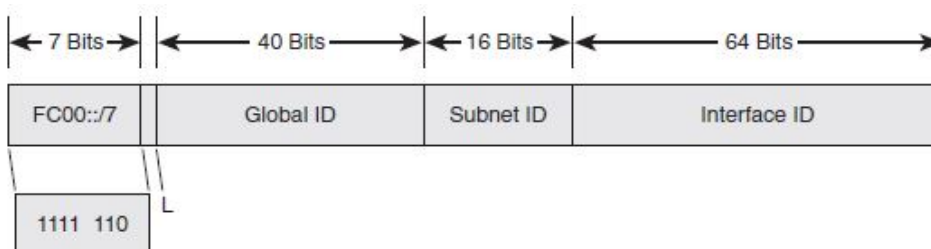
7.3.2.1 Global Unicast

Example of Global Unicast Address Format



7.3.2.2 Unique Local

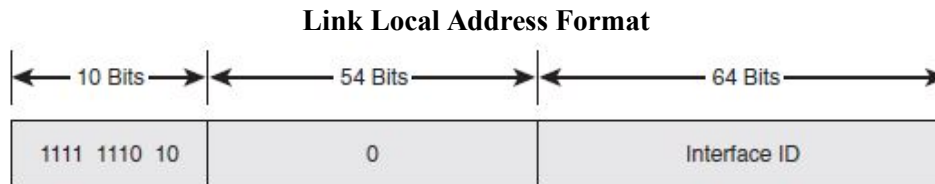
Example of Unique Local Address Format



The bit labeled L is set to 1 if the prefix is locally assigned, and setting it to 0 has not been defined.

<https://t.me/learningnets>

7.3.2.3 Link Local



Out of the scope of CCNP ROUTE:

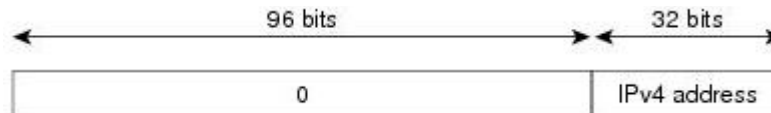
RFC 7404 suggests using only LLA (Link-Local Addresses) on infrastructure links between routers. This approach yields to the following several advantages (the caveats can be easily solved):

- ***Smaller routing tables***
- ***Reduced attack surface***
- ***Lower configuration complexity***
- ***Simpler DNS***

7.3.2.4 IPv6 Addresses with Embedded IPv4 Addresses

7.3.2.4.1 IPv4-Compatible IPv6 Address

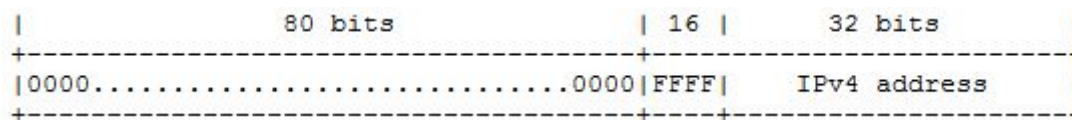
The "IPv4-Compatible IPv6 address" was defined to assist in the IPv6 transition. The format is as follows:



Notes:

- *The IPv4 address used in the "IPv4-Compatible IPv6 address" must be a globally-unique IPv4 unicast address.*
- *The "IPv4-Compatible IPv6 address" is now **deprecated** because the current IPv6 transition mechanisms no longer use these addresses. New or updated implementations are not required to support this address type.*

7.3.2.4.2 IPv4-Mapped IPv6 Address



See [RFC 4038](#) for background on the usage of the "IPv4-mapped IPv6 address".

7.3.2.5 Address Assignment

Method	Dynamic or static	Prefix and length learned from...	Host learned from...	Default router learned from...	DNS addresses learned from...
Stateful DHCP	Dynamic	DHCP Server	DHCP Server	Router, using NDP	(Stateful) DHCP Server
Stateless autoconfig	Dynamic	Router, using NDP	Derived from MAC	Router, using NDP	Stateless DHCP or NDP option
Static configuration	Static	Local config	Local config	Router, using NDP	Stateless DHCP
Static configuration with EUI-64	Static	Local config	Derived from MAC	Router, using NDP	Stateless DHCP

7.3.2.5.1 Dynamic Configuration

7.3.2.5.1.1 RA Flags

These flags may be set in Router Advertisements sent by the routers :

- **A: Auto-configuration:** The prefix may be used for SLAAC; when a host, server or router receives a RA with this flag set, it may use SLAAC even though its address is statically configured, and use the SLAAC address for inbound and outbound sessions (depending on how the “source address selection” algorithm is implemented). It is possible to clear it in the campus/data center to avoid this situation if SLAAC is not expected.
- **L: On-Link :** Prefixes with this flag set are on-net; destinations with these prefixes are sent directly ; others go through the first-hop router.
- **O: Other configuration:** Other information is available through DHCPv6, such as DNS servers addresses.
- **M: Managed addresses:** Addresses are available via DHCPv6

There's an RA option ([RFC 6106](#)) to learn the DNS servers IPv6 addresses which makes DHCPv6 useless in SLAAC.

7.3.2.5.1.2 DHCPv6

- The DHCPv6 server tracks state information: **stateful** DHCPv6 servers fill the same role as the older DHCPv4 servers.
- All the DHCPv6 options – including DNS and NTP servers - are available [here](#).
- Hosts are identified by their **DUID** (DHCP Unique ID) instead of their MAC address.
- **RAs are still needed in DHCPv6 networks, as there is no DHCPv6 Default Gateway option.**

7.3.2.5.1.3 SLAAC

There are two types of **Stateless Address Auto-Configuration SLAAC**:

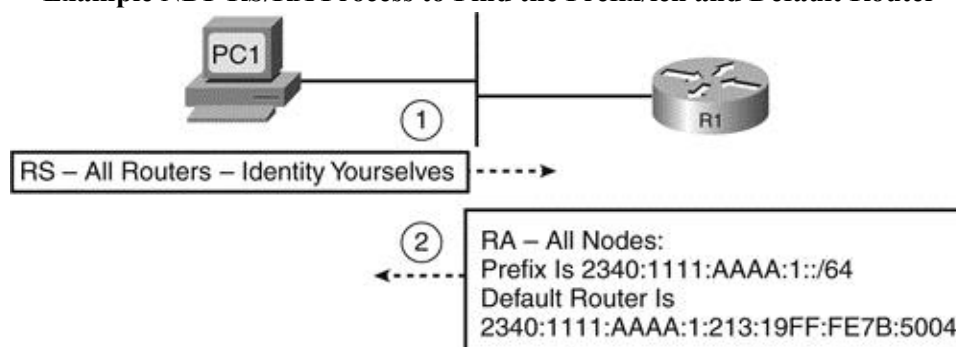
- First, a **stateless auto-configuration (EUI-64) of the link-local address(es)** without any connection to DHCPv6 server, with a DAD (Duplicate Address Detection) for each link local address,
- Then, **stateless auto-configuration of global unicast address** with a connection to a DHCPv6 server for DNS servers IPv6 addresses: these parameters can also be obtained through [an option in router advertisements](#).

Here the DHCPv6 server does not track any state information: stateless DHCPv6 servers fill a different purpose as one part of the stateless auto-configuration process.

The **stateless auto-configuration process** for a global unicast address is detailed below:

- **Step 1: IPv6 Neighbor Discovery Protocol (NDP)**, particularly the Router Solicitation (RS) and Router Advertisement (RA) messages, learn the prefix, prefix length, and default router.

Example NDP RS/RA Process to Find the Prefix/len and Default Router



The RS message uses:

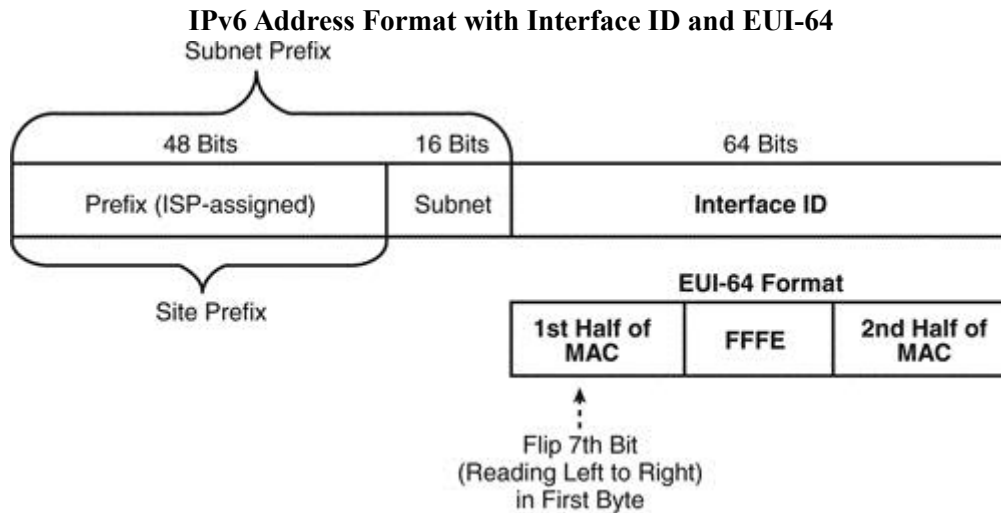
- Source IPv6: FE80::EUI-64 : IPv6 link local of host
- Destination IPv6: FF02::2 : all routers on the link multicast

The RA message uses:

- Source IPv6: FE80::EUI-64 : IPv6 link local of router
- Destination IPv6: FF02::1 : all nodes on the link multicast

- *NDP RA messages can be also used to achieve sub-second failover time (minimum RA interval is 30 ms) for High-Availability to replace a FHRP solution, but [should be avoided on MLAG \(Multi-Chassis Link Aggregation\) links](#), otherwise the ISL (Inter-Switch Link) might become overloaded.*

- **Step 2:** Some math to derive the interface ID (host ID) portion of the IPv6 address, using a format called EUI-64.



- Routers normally form a serial interface's link local IPv6 address using EUI-64 rules based on the MAC address of the first LAN interface on the router.
- **Step 3: Stateless DHCPv6 or RA option** ([RFC 6106](#)) to learn the *DNS servers IPv6 addresses* and DHCPv6 for *NTP servers IPv6 addresses*
- **Step 4: Optional DNS registration**

7.3.2.5.2 Static Configuration

- For one option, you configure the entire 128-bit IPv6 address,
- for the other, you just configure the 64-bit prefix and tell the device to use an EUI-64 calculation for the interface ID portion of the address.

7.3.2.5.3 Privacy Extensions Address

From "[IPv6 Security Brief, Cisco](#)"

In order to prevent the potential tracking of a host by identifying the host with its static EUI-64 Interface Identifier (IID) part of his IPv6 address, a host can use the privacy extension, which uses a random number as the IID. With privacy extension addresses, a host cannot be tracked any more as the address changes over time.

This is fine for a residential user but not acceptable for hosts inside a managed organization: the security/network operators must be able to track a malicious or misconfigured host within their network. It is recommended to disable privacy extensions within a managed network.

7.3.2.6 On-Link Determination

[RFC 5942](#) defines "The Relationship between Links and Subnet Prefixes" in IPv6 subnet model.

The behavior of IPV6 is quite different from IPV4 where the prefix/netmask are used for on-link determination:

- *A host can have IPv6 addresses without any related on-link prefixes or can have on-link prefixes that are not related to any IPv6 addresses that are assigned to the host.*
- *The on-link determination is separate from the address assignment.*

Instead, IPv6 devices use a 'Prefix List' of on-link addresses which is built from different sources:

- *All their link-local prefixes*
- *Valid Router Advertisements (RA) that specify a prefix with the L-bit (on-Link) set.*
- *Manual configuration*
- *Redirects from the default gateway to a local host*

All other addresses are considered off-link; traffic to hosts not in the "Prefix List" is automatically sent to the default gateway for forwarding. If the destination happens to be on the same link, the router forwards the packet to the destination and sends a Redirect to the sender.

7.3.3 Anycast Addresses

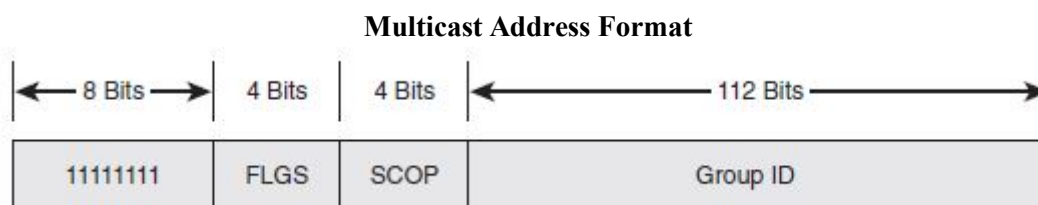
This *destination-only* address type allows the implementation of a nearest server among duplicate servers concept. This design choice allows servers that support the exact same function to use the exact same unicast IP address. The routers then forward a packet destined for such an address **to the nearest server** —as defined by the routing protocols in use.

The Subnet-Router anycast address: syntactically the same as a unicast address for an interface on the link with the interface identifier set to zero, i.e subnet-prefix::n.

It is intended to be used for applications where a node needs to communicate with any one of the set of routers on the subnet.

7.3.4 Multicast Addresses

Like IPv4, these addresses represent a dynamic group of hosts, allowing a host to send one packet that is then delivered to every host in the multicast group. IPv6 defines some special-purpose multicast addresses for overhead functions (such as NDP). IPv6 also defines ranges of multicast addresses for application use.



All IPv6 multicast addresses begin with **FF::/8**, the fourth hex digit **SCOP** identifies the scope:

- **FF02::/16**: multicasts with a **link local scope**,
- **FF05::/16**: multicasts with a **site local scope**,
- **FF08::/16**: multicasts with an **organization local scope**,
- **FF0E::/16**: multicasts with a **global scope**.

The **FLGS** (flags) field consists of three leading 0s followed by a **T bit**: 000T. If T = 0, the address is a well-known multicast address assigned by the global IANA. If T = 1, the address is not a permanently assigned address.

Purpose	IPv6 Address	IPv4 Equivalent
All IPv6 nodes on the link	FF02::1	subnet broadcast address
All IPv6 routers on the link	FF02::2	N/A
OSPF messages	FF02::5, FF02::6	224.0.0.5, 224.0.0.6
RIP-2 messages	FF02::9	224.0.0.9
EIGRP messages	FF02::A	224.0.0.10
Mobile agents	FF02::B	224.0.0.11
All PIM routers	FF02::D	224.0.0.13
DHCP relay agents (routers that forward to the DHCP server)	FF02::1:2	N/A
DHCP servers (site scope)	FF05::1:3	N/A
All NTP servers (site scope)	FF05::101	N/A

Multicast Groups Joined by IPv6 Router Interfaces:

- FF02::1 represents **all IPv6 devices**, so router interfaces must listen for packets sent to this address.
- FF02::2 represents **all IPv6 routers**, so again, router interfaces must listen for packets sent to this address.
- FF02::5/6/9/A, depending on the routing protocol(s) running on the interface,
- **FF02::1:FFxx:xxxx** is the range for an address **solicited node multicast** address (with xx:xxxx matching the last 3 bytes of its IPv6 addresses, one per IPv6 address), used by several functions, including the **Neighbor Discovery (ND)** and **Duplicate Address Detection (DAD)**.

7.4 ICMPv6

7.4.1 Messages Types

- **Informational** messages.
 - Echo request
 - Echo reply
- Some **Error** messages
 - Destination unreachable
 - No route to destination
 - Destination administratively prohibited
 - Address unreachable
 - Port unreachable
 - Packet too big
 - Time exceeded
 - Parameter problem

Other IPv6 mechanisms use ICMPv6 to determine **neighbor availability, path MTU, destination address or port reachability**.

7.4.2 Path MTU Discovery

IPv6 **does not allow packet fragmentation throughout the internetwork**. Only sending hosts are allowed to fragment. Routers are not allowed to fragment packets. [IPv6 RFC 2460](#) specifies that the MTU of every link in an IPv6 must be **1280 bytes or greater**. [Path MTU discovery RFC 1981](#) recommends that nodes should implement IPv6 path MTU discovery to determine whether any paths are greater than 1280 bytes. ICMPv6 packet-too-big error messages determine the path MTU. Nodes along the path send the ICMPv6 packet-too-big message to the sending host if the packet is larger than the outgoing interface MTU.

7.5 NDP

Hosts use **Neighbor Discovery Protocol** to implement plug-and-play functions that:

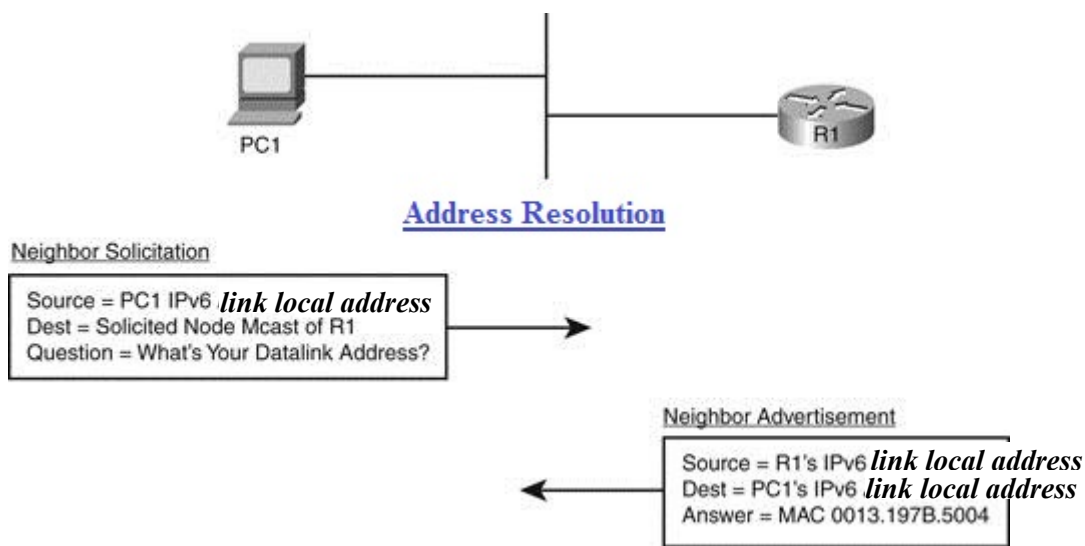
- discover all other nodes in the same link,
- check for duplicate addresses,
- find routers in the link,
- search for alternative routers if the primary fails.

The IPv6 NDP protocol performs the following functions:

- **Stateless address auto-configuration**: The host can determine its full IPv6 address without the use of DHCP.
 - **Prefix discovery**: The host finds out the link IPv6 prefix/len.
 - **Router discovery**: The host finds local routers without the use of DHCP.
 - **Parameter discovery**: The host finds out the link MTU and hop count.
- **Duplicate Address Detection**: The host can determine whether the address it will use is already in use on the network.
- **Address mapping resolution**: The host can determine the MAC address of other nodes without the use of ARP.
- **Neighbor unreachability detection**: The host can determine whether a neighbor is no longer reachable.
- **Redirect**: The host can tell another host if a preferred next hop exists to reach a particular destination.

IPv6 **NDP uses ICMPv6** to implement some of its functions. These ICMPv6 messages are:

- **Router Solicitation (RS)**: Sent by hosts to request RA from local routers
- **Router Advertisement (RA)**: Sent by routers to advertise their presence and link specific parameters
- **Neighbor Solicitation (NS)**: Sent by hosts to request link layer addresses of other hosts (also used for duplicate address detection)
- **Neighbor Advertisement (NA)**: Sent by hosts in response to an NS
- **Redirect**: Sent to a host to notify it of a better next hop to a destination



The NS message uses:

- Source IPv6: FE80::EUI-64 : IPv6 link local of sender
- Destination IPv6: FF02::1:FFxx:xxxx: solicited node multicast address
- ↑
- xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx: solicited node IPv6 unicast address or link local address (whose MAC address is unknown)
- ↓
- Destination MAC: 0100:5Exx:xxxx LAN multicast address (with first left high-order bit =0)

All IPv6 listen for frames sent to their own solicited node multicast address, so that when a host or router receives such a multicast, the host can realize that it should reply.

The NA message uses:

- Source IPv6: FE80::EUI-64 : IPv6 link local of receiver
- Destination IPv6: FE80::EUI-64 : IPv6 link local of previous sender

- *Unsolicited NA is the equivalent of the gratuitous ARP in IPv4 world which is used to refresh ARP caches.*
- *However, if for example IPv6 routers periodically send unsolicited NA, IPv6 end stations are still required to send unicast NS messages to their default gateway router because IPv6 requires bidirectional path validation.*

7.6 DAD

➤ Duplicate Address Detection

To check its own IPv6 address(es), a host sends a NS to the solicited node multicast address based on its *own* IPv6 address:

- Source IPv6: ::/128 : IPv6 unspecified address
- Destination IPv6: FF02::1:FFxx:xxxx: own solicited node multicast address

If some host sends a reply with a NA, listing the same IPv6 address as the source address, the original host has found that a duplicate address exists:

- Source IPv6: IPv6 address: IPv6 unicast of the receiver and **probably** of the original sender
- Destination IPv6 : FF02::1: IPv6 all nodes multicast address

7.7 IND

➤ Inverse Neighbor Discovery

The details inside the INS message include the following:

- Source IPv6: IPv6 address: IPv6 unicast of sender
- Destination IPv6: FF02::1: all IPv6 hosts multicast
- Known destination link layer address
- Request: Please reply with your IPv6 address(es)

The INA reply lists *all the IPv6 addresses* corresponding to the data link layer address.

<https://t.me/learningnets>

7.8 Migrating from IPv4 → IPv6

The three **major transition mechanisms** are:

- **Dual-stack** (IPv4 and IPv6 coexist in hosts and networks.)
- **Tunneling** (IPv6 packets are encapsulated into IPv4 packets or IPv4 packets are encapsulated into IPv6 packets)
- **Translation** (IPv6 packets are translated to IPv4 packets or IPv4 packets are translated to IPv6 packets.)

IPv6 **deployment models** are also divided into three major categories:

- **Dual-stack model** (IPv4 and IPv6 coexist on hosts and network.)
- **Hybrid model** (combination of Intra-Site Automatic Tunneling Protocol (ISATAP) *or* Manually Configured Tunnels (MCT) and dual-stack mechanisms)
- **Service block model** (combination of ISATAP *and* MCT and dual-stack mechanisms)

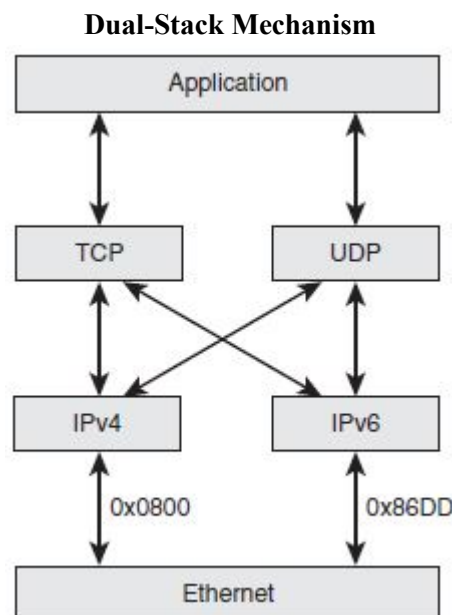
7.8.1 IPv4/IPv6 Dual Stacks

The hosts using dual stacks follow the same general process of using DNS to resolve a name into an IP address. The DNS requests can return either an IPv4 address (A record) or an IPv6 address (AAAA record). The dual stack host can then choose to use IPv4 to communicate with an IPv4 host or IPv6 to communicate with an IPv6 host.

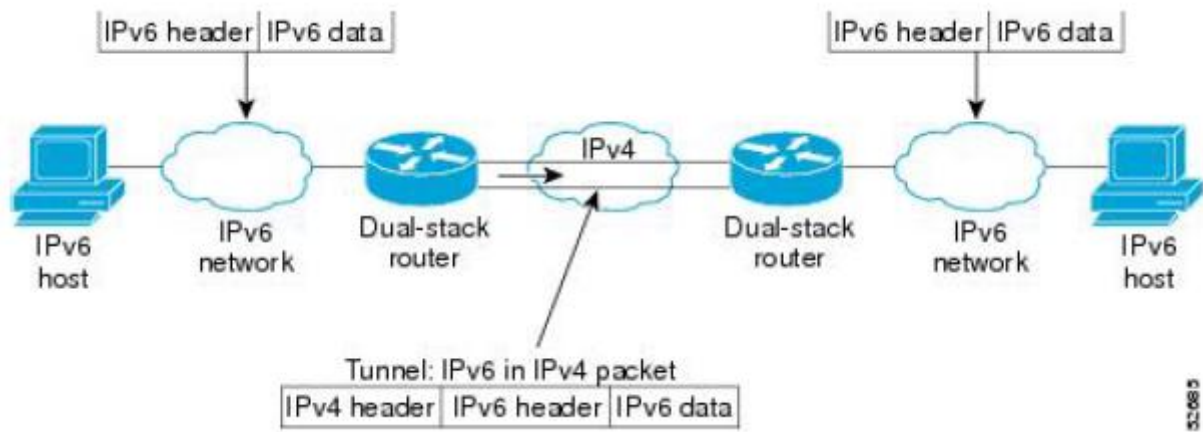
When a frame is received, the **Ethernet type code** identifies whether the packet needs to be forwarded to IPv4 (0x0800) or IPv6 (0x86DD).

To support dual stack hosts, routers need to forward both IPv4 and IPv6 packets. To forward IPv6 packets to the various destinations, the design engineer can use one of two general approaches:

- **Native IPv6:** Configure IPv6 on most or all routers, on most or all production interfaces, making all routers use a dual stack.
- **IPv6 tunnels:** Configure some routers with IPv6, others without IPv6, and tunnel the IPv6 packets over the IPv4 network by encapsulating IPv6 packets inside IPv4 packets.



7.8.2 Tunneling IPv6 Over IPv4: IPv6 → IPv4



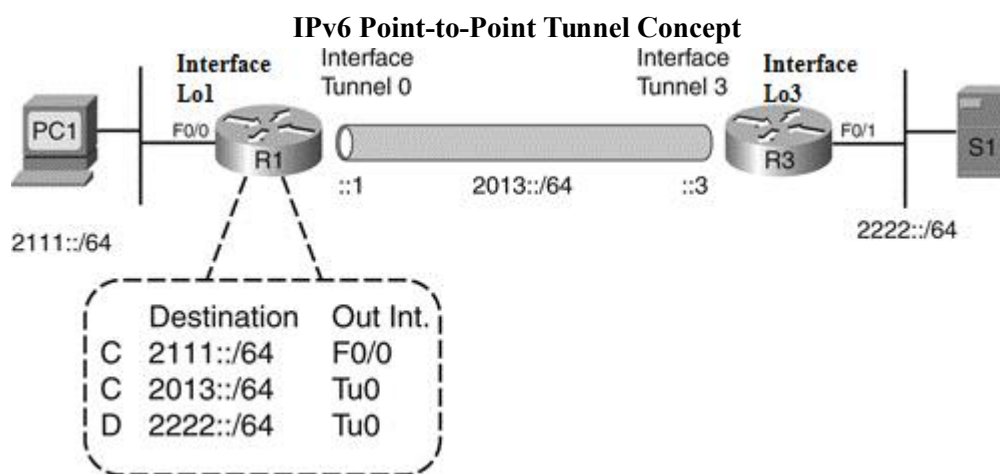
- For **statically** configured tunnels, the **tunnels are configured with IPv4 and IPv6 addresses for tunnel source and destination**. Tunnels can be built between border routers or between routers and hosts.
- In **semi-automatic** configured tunnels, a tunnel broker is used. The tunnel broker is a server on the IPv4 network that receives requests from dual-stack clients and builds a tunnel on the tunnel dual-stack router and associates it with the client.
- **Automatic** (dynamic) tunnel mechanisms are:
 - ISATAP
 - 6to4
 - 6over4
 - 6RD
 - 6PE
 - **LISP**

7.8.2.1 Static Point-to-Point Tunnels

Static Point-to-Point Tunnels	PT	Tunnel IPv6 link local prefix*	Embedded Source IPv4 address	RFCs
6in4 (Manually Configured Tunnels)	41	FE80::/96	Quartets 7/8	4213
GRE (Generic Routing Encapsulation)	47	FE80::/64 (EUI-64)	No	2784

*:

- The tunnel interface has not necessarily statically configured IPv6 address and has an automatically generated link local IPv6 address,
- For IPv6 IGP multicast traffic,
- For IPv6 routing out on the Tunnel interface.
- For 6in4 (*also called MCT*), the router forms the link local address with a FE80::/96 prefix and then adds the 32-bit source IPv4 address as the last 32 bits. The Protocol Type (PT) in IPv4 header is 41 (IPv6). One of the beauties of this MCT (6in4) is that they are supported by most of the stacks and routers that you might run into within production networks today.
- For GRE, it works as if the tunnel interface was a serial interface, deriving the interface ID using EUI-64 rules and the MAC address of the first LAN interface on the router. The Protocol Type in IPv4 header is 47 (GRE).



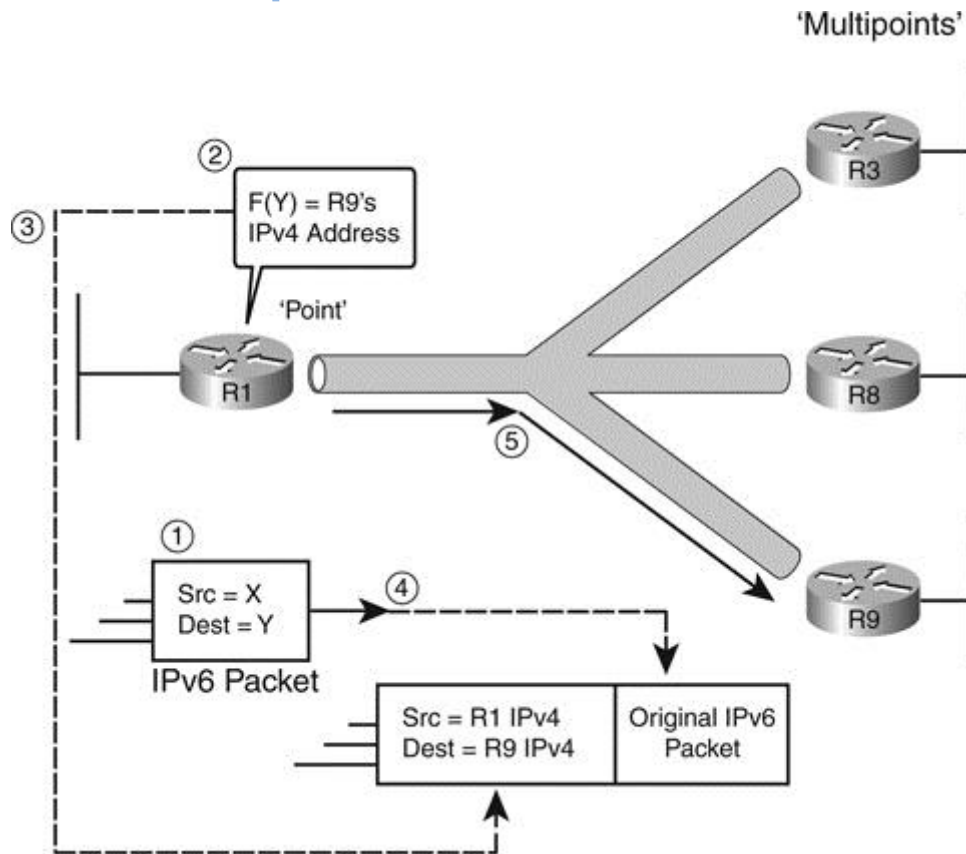
6in4 and GRE tunnels have many *similarities*, including the configuration:

- They both create a **virtual point-to-point link** between two IPv6/IPv4 routers for the purpose of supporting the forwarding of IPv6 packets.
- **IPv6 IGP routing protocols can run over these virtual links.** The routers will:
 - assign link local addresses on these links,
 - allow the forwarding of IPv6 multicast traffic.
- Both types allow the configuration of **additional security features** over the tunnel interfaces.
- Finally, both require static configuration of both the **tunnel source and the tunnel destination IPv4 addresses**.
- **Defining a specific unicast IPv6 address on the Tunnel interface is not absolutely necessary: a link-local IPv6 address is enough!**

To the depth used for these topics in this summary, these two tunneling options have only minor *differences*:

- **6in4** simply conforms to the rules about generic IPv6 tunnels outlined in [RFC 4213](#), which defines how to encapsulate IPv6 inside an IPv4 packet, without additional headers.
- When a *multicast* needs to be sent out a 6in4's interface, the router encapsulates the IPv6 multicast and forwards it inside a unicast IPv4 packet to the other end of the tunnel.
- **GRE** tunnels use a generic encapsulation originally defined by Cisco and later standardized in [RFC 2784](#). GRE uses an **additional stub header** between the IPv4 and IPv6 header; this extra header includes a protocol type field, which allows a GRE tunnel to **carry many passenger protocols**. (The *passenger protocol* is the protocol encapsulated inside another protocol's header.)
- **GRE** also **supports several transport protocols** (*transport protocols* encapsulate the passenger protocol), although IPv4 typically is the only transport protocol used for tunneling IPv6.
- GRE's flexibility allows a single GRE tunnel to carry IPv6 plus other traffic as well, whereas 6in4 cannot.

7.8.2.2 Dynamic Point-to-Multipoint Tunnels

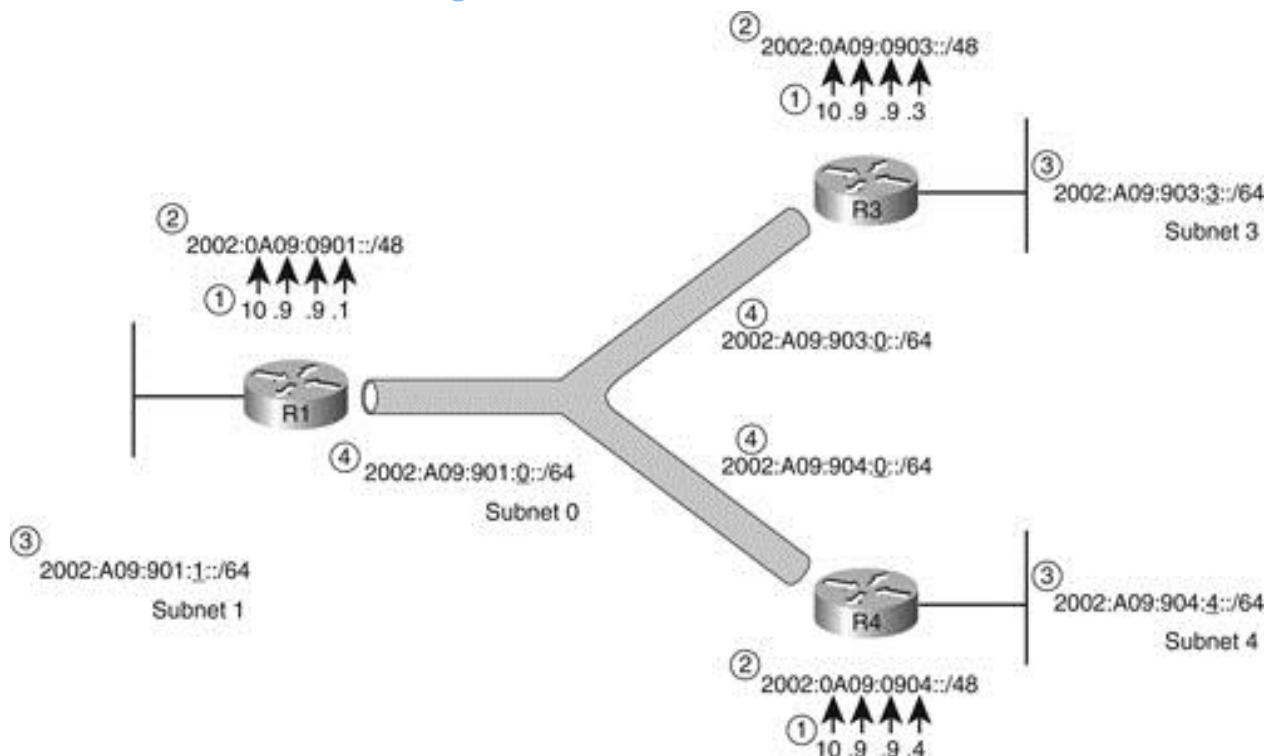


<i>Dynamic Point-to-Multipoint Tunnels</i>	<i>Tunnel IPv6 prefixes</i>	<i>Embedded IPv4 address</i>	<i>RFC</i>
<i>6to4</i>	<i>FE80::/96</i> <i>2002::/16</i>	<i>Quartets 7/8</i> <i>Quartets 2/3</i>	<i>3056 (NAT compatible)</i>
<i>ISATAP</i> (Intra-Site Automatic Tunnel Addressing Protocol)	<i>FE80:0:0:0:5EFE::/96</i> <i>Global Unicast:0:5EFE::/96</i> (Modified EUI-64 rules)	<i>Quartets 7/8</i> <i>Quartets 7/8</i>	<i>4214 (collides with NAT)</i> <i>6964 (operation)</i>
<i>6RD</i> (Rapid Deployment)	<i>FE80::/64 (EUI-64)</i> <i>ISP specific::/32</i>	<i>Quartets 3/4</i>	<i>5969 (NAT compatible)</i> (between CPEs and ISPs' Internet routers)
<i>Teredo</i> (Tunneling IPv6 over UDP through NAT)	<i>2001::/32</i>	<i>Quartets 7/8</i>	<i>4380 (NAT compatible)</i> (between hosts)

➤ *Protocol Types in IPv4 header: 41 (IPv6)*

7.8.2.2.1 6to4 Tunnels

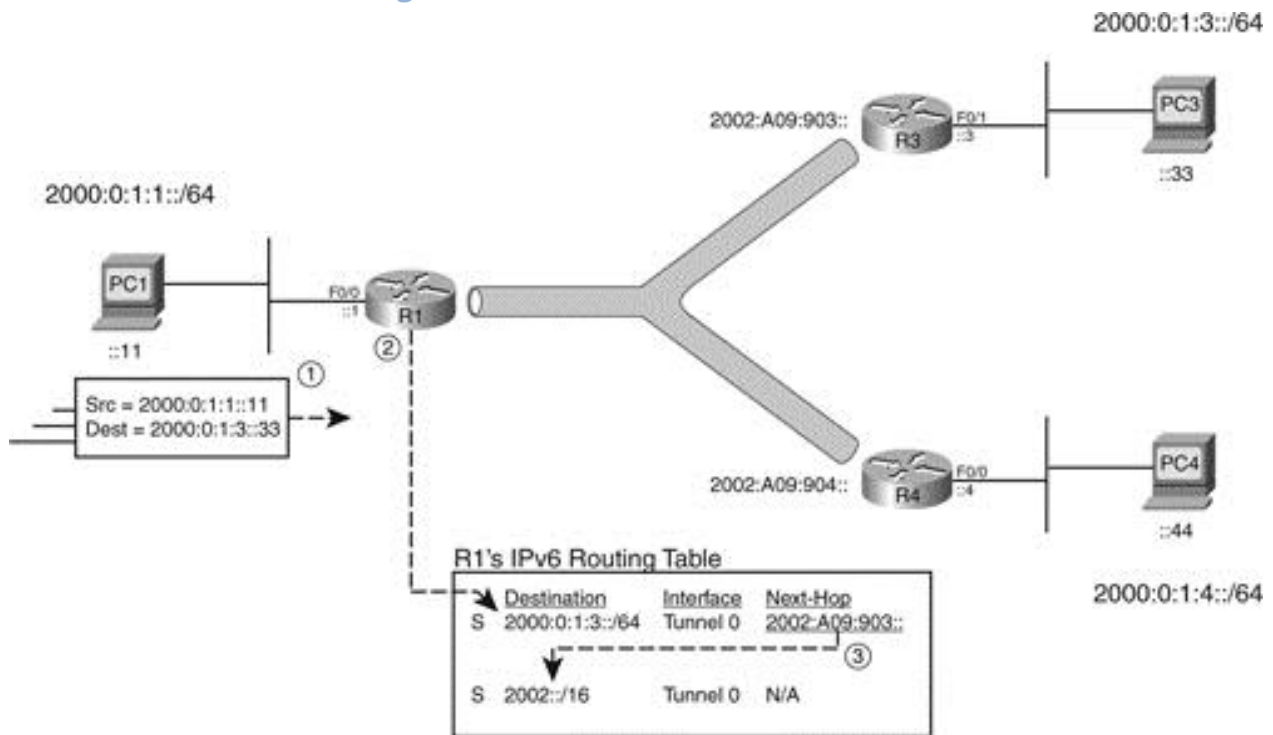
7.8.2.2.1.1 'Reserved Mode': Using 2002::/16 in the LANs



[RFC 3056](#) specifies the 6to4 method for transition by assigning an interim unique IPv6 prefix. **2002::/16** is the assigned range for 6to4. Each 6to4 site uses a /48 prefix that is concatenated with 2002. The border router extracts the **IPv4 address that is embedded in the IPv6 destination address** and encapsulates the IPv6 packet in an IPv4 packet with the extracted destination IPv4 address. The destination router extracts the IPv6 packet and forwards it to the IPv6 destination.

- *The IPv6 unicast 'reserved' address for the tunnel interface in this situation is not absolutely necessary: the link local address is enough!*
- *The destination IPv4 information comes from the IPv6 header of the packets sent by the hosts.*
- *If the IPv4 address is public, then the encapsulated IPv6 traffic could go through Internet, despite the fact that IPv6 addresses are not public global unicast. But when migrating to public IPv6 addresses, all the 6to4 reserved addresses will need to be changed.*

7.8.2.2.1.2 'Public Mode': Using IPv6 Global Unicast Addresses in the LANs



- When the router matches a route with an outgoing 6to4 tunnel interface, but *with no next-hop IPv6 address*, AND this route was matched due to route recursion, then derive the tunnel's destination IPv4 address based on the previously matched route's next-hop IPv6 address.

Summarizing, the *differences* in planning and configuration for using *global unicast*s with *automatic 6to4 tunnels* are:

- **Step 1:** Plan the prefixes and addresses for the LANs using the global unicast range assigned to the Enterprise.
- **Step 2:** Configure an *additional static route* for each remote subnet, configuring the tunnel as outgoing interface **and** configuring the next-hop **reserved** IPv6 address. That next-hop must be the remote router tunnel IPv6 address, which embeds the destination IPv4 address as the second and third bytes.

- *The IPv6 unicast 'reserved' address for the tunnel interface in this situation is necessary!*
- *The destination IPv4 information here comes from the IPv6 routing table, and not from the packets.*

Note : You can also use BGP for IPv6 to learn the route listed in Step 2. You cannot use EIGRP nor OSPF for that purpose because the IPv6 routers do not share a common subnet in the tunnel.

7.8.2.2.2 6over4

6over4 is another tunnel method that **requires an IPv4 multicast-enabled network**. IPv6 multicast packets get encapsulated into IPv4 multicast packets to communicate with other 6over4 hosts. 6over4 is of limited practical use.

7.8.2.2.3 ISATAP Tunnels

The items that *are similar* between ISATAP tunnels and automatic 6to4 tunnels **in 'Public mode'** are :

- ISATAP uses *IPv6 global unicast prefixes for user subnets*.
- ISATAP tunnel interfaces use *IPv6 addresses that embed the tunnel destination IPv4 address*.
- The routers *need static (or BGP) routes for the destination end-user IPv6 prefixes*; the route must list a next-hop IPv6 address, which in turn embeds the tunnel destination IPv4 address.
- ISATAP *does not support IPv6 multicast traffic*; so IGP's *cannot run over the tunnel* despite the fact that the routers share a single IPv6 subnet.

The items that *differ* between ISATAP tunnels and automatic 6to4 tunnels are :

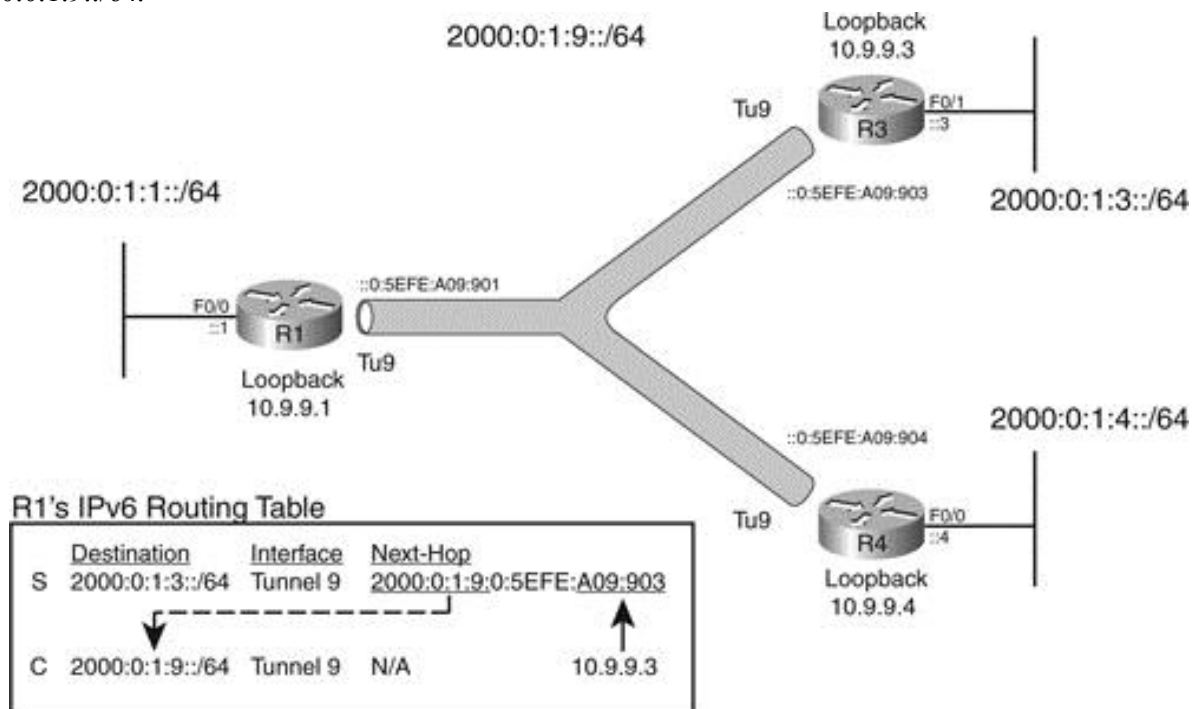
- ISATAP tunnel interface IPv6 addresses embed the IPv4 address in the *last two quartets*.
- ISATAP tunnels can *automatically* derive the tunnel interface IPv6 interface ID by using *modified EUI-64 rules*.
- ISATAP tunnels do not use a special reserved range of IPv6 addresses at all, instead using just normal IPv6 unicast prefixes.
- ISATAP tunnels typically use *a single prefix to which all tunnel interfaces connect*, so all routers have a connected IPv6 route to that same subnet.

➤ **The destination IPv4 information here also comes from the IPv6 routing table.**

➤ Although the ISATAP tunneling mechanism is designed for transporting IPv6 packets *within* a site, *not between* sites, **we could use public IPv4 addresses for encapsulation and the Internet to interconnect all these ISATAP sites.**

The figure is the same as previous Figure's automatic 6to4 tunnel example with the same user IPv6 prefixes from the global unicast range. However, the design and configuration has been changed to work with ISATAP tunnels, as follows:

- The three tunnel interfaces now have IPv6 addresses in *common IPv6 subnet* 2000:0:1:9::/64. (The actual subnet number is not important; just choose a currently unused IPv6 subnet.)
- The tunnel interfaces' IPv6 addresses conform to modified EUI-64 rules (explained following the figure) embedding the IPv4 address in the last two quartets.
- The routers *no longer need a route for 2002::/16*, instead relying on the connected route created for subnet 2000:0:1:9::/64.



The only other item that needs some explanation is the *format of the ISATAP-defined IPv6 address*. The addresses can be *configured manually* but can also be *derived by the router using modified EUI-64 rules*. The rules work as follows:

- Configure a 64-bit prefix on the tunnel interface, and use the **eui-64** parameter, telling the router to derive the second half (interface ID) of the address.
- The router adds 0000:5EFE as quartets 5 and 6.
- The router finds the *tunnel's source IPv4 address*, converts it to hex, and adds that as quartets 7 and 8.

7.8.2.2.4 6RD Tunnels

Out of CCNP ROUTE scope.

- An ISP could use 6RD as an IPv4 → IPv6 migration tool with the following benefits:
 - It can be used on a large scale between customer CPEs (STBs) and ISPs' Internet routers,
 - No need to change all customer's IPv6 addresses later when the whole ISP infrastructure becomes dual-stacked or based solely on IPv6,
 - Immediate IPv6 intra-ISP and Internet connectivity: the subscriber gets native dual-stack connectivity,
- Originally designed by the French broadband 'Free' ISP because their DSLAMs did not support IPv6.

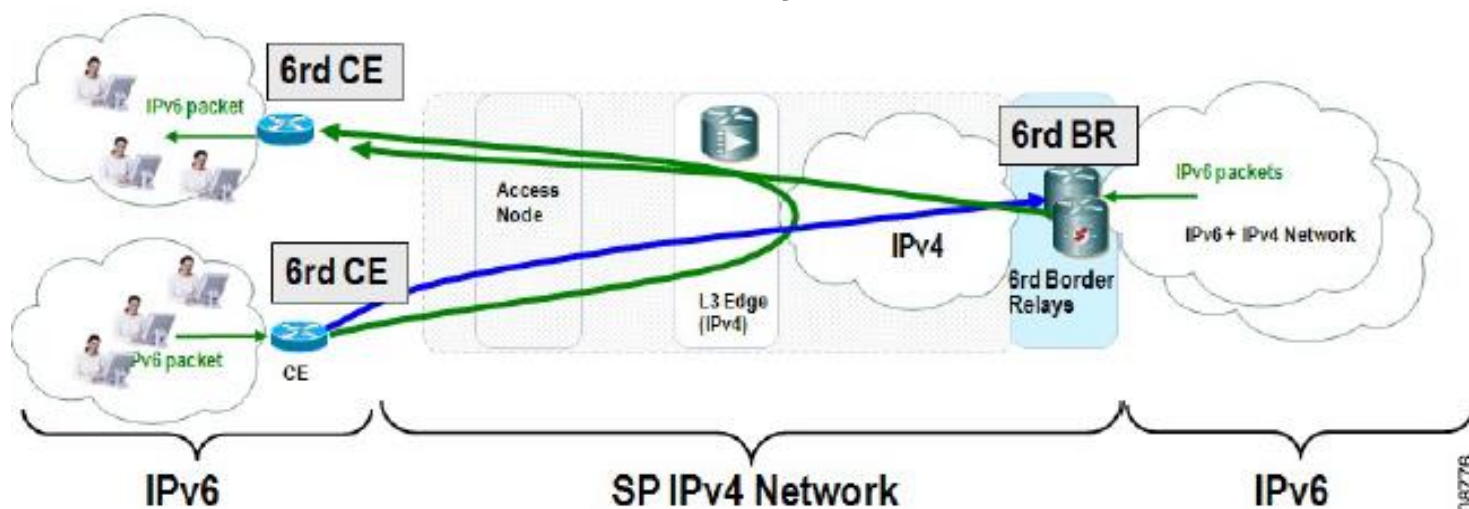
From [IOS XE 2 IPv6 Configuration Guide, Cisco](#):

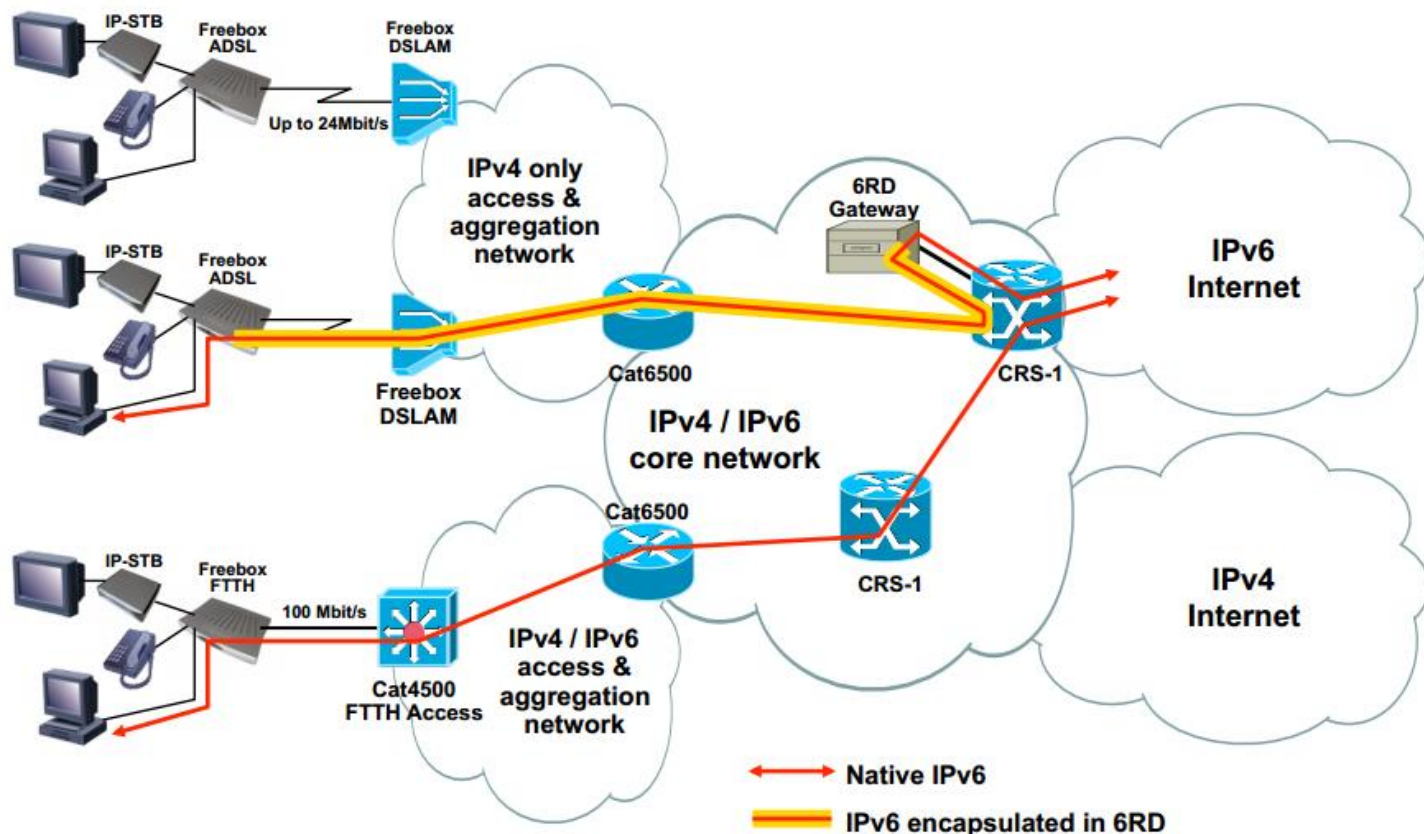
The 6RD feature allows a service provider (SP) to provide a unicast IPv6 service to customers over its IPv4 only access network by using encapsulation of IPv6 in IPv4. The main differences between 6RD and 6to4 tunneling are as follows:

- 6RD does not require addresses to have a 2002::/16 prefix; therefore, the prefix can be from the SP's own address block. This function allows the 6RD operational domain to be within the SP network.
- From the perspective of customer sites and the general IPv6 internet connected to a 6RD-enabled SP network, the IPv6 service provided is equivalent to native IPv6.
- All 32 bits of the IPv4 destination need not be carried in the IPv6 payload header. The IPv4 destination is obtained from a combination of bits in the payload header and information on the router. Furthermore, the IPv4 address is not at a fixed location in the IPv6 header as it is in 6to4.

The 6RD SP prefix was selected by the SP for the IPv6 deployment shown in the figure below. The 6RD delegated prefix is derived from the SP prefix and the IPv4 address bits, and is used by the CE for hosts within its site.

6RD Deployment



Free 6RD Implementation

05/05/2009

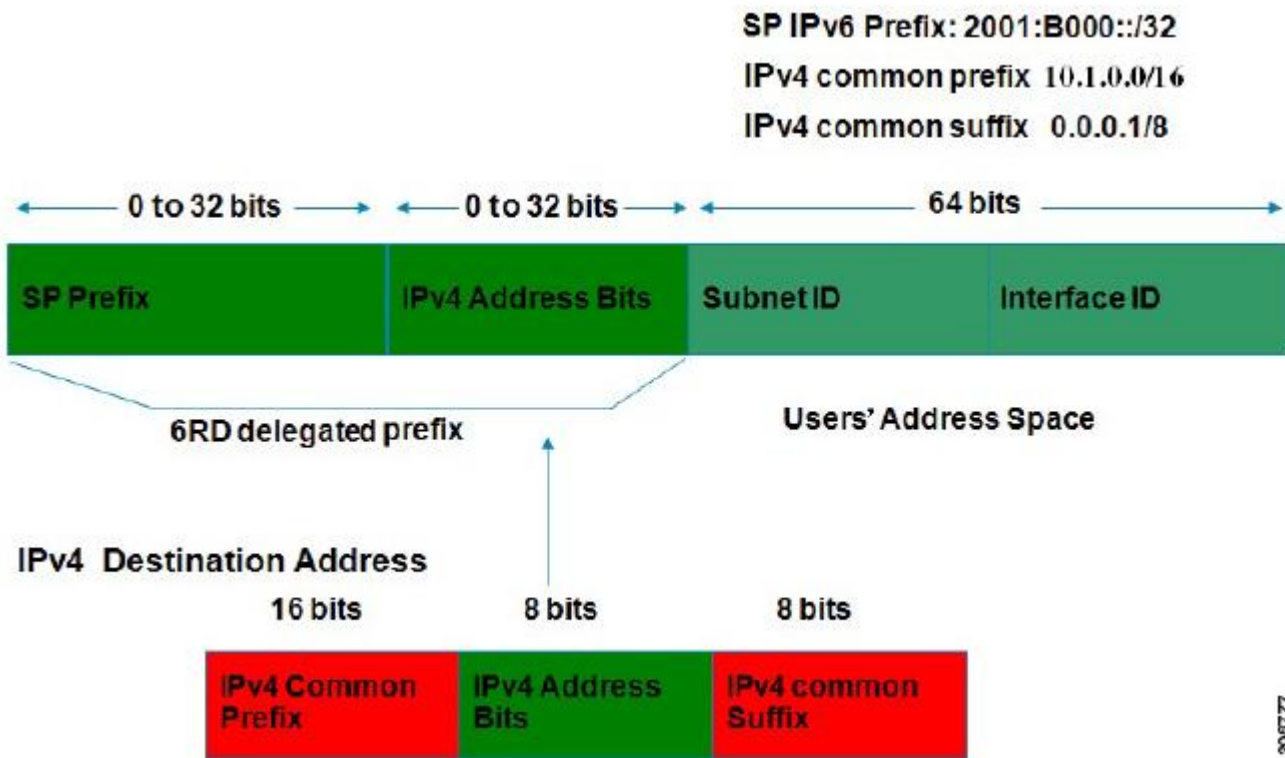
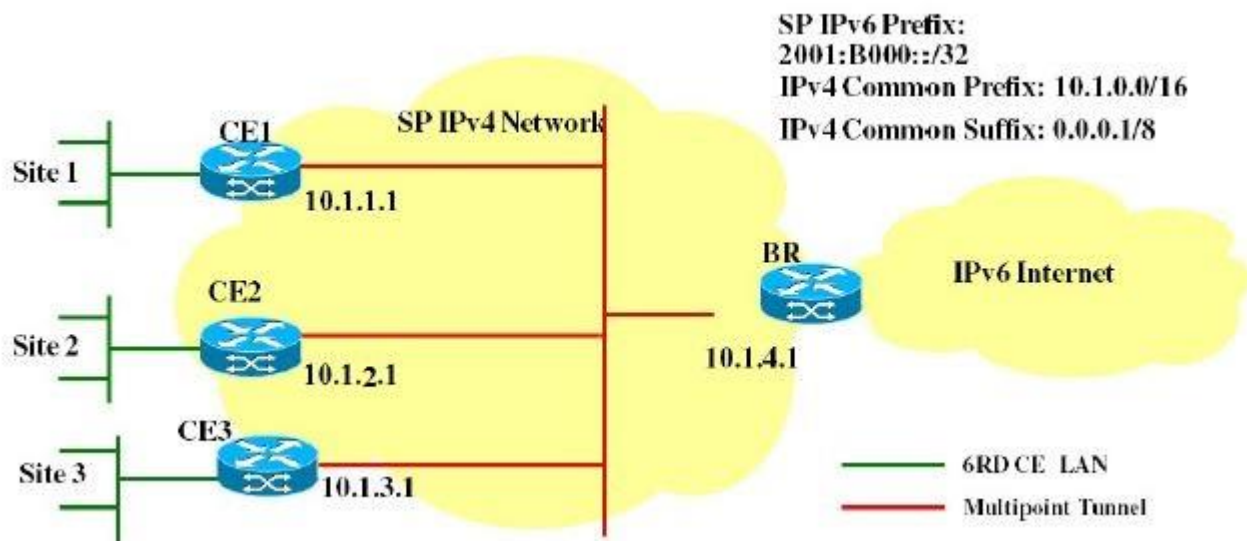
IPv6 @ Free

10

6RD Improvements over 6to4:

- **Uses PA address space, directly under SP control**
- **Configurable delegated prefix length**
- **High-order and low-order IPv4 common bits can be ignored, raising the number of IPv6 networks available to customers**
- **Configurable Border Relay IPv4 address**
- **Possible to use one IPv4 BR address and load-balance the traffic between multiple BRs with GLBP**
- **DHCPv4 6DR option can be used to pass the parameters to CE routers**

Furthermore, 6RD solves first-hop L2 security IPv6 issues (such as RA spoofing) with IPv4 mechanisms.

6RD Prefix Delegation Explanation6RD Prefix Delegation and Explanation

SP Prefix	2001:B000::/32
IPv4 Common Prefix	10.1.0.0/16
IPv4 Common Suffix	0.0.0.1/8
CE1: Delegated 6RD prefix	2001:B000:0100::/40
CE2: Delegated 6RD prefix	2001:B000:0200::/40
BR: Delegated 6RD prefix	2001:B000:0400::/40
CE1 (IPv4) tunnel transport source	10.1.1.1
CE2 (IPv4) tunnel transport source	10.1.2.1
BR (IPv4) tunnel transport source	10.1.4.1

In this example, the 6RD Delegated Prefix is only 8 bits wide, which means only a maximum of 256 different customer sites. For a typical ISP, the IPv4 address bits field would be much wider.

7.8.2.2.5 6PE

Cisco 6PE (IPv6 Provider Edge Router over MPLS) allows IPv6 islands to communicate over an **MPLS/IPv4 core** network using MPLS label-switched paths (LSP). The Cisco 6PE routers are dual stacked. The method relies on BGP extensions in the IPv4 6PE routers to exchange IPv6 reachability information, along with an MPLS label for each IPv6 address prefix announced.

7.8.2.3 Comparing IPv6 Over IPv4 Tunnels

Method	Static or Dynamic	Topology	Advantages and Other Notes
Manually Configured	Static	Pt-pt	Acts like a virtual point-to-point link, supporting IPv6 IGPs. Good for more permanent tunnels. Supports IPv6 IGPs. Slightly less overhead than GRE.
GRE	Static	Pt-pt	Generic Routing Encapsulation. Same advantages as previous row, plus it can support other Layer 3 protocols over the same tunnel.
6to4	Dynamic	Mpt	It may require less configuration than all other types when adding a new site. Supports global unicasts, with some extra configuration. Uses second and third quartets to store IPv4 address.
ISATAP	Dynamic	Mpt	It easily supports global unicast addresses for all prefixes. Uses seventh and eighth quartets to store IPv4 address.
<i>6RD</i>	<i>Dynamic</i>	<i>Mpt</i>	<i>Very scalable option for ISPs providing native IPv6 service with global unicasts. Uses configurable 3rd and 4th quartets to store IPv4 address.</i>

Supports multiple transport and passenger protocols

- [RFC 7059](#) compares all IPv6-over-IPv4 tunneling mechanisms
- [RFC 7123](#) details the security implications of tunneling IPv6 over IPv4.

7.8.3 Tunneling IPv4 Over IPv6: IPv4 -> IPv4

Out of CCNP ROUTE scope.

These are modern IPv4 over IPv6 tunneling techniques used when there is an IPv6-only access/core network interconnecting IPv4 hosts and services.

Automatic (dynamic) tunnel mechanisms are:

- *DSTM*
- *DS-Lite*
- *Lightweight 4over6*
- *MAP-E*

7.8.3.1 DSTM

DSTM (Dual Stack Transition Mechanism) proposes the use of a dual stack that uses IPv4 addresses only when needed and the use of **IPv4 over IPv6 tunneling** to reach a destination IPv4 address. It is used when there is an *IPv6-only backbone* but an application needs to reach an IPv4 address.

7.8.3.2 DS-Lite

Dual-Stack Lite is described in [RFC 6333](#).

7.8.3.3 Lightweight 4over6

The current draft is available [here](#).

7.8.3.4 MAP-E

The current draft of “Mapping of Address and Port – Encapsulation” is available [here](#).

7.8.4 Translating IPv4 <-> IPv6

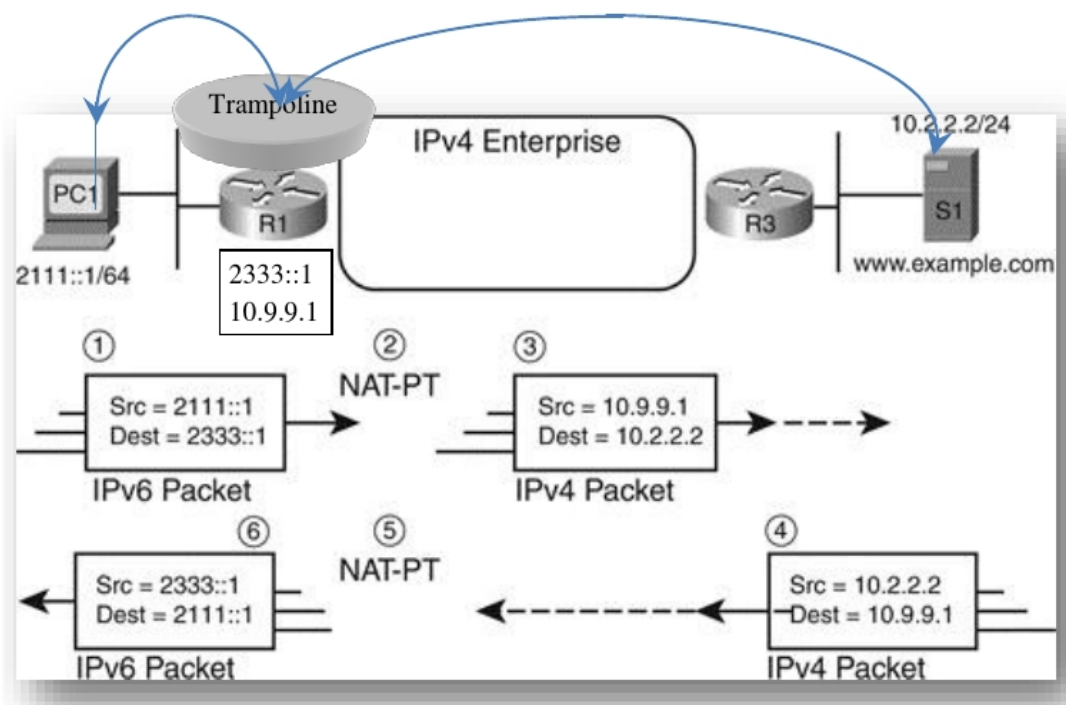
Translation is basically an extension to IPv4 NAT techniques. Some techniques are:

- **Application layer gateways (ALG):** These use dual stacks and allow one host on the IPv4 domain to communicate with the host on the IPv6 domain.
- **Application programming interfaces (API):** An API module intercepts IP traffic through an API and coverts it for the IPv6 counterpart.
- **Translation techniques:** Such as NAT-PT (or NAT46, *deprecated*), NAT64/DNS64, MAP-T, and SIIT (Stateless IP/ICMP Translation). The prefix **64:FF9B::/96** is used.

Framework for IPv4/IPv6 Translation: [RFC 6144](#).

7.8.4.1 NAT-PT

NAT-PT operates similarly to the NAT mechanisms to translate IPv4 private addresses to public address space. NAT-PT binds addresses in the IPv6 network to addresses in the IPv4 network and vice versa.



- For the process in previous figure to work, NAT-PT must also be *heavily involved in DNS flows* as well.
- *NAT-PT has been deprecated by [RFC 4966](#).*

7.8.4.2 NAT64/DNS64/SLB64: IPv6 -> IPv4

- *NAT64 is described in [RFC 6146](#).*
- *DNS64 can be used in conjunction with an IPv6/IPv4 translator such as NAT64 and is described in [RFC 6147](#).*
- *SLB64 (Server Load-Balancer) is described [here](#).*

7.8.4.3 MAP-T: IPv6 -> IPv4

The current draft of “Mapping of Address and Port – Translation” is [here](#).

7.8.4.4 NAT46/SLB46/SIIT: IPv4 -> IPv6

- *NAT46 for IPv6-only data center/campus*
- *SLB46 for IPv6-only servers*
- *Stateless IP/ICMP Translation is described in [RFC 6145](#).*

Cf. [SIIT Model](#).

➤ *The source IPv4 address is mapped into the source IPv6 address.*
<https://t.me/learningnets>

7.8.4.5 Issues

- SSL works fine over NAT64/SIIT but IPsec does not work over NAT64.
- *Multi-homing may also be a concern for the previous solutions.*
- *NAT64 / SLB64:*
 - *hide all IPv6 remote users behind a single IPv4 address which leads to several audit trail and security issues; the SLB is able to insert an HTTP X-Forward-For header which contains the remote IPv6 address. The latter feature allows for logging, and rate-limiting on the real servers based on the IPV6 address even if those servers run only IPv4.*
 - *break IPv4 URL literals*
 - *have performance issues*
- *NAT46 has issues that have been summarized in [RFC 4966](#).*

7.8.5 IPv6 Deployment Models

7.8.5.1 Dual-Stack Model

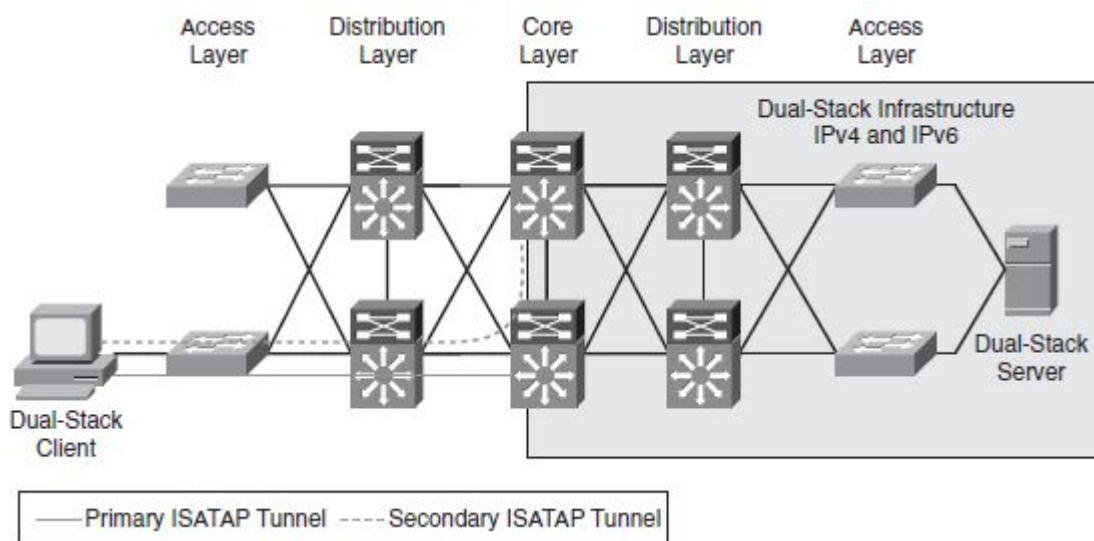
7.8.5.2 Hybrid Model

The transition mechanisms used are based on multiple network criteria such as number of hosts, IPv6 capable hardware, and location of IPv6 services. The hybrid model uses a combination of transition mechanisms:

- Dual-stack mechanism
- ISATAP
- Or MCT

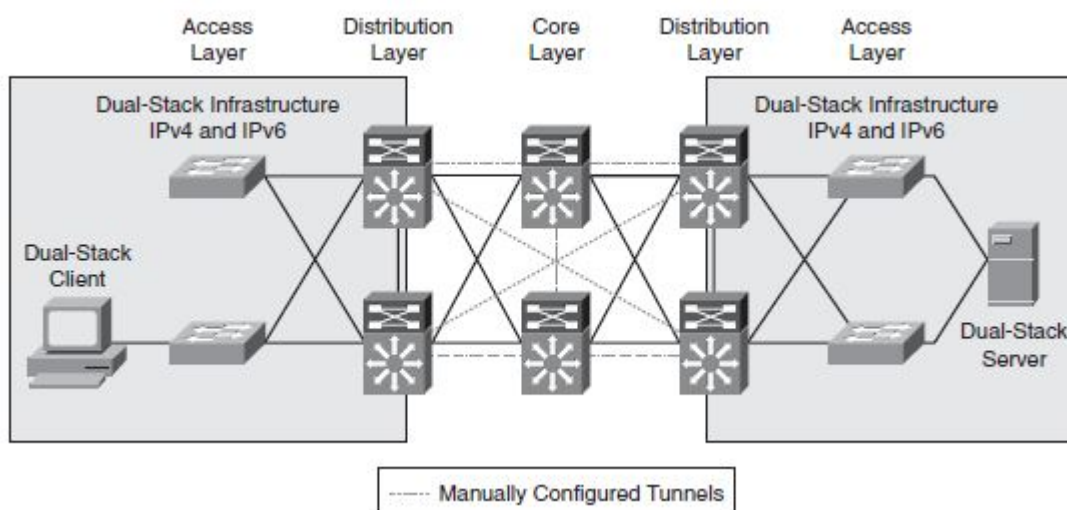
The hybrid model can be used **to tunnel a dual-stack host on an IPv4 access layer to an IPv6 dual-stacked core**. As shown in next figure, the dual-stack computer establishes an ISATAP tunnel to the core layer to access services from the dual-stack server on the right.

IPv6 Hybrid Model with ISATAP Tunnel



Another scenario is **to tunnel dual-stack distribution layers over an IPv4-only core**. As shown in next figure, the dual-stack computer on the left can access the dual-stack server on the right via the MCT. Multiple tunnels are configured to provide redundancy and load balancing.

IPv6 Hybrid Model with MCT

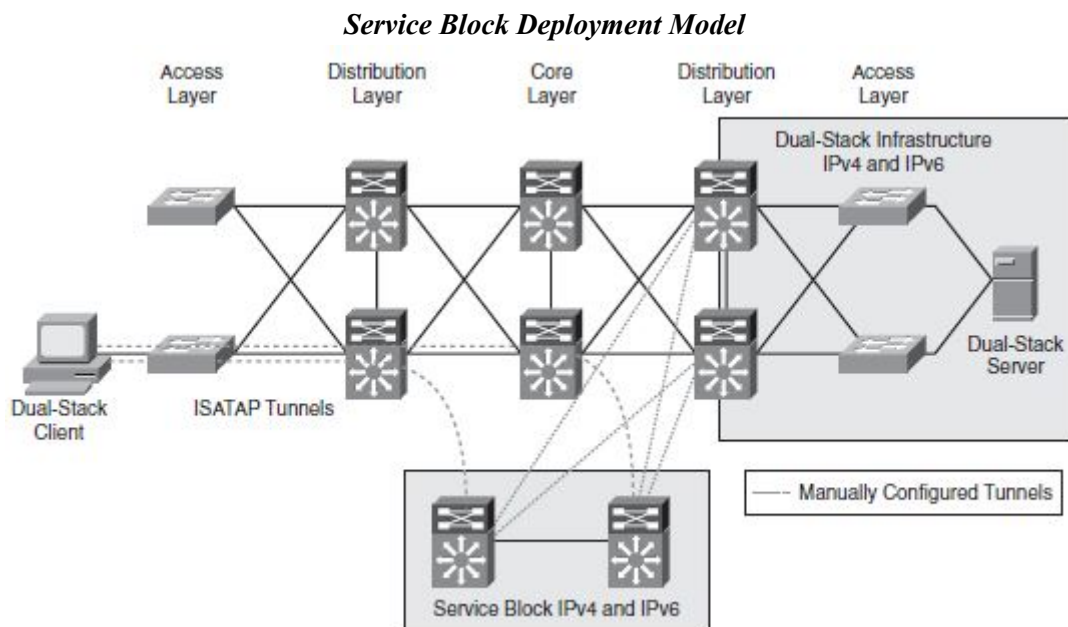


7.8.5.3 Service Block Model

In the service block model, a centralized layer that services dual-stack devices is created with tunnels manually configured between the distribution layer and the service block.

Dual-stack hosts also connect via ISATAP tunnels. In next figure, the dual-stack client on the left connects to the service block to establish connectivity with the dual-stack server on the right.

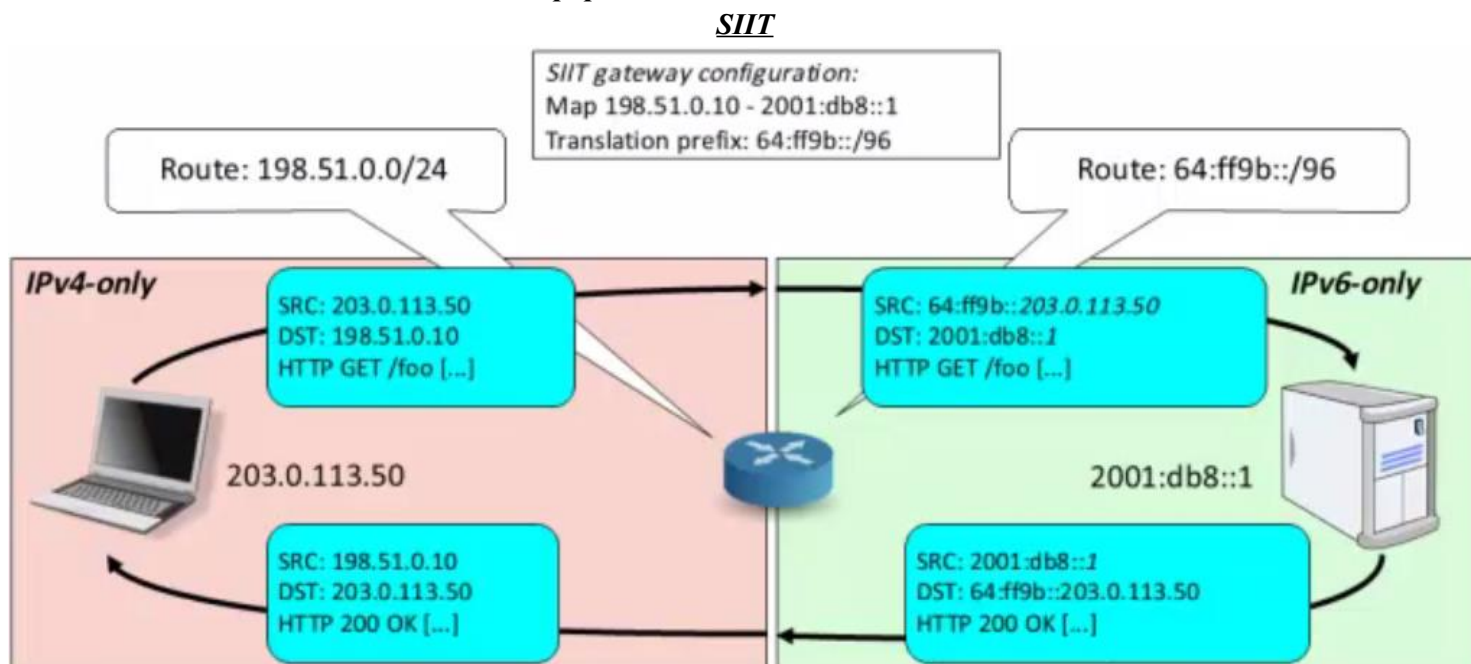
- *This means tunneling twice!!! Why not implementing a direct ISATAP tunnel from the dual-stack client to the dual-stack distribution layer on the right?!*



7.8.5.4 SIIT Model

Out of the scope of CCNP ROUTE:

- *This model involves IPv4-only hosts accessing IPv6-only services through a standard translation mechanism.*
- *The Data Center should migrate towards an IPv6-only network with SIIT ([RFC 6145](#)) on the edge to allow access to IPv4 only hosts: Stateless IP/ICMP Translation between IPv4 and IPv6 packet headers,*
- *More details [here](#), [here](#) and [here](#) on IpSpace:*



7.8.5.5 IPv6 Deployment Model Comparison

Description		IPv6 Deployment Model
All routers and hosts run IPv6 and IPv4.		Dual-stack model
Uses ISATAP or manually configured tunnels to allow dual-stack clients to connect to dual-stack servers over a IPv4 core.		Hybrid model
Uses ISATAP and manually configured tunnels to a service module.		Service block model
Uses an IPv6-only network with SIIT on the edge for IPv4-only hosts		SIIT model
IPv6 Deployment Model	Advantages	Disadvantages
Dual-stack model	Tunneling not required. Better processing performance. IPv4 & IPv6 independent routing, QoS, security, and multicast policies.	Network equipment upgrades. Double-protocol maintenance Complex troubleshooting
Hybrid model 1	Existing network can be leveraged with no upgrades.	IPv6 multicast not supported within ISATAP tunnels. Terminating ISATAP tunnels in core makes the core appear to be in IPv6 access layer.
Hybrid model 2	IPv4 and IPv6 have independent routing, QoS, security, and multicast policies.	Many static tunnels which makes it difficult to manage.
Server block model	Lesser impact on existing network. Flexible when controlling access to IPv6-enabled applications.	Large amounts of tunneling. Cost of additional equipment.
SIIT Model	Single routed protocol maintenance Simpler troubleshooting	

7.9 Routing Protocols

Routing Protocol	Full Name	RFC
RIPng	RIP Next Generation	2080
OSPFv3	OSPF Version 3	5340
MP-BGP4	Multiprotocol BGP-4	4760
EIGRP for IPv6	EIGRP for IPv6	Proprietary

7.9.1 RIPng

Comparing RIP-2 to RIPng

Feature	RIP-2	RIPng
Advertises routes for...	IPv4	IPv6
RIP messages use these Layer 3/4 protocols	IPv4, UDP	IPv6, UDP
UDP Port	520	521
Use Distance Vector	Yes	Yes
Default Administrative distance	120	120
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses Split Horizon	Yes	Yes
Uses Poison Reverse	Yes <i>(Cisco)</i>	Yes
30 second periodic full updates	Yes	Yes
Uses triggered updates	Yes	Yes
Uses Hop Count metric	Yes	Yes
Metric meaning infinity	16	16
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.9	FF02::9
Authentication	RIP-specific	uses IPv6 AH/ESP

The overall operation of RIPng closely matches RIP-2:

- Routers send **periodic full updates with all best routes**, except for routes omitted due to Split Horizon rules.
- **No neighbor relationships occur**; the continuing periodic Updates, on a slightly-variable 30 second period, also serve the purpose of confirming that the neighboring router still works.
- The metrics work exactly the same.
- When a router ceases to see a route in received updates, ceases to receive updates, or receives a poisoned (metric 16) route, it reacts to converge, but relatively slowly compared to EIGRP and OSPF.

Some differences relate specifically to IPv6:

- First, the messages themselves list IPv6 prefixes/lengths, rather than IPv4 subnet/mask.
- In RIP-1 and RIP-2, RIP encapsulated RIP Update messages inside an UDP and IPv4 header; with IPv6, the encapsulation uses UDP inside IPv6 packets.
- Some small differences in the Update message format exist as well, with the most obvious difference being that the updates list IPv6 prefixes and prefix lengths.
- Because IPv6 supports authentication using the IPsec Authentication Header (AH) and/or Encapsulating Security Payload (ESP), **RIPng** does not natively support authentication, instead relying on **IPsec**. **RIPv2** supports **MD5**.
- **The last difference of note is that some routers sitting on different IPv6 subnets in a common link can still exchange their RIPng routes.**

7.9.2 EIGRPv6

For the most part, EIGRP for IPv4 and for IPv6 (*EIGRPv6 is a personal shortcut*) have many similarities, including neighborship, timers, packet types, DUAL, metrics, load-balancing, administrative distance and diameter.

The following list outlines some of the key *differences*:

- EIGRPv6 advertises **IPv6 prefixes/lengths**, rather than IPv4 subnet/mask information.
- EIGRPv6 uses **the neighbor's link local address as the next-hop IP address**; EIGRP for IPv4 has no equivalent concept.
- EIGRPv6 encapsulates its messages in **RTP/IPv6** packets, rather than RTP/IPv4 packets.
- Like RIPng and OSPFv3, EIGRPv6 **authentication and privacy features rely on IPsec**.
- EIGRP for IPv4 defaults to use automatic route summarization at the boundaries of classful IPv4 networks; IPv6 has **no concept of classful networks**, so EIGRPv6 *cannot perform any automatic summarization*.
- EIGRPv6 **does not require neighbors to be in the same IPv6 subnet**.

Some routers sitting on different IPv6 subnets in a common link can still be neighbors and exchange their EIGRPv6 routes.

- EIGRPv6 requires a **router ID** before it can start running.
- EIGRPv6 is **configured on interfaces**. No network statements are used.
- EIGRPv6 **defaults to a shutdown state** and must be manually or explicitly enabled on an interface to become operational.

Other than these differences, most of the details of EIGRPv6 work like EIGRP for IPv4.

Comparing EIGRP for IPv4 and IPv6

Feature	EIGRP for IPv4	EIGRP for IPv6
Advertises routes for...	IPv4	IPv6
Layer 3 protocol for EIGRP messages	IPv4	IPv6
Layer 3 header protocol type (<i>RTP</i>)	88	88
UDP Port	N/A	N/A
Uses Successor, Feasible Successor logic	Yes	Yes
Uses Dual	Yes	Yes
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses triggered updates	Yes	Yes
Uses composite metric, default using bandwidth and delay	Yes	Yes
Metric meaning infinity	$2^{32} - 1$	$2^{32} - 1$
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.10	FF02::A
Authentication	EIGRP-specific	Uses IPv6 AH/ESP

7.9.3 OSPFv3

7.9.3.1 Comparing OSPFv3 with OSPFv2

- The multicast addresses used by OSPFv3 of course differ, but they keep similar numbers compared to OSPFv2.
- OSPF uses IPv6 inherent **IPsec** capabilities, rather than defining a separate authentication process.
- **Per-link processing** (instead of per-subnet): As with RIPng and EIGRP for IPv6, OSPFv3 **does not require neighboring routers to be in the same subnet** as a requirement for becoming neighbors. However, OSPFv3 does follow all other neighbor verification checks as compared with OSPFv2,
Some routers sitting on different IPv6 subnets in a common link can still be neighbors and exchange their LSAs.
- **Address semantics removed**: Addresses are removed from the router and network LSAs. These LSAs now provide topological information only.
- **Multiple instances of OSPF on a single link** are supported, whereas OSPFv2 supports only a single instance per link.
- OSPF uses the **neighbor's link local IPv6 address as the next-hop IP address**.
- **New link LSA**: For local-link flooding scope.
- **New intra-area-prefix LSA**: Carries all the IPv6 prefix information. Similar to OSPFv2 router and network LSAs.
- **Identifying neighbors by router ID**: Neighbors are always identified by the router ID. This does not occur in OSPFv2 point-to-point and broadcast networks. The **RID is still based on IPv4 addresses**, not IPv6 addresses. Like EIGRP for IPv6, a router must have a RID before OSPFv3 will work.

Note: In OSPFv3, the router IDs, area IDs, and LSA link-state IDs remain at the size of 32 bits. Larger IPv6 addresses cannot be used.

Comparing OSPF for IPv4 and IPv6

Feature	OSPFv2	OSPFv3
Advertises routes for...	IPv4	IPv6
OSPF messages use this layer 3 protocol	IPv4	IPv6
IP Protocol Type	89	89
Uses Link State logic	Yes	Yes
Supports VLSM	Yes	Yes
Process to choose RID, compared to OSPFv2	Same	Same
LSA flooding and aging compared to OSPFv2	Same	Same
Area structure compared to OSPFv2	Same	Same
Packet types and uses compared to OSPFv3	Same	Same
RID based on highest up/up loopback IPv4 address, or highest other IPv4 interface address?	Yes	Yes
32-bit LSID	Yes	Yes
Uses interface cost metric, derived from interface bandwidth	Yes	Yes
Metric meaning infinity	$2^{16} - 1$	$2^{16} - 1$
Supports route tags	Yes	Yes
Elects DR based on highest priority, then highest RID	Yes	Yes
Periodic reflooding every...	30 minutes	30 minutes
Multicast—all SPF routers	224.0.0.5	FF02::5
Multicast—All Designated routers	224.0.0.6	FF02::6
Authentication	OSPF-specific	Uses IPv6 AH/ESP
Neighbor checks compared to OSPFv2	Same	Same, except no "same subnet" check
Multiple instances per interface	No	Yes

/ interface

7.9.3.2 LSA Types

LSA Name	LS Type	Description
Router LSA	0x2001	Cost and state of all the router's interfaces. <u>No IPv6 prefixes</u>
Network LSA	0x2002	Generated by DR routers. <u>No IPv6 prefixes</u>
Interarea-prefix LSA	0x2003	Routes to prefixes in other areas. Same as OSPFv2 Type 3
Interarea-router LSA	0x2004	Routes to routers in other areas. Same as OSPFv2 Type 4 <u>No IPv6 prefixes</u> - only ASBR and ABR RIDs
Autonomous system external LSA	0x4005	Routes to networks external to the autonomous system Same as OSPFv2 Type 5
Group-membership LSA	0x2006	Networks that contain multicast groups Same as OSPFv2 Type 6
NSSA LSA	0x2007	Routes to networks external to the autonomous system, injected into the NSSA. Same as OSPFv2 Type 7
Link LSA	0x0008	Link-local addresses and list <u>IPv6 prefixes</u> associated with the link
Intra-area-prefix LSA	0x2009	<u>IPv6 prefixes</u> associated with a router, a stub network, or an associated transit network segment

- **Link LSA** is a new LSA that describes the router link-local address and a list of IPv6 prefixes associated with the link. This LSA is flooded to the local link only.
- The **intra-area-prefix LSA** is a new LSA type that is used to advertise IPv6 prefixes associated with a router, a stub network, or an associated transit network segment. This LSA contains information that used to be part of the router LSAs and network LSAs.

7.9.4 IS-IS for IPv6

Specifications for routing IPv6 with integrated IS-IS are currently a standard [RFC 5308](#) of the IETF. The RFC specifies new type, length, and value (TLV) objects, reachability TLVs, and an interface address TLV to forward IPv6 information in the network.

7.9.5 MP-BGP, a.k.a BGP4+

- **BGP4 Multiprotocol Extensions (MP-BGP) for IPv6**

[RFC 2545](#) specifies the use of BGP attributes for passing on IPv6 route information. MP-BGP is also referred as **BGP4+**.

IOS currently supports these BGP4 **multiprotocol attributes** to communicate reachability information for IPv6 networks:

- The **MP_REACH_NLRI** (multiprotocol-reachable) attribute describes reachable destinations. It includes the next-hop address and a list of prefixes (NLRI) of reachable networks.
- The **MP_UNREACH_NLRI** (multiprotocol-unreachable) attribute conveys unreachable networks.

7.9.6 Summary of all IPv4 and IPv6 IGPs Summary Route Metric

IGPs	Metric of Summary Route [^]
<i>RIP</i>	<i>Lowest</i>
<i>RIPng</i>	<i>Lowest</i>
<i>EIGRP</i>	<i>Lowest</i>
<i>EIGRPv6</i>	<i>Lowest</i>
<i>OSPFv2</i>	<i>Lowest*</i>
<i>OSPFv3</i>	<i>Highest</i>

[^]: As compared with the metric of all subordinate subnets.

^{*}: Except if the "(config-router)#no compatible rfc1583" command is issued on the summarizing router, in which case the highest metric is chosen among all subordinate subnets.

<https://t.me/learningnets>

7.10 IGP Redistribution

The following list summarizes some of those key *similarities* between both IPv4 and IPv6 route redistribution:

- Redistribution takes routes from the IP routing table, *not* from the topology tables and databases controlled by the source routing protocol.
- Route maps can be applied when redistributing for the purpose of :
 - filtering routes,
 - setting metrics and metric types,
 - setting route tags, ...
- IPv6 routing protocols use the same default administrative distances (AD) settings for internal and external routes, with the same basic mechanisms to override those defaults.
- The same basic mechanisms exist in IPv6 to defeat **IIDR** problems:
 - administrative distance,
 - filtering,
 - route tags.
- The redistribution configuration uses practically the same syntax with the same commands.

7.11 IPv6 Comparison with IPv4

Characteristic	IPv6	IPv4
Address length	128 bits	32 bits
Address representation	Hexadecimal	Dotted-decimal
Header length	Fixed (40 bytes)	Variable
Upper-layer protocols	Next header field	Protocol Type field
Link address resolution	ND	ARP
Address configuration	Stateless autoconfiguration or stateful DHCP	Stateful DHCP
DNS (name-to-address resolution)	AAAA records	A records
Interior routing protocols	EIGRPv6, OSPFv3, RIPng, IS-IS for IPv6	EIGRP, OSPFv2, RIPv2, IS-IS
Classification and marking	Traffic Class and Flow Label fields, differentiated services code point (DSCP)	IP Precedence bits, Type of Service field, DSCP
Private addresses	Unique-local addresses	RFC 1918 private address space
Fragmentation	Sending host only	Sending host and intermediate routers
Loopback address	0:0:0:0:0:0:1	127.0.0.1
Address scope types	Unicast, anycast, multicast	Unicast, multicast, broadcast

7.12 Guidelines

7.12.1 Enterprise Deployment

Very interesting [RFC 7381](#) written mainly by some people at Cisco, Time Warner Cable and HP.

7.12.2 Data Center Operation

IPv6 Operational Guidelines for data centers: [draft-ietf-v6ops-dc-ipv6-01](#)

7.13 Summary of Features – RIPE-544

IPv6 includes some attractive features and migration tools:

- **Larger address space:** IPv6 uses 128-bit addresses rather than the 32-bit addresses in IPv4.
- **Address assignment features:** IPv6 hosts can automatically configure themselves, *with or without a DHCP server*. Stateful and stateless auto-configuration is supported.
- **Support for address renumbering:** IPv6 supports the ability to change the public IPv6 prefix used for all addresses in an Enterprise, using the capability to advertise the current prefix with a short timeout and the new prefix with a longer lease life.
- **Support for mobility and multicast:** IPv6 supports mobility such that IPv6 hosts can move around the Internet and retain their IPv6 address without losing current application sessions.
- **Provides independent and dependent public address space:** ISPs can assign public IPv6 address ranges (dependent), or companies can register their own public address space (independent), called PI (Provider-Independent) address space.
- **Site multihoming:** IPv6 allows multihoming by allowing hosts to have multiple IPv6 addresses and networks to have multiple IPv6 prefixes, which facilitates *connection to multiple ISPs*.
- **Aggregation:** IPv6 huge address space makes for much easier aggregation of blocks of addresses in the Internet, making routing in the Internet more efficient.
- **No need for NAT/PAT:** The huge public IPv6 address space removes the need for NAT/PAT, which avoids some NAT-induced application problems and makes for more efficient routing.
- **IPsec:** Unlike IPv4, IPv6 requires that every IPv6 implementation support IPsec. IPv6 does not require that each device use IPsec, but any device that implements IPv6 must also have the ability to implement IPsec.
- **Header improvements:** The IPv6 *header length is fixed* and routers *do not need to recalculate a header checksum* for every packet, lowering header processing time. Additionally, the header includes a *flow label* that allows for easy identification of packets sent over the same single TCP or UDP connection. Traffic Class field replaces the ToS/DS-IPv4 field.
- **MTU path discovery:** IPv6 eliminates the need to fragment packets by implementing MTU path discovery before sending packets to a destination.
- **Transition tools:** IPv6 has many rich tools to help with the transition from IPv4 to IPv6.
- **Formally no broadcasts:** IPv6 does not use Layer 3 broadcast addresses, instead relying on multicasts to reach multiple hosts with a single packet. *In fact, the 'All-Nodes' (FF02::1) address is treated the same way as an IPv4 broadcast address.*

RIPE-NCC has published a well-known document, [RIPE-544](#), which lists the requirements for IPv6 in ICT Equipment and can be used to choose the correct IPv6 devices.

7.14 Configuration Commands

7.14.1 Addresses

- with a dynamic global unicast IPv6 address, either:
 - with stateful DHCP,
 - with stateless autoconfig (EUI-64),
- with a static global unicast IPv6 address, either:
 - without using the EUI-64 option,
 - with the EUI-64 option,
- *without* a global unicast address, instead relying solely on its link local address,
- with a different link local address than the one based on EUI-64 (but beginning with FE80::/10),
- with an *anycast* address,
- with the *unnumbered feature* (on serial interfaces), borrowing another interface's address.

IOS IPv6 Address Configuration Command Reference

(config-if)#...

Command	Description
ipv6 address <i>address/length</i>	Static configuration of the entire IPv6 unicast address.
ipv6 address <i>prefix/length eui-64</i>	Static configuration of the first 64 address bits; the router derives the last 64 bits with EUI-64.
ipv6 address autoconfig	Router uses stateless autoconfig to find address.
ipv6 address dhcp	Router uses stateful DHCP to find address.
ipv6 unnumbered <i>interface-type number</i>	Uses the same IPv6 unicast address as the referenced interface.
ipv6 enable	Enables IPv6 on the interface, but results in only a link local address.
ipv6 address <i>address link-local</i>	Overrides the automatically created link local address. The configured value must conform to the FE80::/10 prefix.
ipv6 address <i>address/length anycast</i>	Designates that the unicast address is an anycast.

ipv6 general-prefix *prefix-name prefix/length* with ...

ipv6 address *prefix-name sub-bits/length*

no ipv6 address

Removes all configured IPv6 addresses from the interface and also disables IPv6 on that interface.

Note : All these interface subcommands enable IPv6 on the interface (except the last one), which means the router derives an IPv6 link local address for the interface

7.14.2 Static Routing

(config)#**ipv6 route *prefix/length*** {*outgoing-interface [next-hop-address]* | *next-hop-address*} [*admin-distance*]

[*tag tag-value*] **You must specify an outgoing-interface when using a link-local address as the next-hop-address**

7.14.3 Routing Protocols

7.14.3.1 RIPng

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing
- **Step 1:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 2:** (config)#**ipv6 router rip name** Enables RIPng (*not necessary to start the process, only for configuration*). The name must be unique on a router but does not need to match neighboring routers.
- **Step 3:** Enable IPv6 on the interface, typically with one of these two methods:
 - (config-if)#**ipv6 address prefix/length [eui-64]** Configures an IPv6 unicast address on each int.,
 - (config-if)#**ipv6 enable** Enables IPv6 and causes the router to derive its link local address.
- **Step 4:** (config-if)#**ipv6 rip name enable** Enables RIP on this interface (where the *name* matches the **ipv6 router rip name** global configuration command).
 - First, it starts sending RIP updates on that interface.
 - It also starts processing any RIP updates received on that interface.
 - Finally, it advertises about the connected routes on that interface. In particular, because IPv6 allows the configuration of multiple IPv6 unicast addresses on an interface, RIP advertises about most IPv6 unicast prefixes associated with the interface. The notable exceptions are that RIP *does not advertise about any link local addresses*, nor does RIP advertise about *the local host routes*—routes with a /128 prefix length—created for each interface IPv6 address. In short, RIP advertises *all routable subnets associated with the interface*.

(config-rtr)#**passive-interface interface** *Disables the sending of routing updates on that interface, this prefix continues to be advertised on other interfaces, and updates from other routers on that interface continue to be received and processed.*

(config-if)#**ipv6 rip name summary-address prefix/len** *Advertises a summary address on this interface*

7.14.3.2 EIGRPv6

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing
- **Step 1:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 2:** (config)#**ipv6 router eigrp asn** Enables EIGRP ($asn \in \{1 - 65535\}$)
- **Step 3:** (config-rtr)#**eigrp router-id rid** If no EIGRP router ID has been automatically chosen, due to not having at least one working interface with an IPv4 address, configure an EIGRP router ID in IPv4 format.
- **Step 4:** (config-rtr)#**no shutdown** Enables EIGRPv6 (*necessary*).
- **Step 5:** Enable IPv6 on the interface, typically with one of these two methods:
 - (config-if)#**ipv6 address prefix/length [eui-64]** Configures an IPv6 unicast address on each interface,
 - (config-if)#**ipv6 enable** Enables IPv6 and causes the router to derive its link local address.
- **Step 6:** (config-if)#**ipv6 eigrp asn** Enables EIGRP on this interface (where the *asn* matches the **ipv6 router eigrp asn** global configuration command).

(config-if)#**ipv6 summary-address eigrp asn prefix/len [admin-distance]** *Advertises a summary address on this int.*

7.14.3.3 OSPFv3

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing
- **Step 1:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 2:** (config)#**ipv6 router ospf process-id** Creates an OSPFv3 routing process (*not necessary to start the process if it already has an RID, only for configuration*).
- **Step 3:** (config-rtr)#**router-id rid** If no OSPF router ID has been automatically chosen, due to not having at least one working interface with an IPv4 address, configure an OSPF router ID in IPv4 format.
- **Step 4:** Enable IPv6 on the interface, typically with one of these two methods:
 - (config-if)#**ipv6 address prefix/length [eui-64]** Configures an IPv6 unicast address on each interface,
 - (config-if)#**ipv6 enable** Enables IPv6 and causes the router to derive its link local address.
- **Step 5:** (config-if)#**ipv6 ospf process-id area area-number** Enables OSPF on this interface (where the *process-id* matches the **ipv6 router ospf process-id** global configuration command).

#clear ipv6 ospf [process-id] process

The OSPF database is cleared and repopulated, and then the shortest path first (SPF) algorithm is performed.

- *The command for entire Area Type-3 LSA filtering has been removed from OSPFv3: (config-rtr)#area filter-list. The filtering could be made on all intended routers with a “distribute-list prefix-list in” command (not very scalable). Or, as suggested by [Orsa](#), another workaround would be to use “area range not-advertise” command on the ABR.*

7.14.4 Migrating from IPv4 → IPv6

7.14.4.1 Static Point-to-Point Tunnels

7.14.4.1.1 6in4 : Manually Configured Tunnels

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing.
- **Step 0:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 1:** Find the IPv4 *encapsulating* addresses planned for the tunnel, and ensure that each router can forward IPv4 packets between the addresses. If using a new loopback interface:
 - (config)#**interface loopback number** Creates the loopback,
 - (config-if)#**ip address ip-address mask** Assigns an IPv4 address to the loopback interface,
 - *Confirm that routes for this interface will be advertised by the IPv4 IGP.*
- **Step 2:** (config)#**interface tunnel number** Creates a tunnel interface, selecting a *locally significant* integer as the tunnel interface number.
- **Step 3:** (config-if)#**tunnel source {interface | ipv4-address}** Defines the *source IPv4 address for encapsulation* in the tunnel interface (This address must be an IPv4 address configured on the local router).
- **Step 4:** (config-if)#**tunnel destination ipv4-address** Defines the *destination IPv4 address for encapsulation* ; this address must match the **tunnel source** command on the other router.
- **Step 5:** (config-if)#**tunnel mode ipv6ip** Defines the tunnel as a *6in4*.
- **Step 6:** (config-if)#**ipv6 address address/len [eui-64]** Enables IPv6 on the tunnel interface with the address. *Not absolutely necessary: ipv6 enable can be enough!*
- **Step 7:** Configuring an *IPv6 IGP* on this tunnel interface is possible
- **Step 8:** Complete the normal IPv6 configuration, including defining the LAN interface IPv6 addresses.

Notes :

- First, note that the tunnel interface *may* have an IPv6 address configured, and an IGP enabled, just as if it were a physical interface.
- However, the tunnel interface *does not have, nor does it need, an IPv4 address*.

7.14.4.1.2 GRE Tunnels

(config-if)#**tunnel mode gre ip**

GRE tunnel encapsulation as defined in RFC 2784. Also, because IOS defaults to use GRE over IP, you can *alternatively* just omit the **tunnel mode** command.

For example, to migrate from 6in4 to GRE configuration, simply use *one* of the following two options:

- (config-if)#**tunnel mode gre ip** On both routers' tunnel interfaces.
- Or (config-if)#**no tunnel mode ipv6ip** On both routers' tunnel interfaces, which reverts to the default of GRE / IP.

7.14.4.1.3 Comparing Point-to-Point Tunnels

	Manual Tunnels	GRE
RFC	4213	2784
Tunnel mode command	tunnel mode ipv6ip	tunnel mode gre ip
Passenger MTU default	1480	1476
Supports IPv6 IGP?	Yes	Yes
Forwards IPv6 multicasts?	Yes	Yes
Uses static configuration of tunnel destination?	Yes	Yes
Supports multiple passenger protocols?	No	Yes
Link local based on...	FE80::/96, plus 32 bits from tunnel source IPv4 address	IPv6 EUI-64, using lowest numbered interface's MAC address
<i>Mandatory definition of an IPv6 address for the tunnel interface</i>	No	No
<i>Protocol in IPv4 header</i>	41 (IPv6)	47 (GRE)

7.14.4.2 Dynamic Point-to-Multipoint Tunnels

7.14.4.2.1 6to4 Tunnels

7.14.4.2.1.1 'Reserved Mode'

- For the tunnel itself, unlike point-to-point tunnels:
 - The multipoint tunnel configuration *does not need* a **tunnel destination** command, because the destination IPv6 address is instead embedded in the destination IPv6 address.
 - Also, the setting of the **tunnel mode** command differs.
- For the IPv6-specific configuration related to the tunnel:
 - Even though the tunnel interface must be enabled for IPv6 as with point-to-point tunnels, the IPv6 addresses are *not in the same subnet* as the other routers' tunnel interfaces.
 - **Defining a specific unicast reserved IPv6 address on the Tunnel interface is not absolutely necessary: a link-local IPv6 address can be enough!**
 - Although IPv6 IGPs cannot be configured on the automatic 6to4 tunnel, static IPv6 routes and multiprotocol BGP (for IPv6) can be configured.

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing.
- **Step 0:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 1:** Find the IPv4 *encapsulating* addresses planned for the tunnel, and ensure that each router can forward IPv4 packets between the addresses. If using a new loopback interface:
 - (config)#**interface loopback number** Creates the loopback,
 - (config-if)#**ip address subnet mask** Assigns an IPv4 address to the loopback interface,
 - Confirm that routes for this interface will be advertised by the IPv4 IGP.
- **Step 2:** (config)#**interface tunnel number** Creates a tunnel interface, selecting a *locally significant* integer as the tunnel interface number.
- **Step 3:** (config-if)#**tunnel source {interface | ipv4-address}** Defines the *source IPv4 address for encapsulation* in the tunnel interface (This address must be an IPv4 address configured on the local router).
- **Step 4:** (config-if)#**tunnel mode ipv6ip 6to4** Defines the tunnel as a 6to4.
- **Step 5:** (config-if)#**ipv6 address address/len [eui-64]** Enables IPv6 on the tunnel interface with the address. Embeds the chosen IPv4 address of the router. *Not absolutely necessary: ipv6 enable can be enough!*
- **Step 6:** Complete the normal IPv6 configuration, include defining the LAN interface IPv6 addresses.
- **Step 7:** (config)#**ipv6 route 2002::/16 tunneln** Defines a static route for 2002::/16, on outgoing tunnel interface.

The following logic occurs in the background:

- **Step 1:** The router at one end of the tunnel will have two connected IPv6 routes (LAN's and tunneln's) in the 2002::/16 range, plus a static route for the entire 2002::/16 range.
- **Step 2:** When this router receives an IPv6 packet from the LAN, destination in the 2002::/16 range, and the destination is not in one of the connected subnets, it will try to forward the packet out tunneln.
- **Step 3:** The **tunnel mode ipv6ip 6to4** command tells the router to look to the 2nd/3rd quartets to find the destination IPv4 address, and perform the tunneling.

7.14.4.2.1.2 'Public Mode'

Configure an *additional static route for each remote subnet*:

- **Step 8:** (config)#**ipv6 route prefix/len tunneln next-hop** Defines a static IPv6 route for each destination IPv6 prefix, on outgoing tunnel interface **and** with the next-hop address being the destination router's IPv6 Tunnel address that embeds the IPv4 address as the second and third octets.

- *The tunnel interface must have an IPv6 address which embeds the chosen IPv4 address of the router.*
- The use of global unicast addresses requires more configuration changes: when a new router is added to the multipoint tunnel, *each router already on the tunnel needs to add additional static routes or the alternative additional BGP configuration.*

7.14.4.2.2 ISATAP Tunnels

- **Step 0:** (config)#**ipv6 unicast-routing** Enables IPv6 routing.
- **Step 0:** (config)#**ipv6 cef** *Enables IPv6 Cisco Express Forwarding*
- **Step 1:** Find the tunnel IPv4 *encapsulating* addresses planned for the tunnel, and ensure that each router can forward IPv4 packets between the addresses. If using a new loopback interface:
 - (config)#**interface loopback number** Creates the loopback,
 - (config-if)#**ip address subnet mask** Assigns an IPv4 address to the loopback interface,
 - Confirm that routes for this interface will be advertised by the IPv4 IGP.
- **Step 2:** (config)#**interface tunnel number** Creates a tunnel interface, selecting a *locally significant* integer as the tunnel interface number.
- **Step 3:** (config-if)#**tunnel source {interface | ipv4-address}** Defines the *source IPv4 address for the encapsulation* in the tunnel interface (This address must be an IPv4 address configured on the local router).
- **Step 4:** (config-if)#**tunnel mode ipv6ip isatap** Defines the tunnel as *ISATAP*.
- **Step 5:** (config-if)#**ipv6 address prefix/len [eui-64]** Configure a *necessary* IPv6 prefix *with modified EUI-64 option* on the tunnel interface.
- **Step 6:** Complete the normal IPv6 configuration, include defining the LAN interface IPv6 addresses.
- **Step 7:** (config)#**ipv6 route prefix/len tunneln next-hop** Defines *a static IPv6 route for each destination IPv6 prefix* on outgoing tunnel interface **and** with the next-hop address being the destination router's IPv6 Tunnel address that embeds the IPv4 address as the last two octets.

7.14.4.2.3 6RD Tunnels

Out of CCNP ROUTE scope (at the time of last edition), but this migration tool is more and more used nowadays.

- Cf. [6RD Configuration Example](#).
- Configuration Mode: `ipv6ip 6rd`

7.14.4.2.4 Comparing Multipoint Tunnels

	Automatic 6to4	ISATAP
RFC	3056	4214
Uses a reserved IPv6 address prefix.	Yes (2002::/16)	No
Supports the use of global unicast addresses?	<i>In 'Public' mode</i>	Yes
Quartets holding the IPv4 destination address.	2/3	7/8
End-user host addresses embed the IPv4 destination?	<i>In 'Reserved' mode</i>	No
Tunnel endpoints IPv6 addresses embed IPv4 destination.	Yes	Yes
Uses modified EUI-64 to form tunnel IPv6 addresses?	No	Yes
<i>Supports IPv6 IGPs</i>	<i>No</i>	<i>No</i>
<i>Supports MP-BGP4</i>	<i>Yes</i>	<i>Yes</i>
<i>Needs an IPv4 'tunnel destination' address</i>	<i>No</i>	<i>No</i>
<i>Mandatory conf of IPv6 address for the tunnel interface</i>	<i>In 'Public' mode</i>	<i>Yes</i>
<i>Needs a static route for '2002::/16' prefix</i>	<i>Yes</i>	<i>-----</i>
<i>Needs one static route per <u>Public</u> destination prefix</i>	<i>In 'Public' mode</i>	<i>Yes</i>
<i>Protocol in IPv4 header</i>	<i>41 (IPv6)</i>	<i>41 (IPv6)</i>

7.14.5 IGP Redistribution

Some *differences* do exist between both IPv4 and IPv6 route redistribution:

- Any matching done with distribution lists or route maps would use **IPv6 prefix lists and IPv6 ACLs**, which match based on IPv6 prefix and length.
- The IPv6 version of the **redistribute** command takes *only routes learned from an IGP or BGP (no ‘not-IGP-enabled’ connected interfaces nor static routes)* but *by default does not take connected routes on interfaces enabled for that IGP*. To also redistribute those connected routes, the **redistribute** command must include the **include-connected** parameter. When an IPv4 routing protocol redistributes from an IGP, it always attempts to take both the IGP-learned routes and the connected routes for interfaces enabled for that IGP.
- *Redistribution into RIPv6 uses a metric as taken from the IPv6 routing table by default*. OSPF/EIGRP metrics will often be more than the maximum usable RIP metric of 15, making these routes instantly unusable in RIP. So when redistributing into RIP, change the configuration to set the metric.
- Unlike OSPFv2, OSPFv3 *does not require* a **subnets** parameter on the **redistribute** command, because IPv6 does not maintain the IPv4 concept of classful networks and the subnets inside those classful networks.
- IPv6 redistribution ignores the “local” routes in the IPv6 routing table (the /128 host routes for a router’s own interface IPv6 addresses). IPv4 has no equivalent concept.

7.15 Verification Commands

#show ipv6 neighbors Lists the *LAN* neighbor table entries (Like IPv4 Mac Table). *Dynamic entries in one of those states (only the first two for static entries):*

- **INCOMP (INCOMPLETE):** Address resolution is being performed on the entry,
- **REACH(ABLE):** Positive confirmation has been received,
- **STALE:** More than *ReachableTime* ms have elapsed since the last positive confirmation was received that the forward path was functioning properly, (this happens for example when there’s no ipv6 traffic with this neighbor),
- **DELAY:** More than *ReachableTime* ms have elapsed since the last positive confirmation was received that the forward path was functioning properly, and no reachability confirmation has been received,
- **PROBE:** A reachability confirmation is actively sought by retransmitting Neighbor Solicitations.
- **????:** unknown state.

#show ipv6 routers Lists the cached contents of any received RA messages, including neighboring routers’ link local address (used to identify the routers) and their *shared* prefix.

Comparing RIP-2 and RIPv6 Verification Commands: [show ip](#) and [show ipv6](#)

Function	IPv4	IPv6
All routes	... route	... route
All RIP learned routes	... route rip	... route rip
Details on the routes for a specific prefix	... route subnet mask	... route prefix/length
Interfaces on which RIP is enabled	... protocols	... protocols /... rip
List of routing information sources	... protocols	... rip next-hops
Debug that displays sent and received Updates	debug ip rip	debug ipv6 rip

“New” IPv6 **show ipv6** commands :

#show ipv6 rip Displays:

- Timers
- Interfaces on which RIPv6 is enabled
- Redistribution

#show ipv6 rip next-hops Displays “Routing Information Sources”.

#show ipv6 rip database Displays the entries in the RIP table.

<https://t.me/learningnets>

Comparing EIGRPv4 and EIGRPv6 Verification Commands: show ip and show ipv6...

Function	show ip...	show ipv6...
All routes	... route	... route
All EIGRP learned routes	... route eigrp	... route eigrp
Details on the routes for a specific prefix	... route <i>subnet mask</i>	... route <i>prefix/length</i>
Interfaces on which EIGRP is enabled, plus metric weights, variance, redistribution, max-paths, admin distance	... protocols	... protocols
List of routing information sources	... protocols ... eigrp neighbors	... eigrp neighbors
Hello interval	... eigrp interfaces detail	... eigrp interfaces detail
EIGRP database	... eigrp topology [all-links]	... eigrp topology [all-links]
Debug that displays sent and received Updates	debug ip eigrp notifications	debug ipv6 eigrp notifications
<i>Current Hold Timer</i>	... eigrp neighbors	... eigrp neighbors

Comparing OSPFv2 and OSPFv3 Verification Commands: show ip and show ipv6...

	Function	show ipv4...	show ipv6...
Including Redistribution and Areas information	All OSPF-learned routes	... route ospf	... route ospf
	Router ID, Timers, ABR, SPF statistics	... ospf	... ospf
	List of routing information sources	... protocols ... ospf neighbor	... ospf neighbor
	Interfaces assigned to each area	... protocols ... ospf interface brief ... <i>ospf</i>	... protocols ... ospf interface brief ... <i>ospf</i>
Including Hello and Dead Timers	OSPF interfaces—costs, state, area, number of neighbors	... interface brief	... interface brief
	Detailed information about OSPF interfaces	... ospf interface	... ospf interface
	Displays summary of OSPF database	... ospf database	... ospf database

The only noticeable difference between the output of the **show ip ospf neighbor** and **show ipv6 ospf neighbor** commands is:

OSPFv2 displays the interface IP address of neighbors, but OSPFv3 displays the neighbor interface ID. OSPFv3 routers create a *locally significant* interface ID, using that in LSAs that describe the intra-area topology, rather than using neighbor IPv4 addresses to describe the topology. OSPFv3 separates the topology information from the Layer 3 addressing information, so OSPFv3 uses the interface ID concept to identify how a router reaches a particular neighbor.

#show ipv6 ospf database router adv-router this-router-id Shows these local and neighbor interface IDs in Router LSAs

#show ipv6 ospf database router self-originate Same with the local router.

The **show ipv6 ospf interface brief** command lists the same kind of information shown in the similar **show ip ospf interface brief** command, but again listing the *local interface ID* as new information.

The **show ipv6 protocols** command lists much sparser information as compared to the similar **show ip protocols** command. Note that both commands identify which OSPFv3 interfaces have been assigned to each area.

➤ *The show ipv6 ospf border-routers command has the same behavior related to:*

- *Exterior → NSSA views,*
- *Stubby → Exterior views.*

#show ipv6 route prefix/len Lists some information not seen in the equivalent IPv4 command. The output reveals some insight into the operation of the routing table manager (RTM), which chooses the best route for a prefix among many sources. The output of this command lists not only the static route, but also the backup route, with a worse administrative distance. When configuring a *floating static route*—a static route with a high administrative distance so that the route will not be used until the IGP routes have all failed—this command can be useful to confirm that this backup route is known to IOS.

6in4 and GRE Tunnels

#show interfaces tunneln Shows IPv4 information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- IPv4 *Source and Destination* addresses
- Tunnel *Passenger* (IPv6 or GRE) and *Transport* protocols (IP)

#show ipv6 interface tunneln Shows IPv6 information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- (*Automatically formed*) Link local and (static) Global Unicast IPv6 addresses
- Joined IPv6 Multicast Group addresses, including:
 - All link local nodes multicast
 - All link local routers multicast
 - IGP multicast
 - the solicited node multicast address associated with *both* the link local and the global unicast address on the interface.
- Tunnel Transport MTU (*1480 for 6in4, 1476 for GRE*)

#traceroute ipv6-address Confirms that the static tunnel is functional.

6to4 Tunnels

#show interfaces tunneln Shows information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- IPv4 *Source* address
- **NO IPv4 Destination address**
- Tunnel *Passenger* (IPv6) and *Transport* protocols (6to4)

#show ipv6 interface tunneln Shows IPv6 information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- (*Automatically formed*) Link local and (static) Reserved Unicast IPv6 addresses
- Joined IPv6 Multicast Group addresses, including:
 - All link local nodes multicast
 - All link local routers multicast
 - the solicited node multicast addresses associated with *both* the link local and the global unicast address on the interface.
- Tunnel Transport MTU (*1480*)

#show ipv6 interface brief Confirms the configured IPv6 address from the 2002::/16 range, plus the link local address derived for the Tunnel interface. **Note** that link local address uses an unusual convention—

instead of using the usual EUI-64 logic, the link local address starts with FE80::/96, with the last two quartets the same as the tunnel's IPv4 source address.

#show ipv6 route

Lists the static and connected routes. **Note** that if the configuration used a /128 prefix length for Tunnel's IPv6 address, the connected route would show up on the same line as the local route, both of which having a /128 prefix length.

#traceroute ipv6-address

Confirms that the 6to4 tunnel is functional.

ISATAP Tunnels

#show interfaces tunneln

Shows information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- Tunnel source IPv4 address
- ***NO Tunnel Destination IPv4 address***
- Tunnel *Passenger* (IPv6) and *Transport* protocols (ISATAP)

#show ipv6 interface tunneln

Shows IPv6 information about this Tunnel interface :

- Line and protocol status of Tunnel interface
- (*Automatically formed*) Link local and (static) Global Unicast IPv6 addresses
- Joined IPv6 Multicast Group addresses, including:
 - All link local nodes multicast
 - All link local routers multicast
 - the ***one*** solicited node multicast address associated with *both* the link local and the global unicast address on the interface (***same last 24 bits***).
- Tunnel Transport MTU (1480)

#show ipv6 interface brief

Lists the Tunnel interface IPv6 address, derived using the modified EUI-64 rules. It also lists a link local address on the interface, formed not with traditional EUI-64 rules but instead with the same modified EUI-64 rules used for the global unicast address.

#show ipv6 route

Lists the static routes for the remote LAN subnets, with remote routers' modified EUI-64 IPv6 address as the next hop. The output also shows the local router's connected route for the tunnel subnet.

#traceroute

Confirms that the local router can send packets to remote routers' LAN IPv6 addresses from the local router's LAN IPv6 address.

#debug ipv6 nd

Tells the router to issue messages related to NDP messages.

8 Branch Internet Connections

8.1 Branch Office Broadband Internet Access

Many options exist today for *private* connectivity between an Enterprise branch office and the core of an Enterprise network:

- leased lines,
- Frame Relay,
- ATM,
- MPLS VPNs,
- Metro Ethernet.

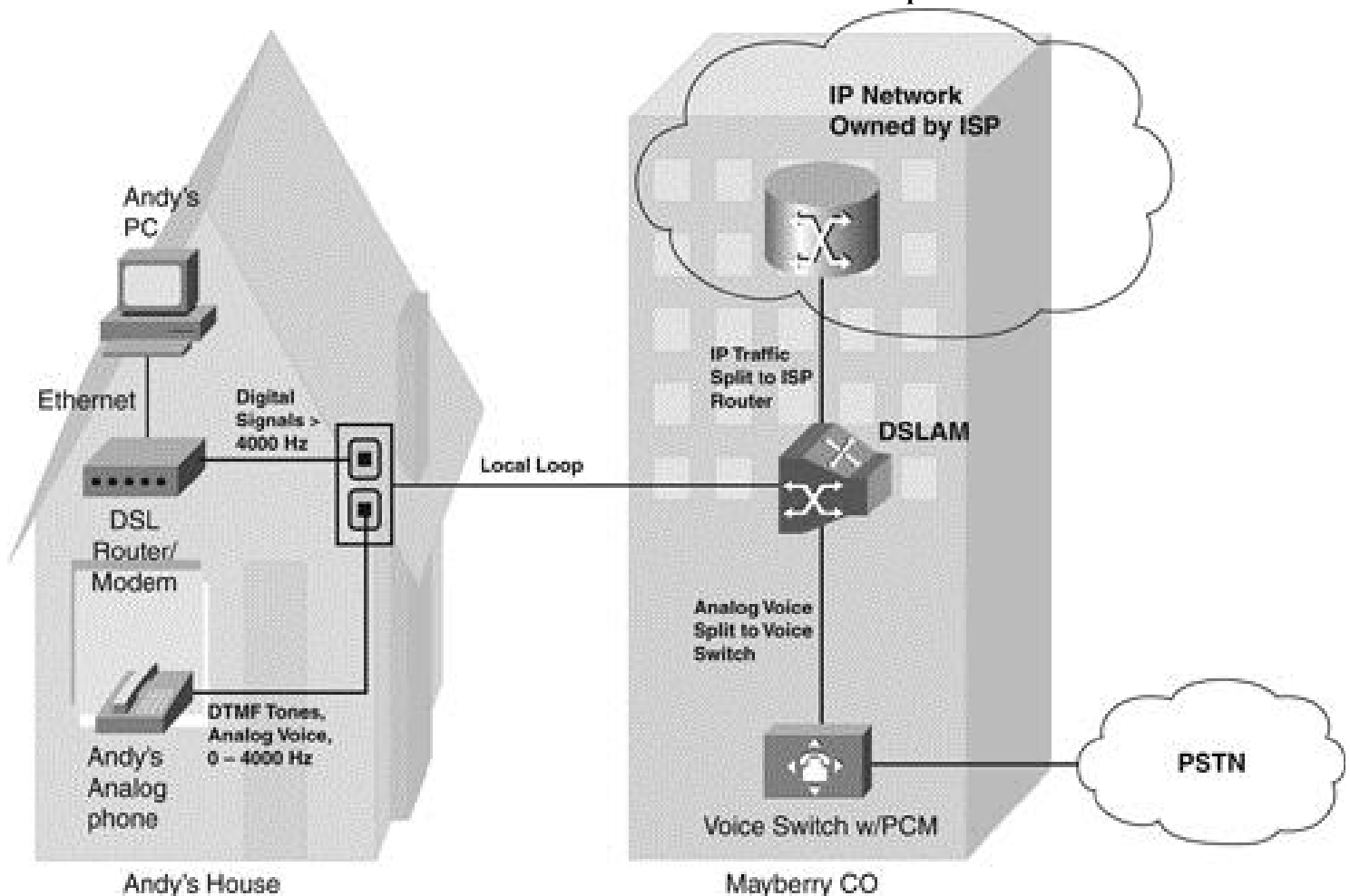
➤ **AToM: Any Transport over MPLS; it unifies layers 2 and 3 over a common IP/MPLS infrastructure.**

Several other *public* options exist for branch office connectivity. All these options use the *Internet* for connectivity between the branch office and the core of the Enterprise network:

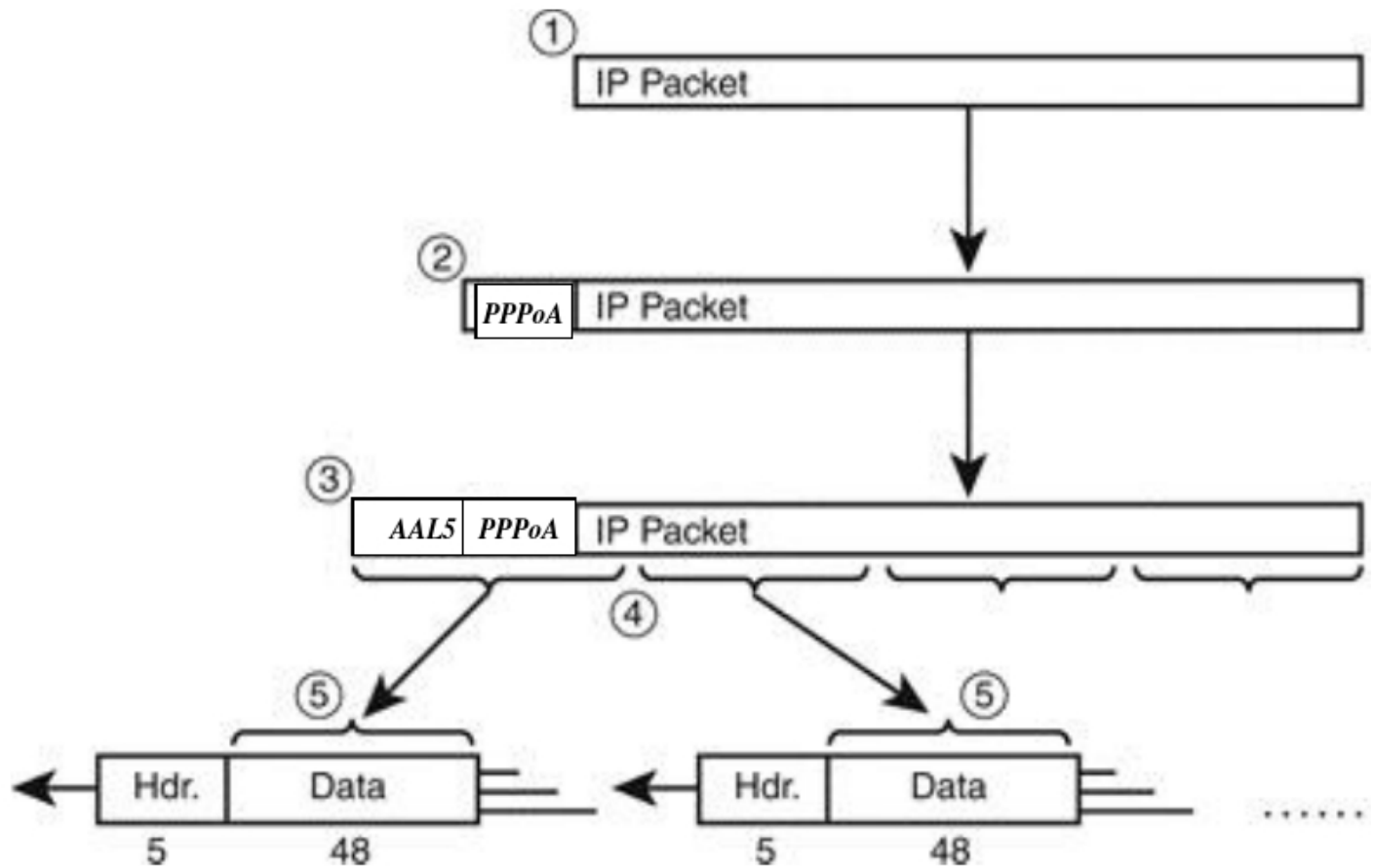
- *digital subscriber line (DSL)*,
- *cable*,
- **FTTx**,
- wireless broadband, (**3G/4G**).

8.1.1 DSL

DSL at the Home and the Telco Local Loop

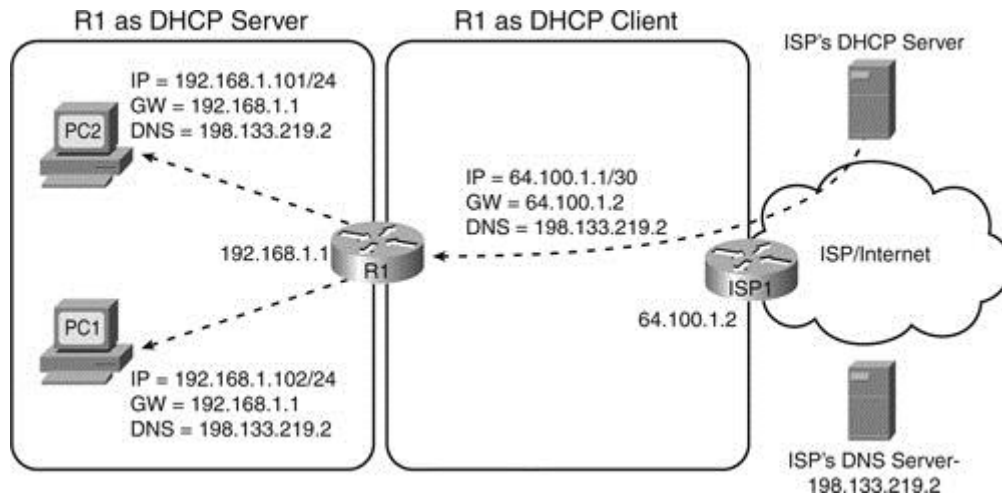


ATM Encapsulation and Segmentation



8.2 Dynamic Address Allocation

Branch Office Router Acting as DHCP Client and DHCP Server

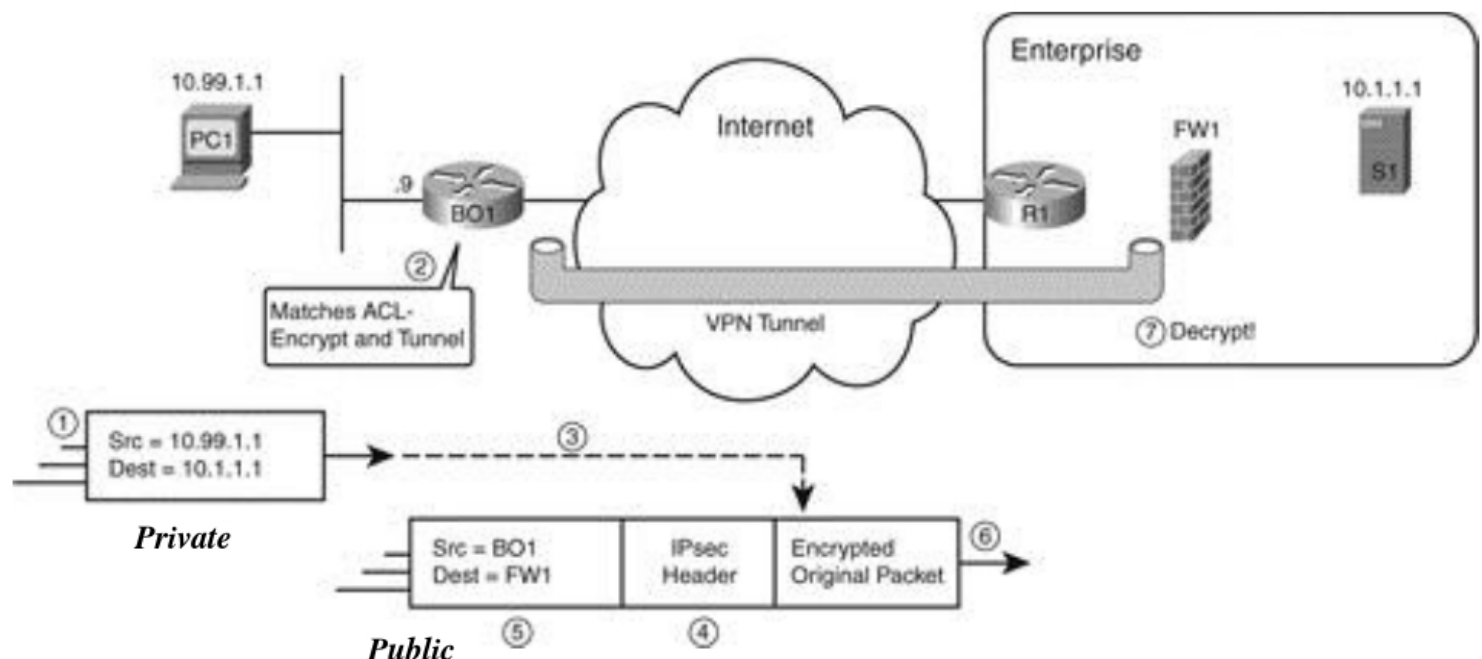


8.3 Securing Communications

8.3.1 IPsec VPN

IPsec is a network layer protocol suite for encrypting IP packets between two hosts and thereby creating a secure “tunnel.” The IETF defined IPsec in [RFC 4301](#). [RFC 6071](#) details all the RFCs related to IPsec/IKE.

IPsec uses open standards and provides secure communication between peers to ensure data confidentiality, integrity, and authenticity through network layer encryption. IPsec connections are commonly configured between firewalls, VPN appliances, or routers.



Two key differences with previous **IPv6** → **IPv4** tunnels :

- First, this tunnel uses IPv4 as both the transport protocol and passenger protocol, rather than using IPv4 to transport IPv6 as the passenger protocol.
- Also, this tunnel uses additional security features, like the encryption.

8.3.1.1 IPsec Protocols

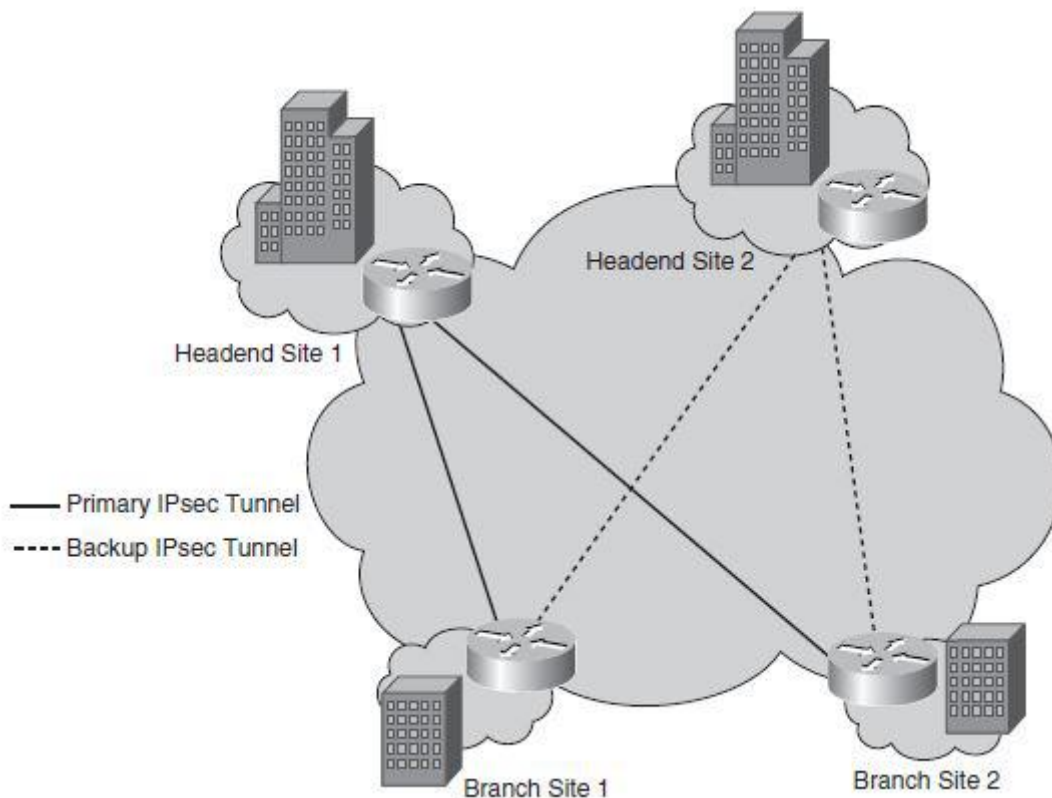
- **ISAKMP** (Internet Security Association and Key Management Protocol): IPsec uses symmetrical encryption algorithms to provide data protection. These algorithms need a secure method to exchange keys to ensure that the data is protected. Internet Key Exchange - Diffie-Hellman (**IKE-DHn**) ISAKMP protocols provide these functions and are used to set up a **security association (SA)**.
- **ESP** (Encapsulating Security Payload): it is used to provide *confidentiality* (**DES, 3DES, AES 128/192/256**), data origin *authentication* (**RSA, 1 public and 1 private keys of at least 2048 bits nowadays**), connectionless integrity (**HMAC-MD5, HMAC-SHA1/256/512**), and anti-replay services. **PKI** (Public Key Infrastructure) is an arrangement that provides for third-party verification of certificates for RSA.
- **AH** (Authentication Header): it is used to provide integrity and data origin authentication (stronger than with ESP), usually referred to as just authentication.

8.3.1.2 IPsec Direct Encapsulation

IPsec provides a tunnel mode of operation that enables it to be used as a standalone connection method and is the most fundamental VPN design model. When you are using IPsec direct encapsulation, **IP multicast and thus dynamic routing protocols are not supported**.

The **headend IPsec** terminating device needs to use *static IP addressing*, but the **remote IPsec** endpoints can use *static or dynamic IP addressing*. Redundancy can be provided at the headend by using multiple IPsec terminating devices, and each remote IPsec endpoint can be populated with a list of headend endpoints to make connections with.

IPsec packet payloads can be encrypted, and IPsec receivers can authenticate packet origins.



Different alternatives exist for supporting **IGPs** over an IPsec tunnel. These options include:

- **GRE Tunnel Interface: Hub-and-Spoke VPN**
- **Virtual Tunnel Interfaces: Hub-and-Spoke VPN:** Similar in concept to GRE tunnels but it uses an encapsulation that does not add an extra 4-byte header. (GRE adds such a header.)
- **Dynamic Multipoint VPN (DMVPN): Site-to-Site VPN:** Creates a multipoint VPN concept, allowing less configuration to add new sites.
- **Group Encrypted Transport (GET) VPN: Any-to-Any VPN:** A more recent addition to IOS, also supporting multipoint VPNs with less configuration to add new sites.

8.3.2 GRE Over IPsec

Generic Routing Encapsulation was developed by Cisco to encapsulate a variety of protocols inside IP tunnels.

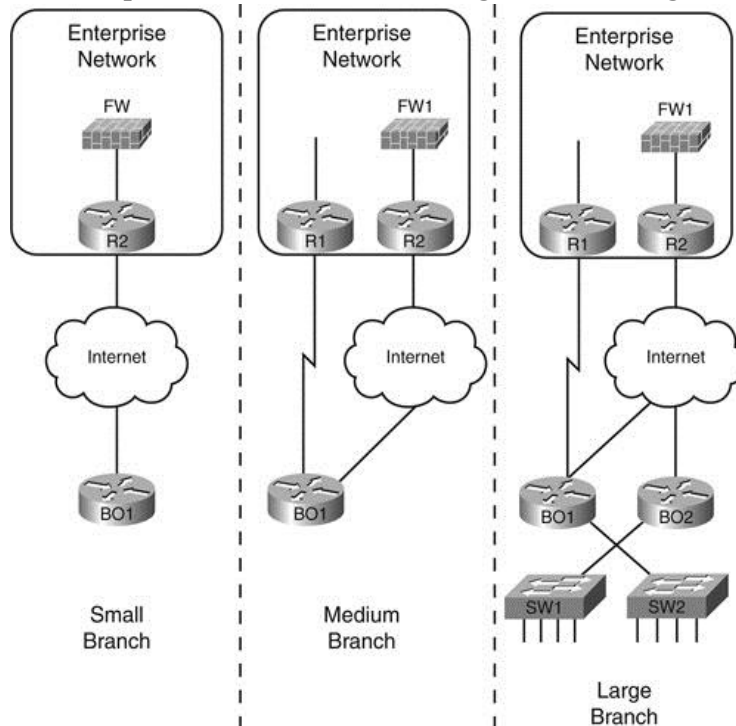
Using IPsec with GRE tunnels provides for secure VPN tunnels by encrypting the GRE tunnel:

- support for **IP multicast, hence dynamic IGP routing protocols, and non-IP protocols**.
- support for **QoS policies** and **deterministic routing metrics** for headend IPsec termination points.
- Because all the primary and backup GRE over IPsec tunnels are preestablished, there is built-in redundancy to support failure scenarios.

The IP addressing for the remote sites can have dynamic or static addressing, but the headend site requires static IP addressing. Primary tunnels can be differentiated from backup tunnels by modifying the routing metrics slightly to prefer the one or the other.

8.4 Branch Routing

Example Small, Medium, and Large Branch Designs

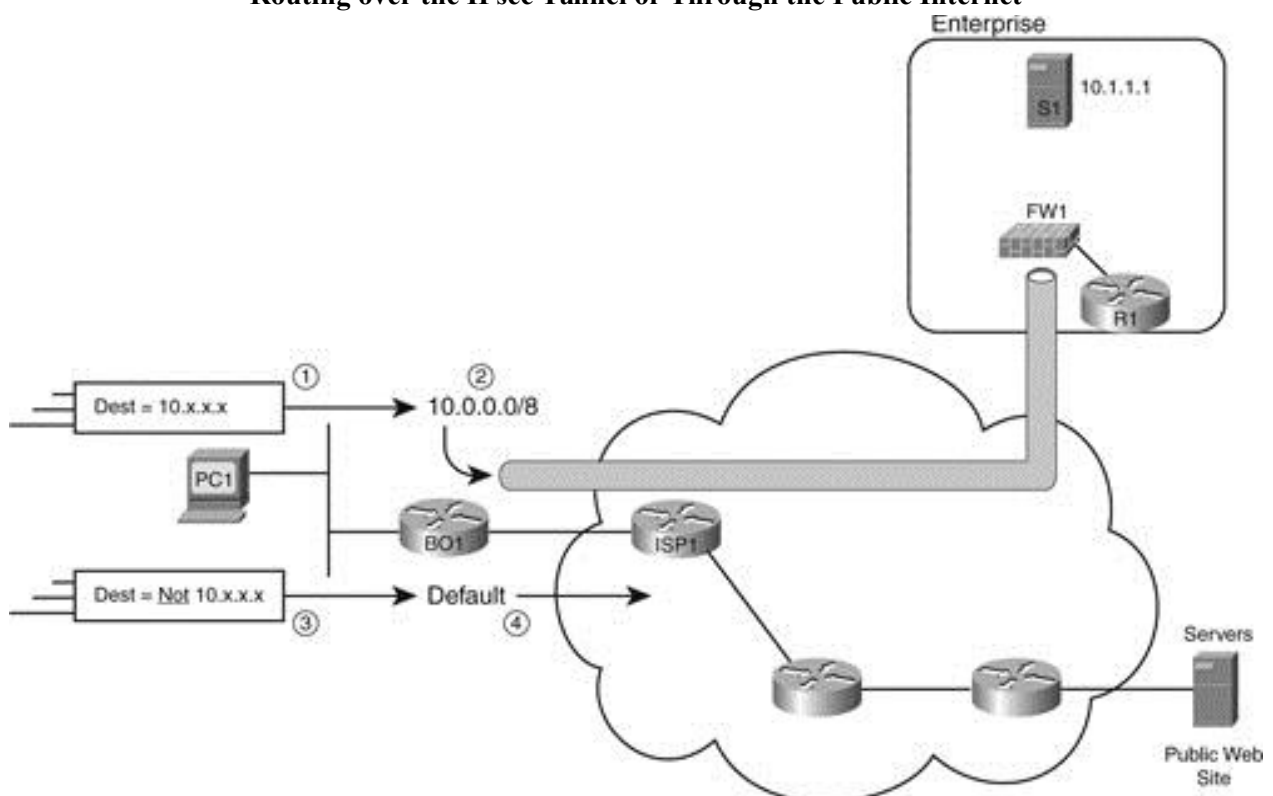


8.4.1 Small Branches

For the *small* branch design, most of the implementation challenges fall into two main categories:

- features needed for communication with hosts in the public Internet. It includes the Internet access details—how to make DSL, cable, and so on work—plus services likely to be required to learn, allocate, and translate the public IP addresses learned from the ISP, such as DHCP and NAT.
- features needed to support secure communications with hosts in the Enterprise. It focuses on *virtual private network (VPN)* options that allow the Enterprise to trust that a packet comes from a legitimate branch office, and to prevent attackers from reading the contents of the packet as it crosses the Internet.

Routing over the IPsec Tunnel or Through the Public Internet

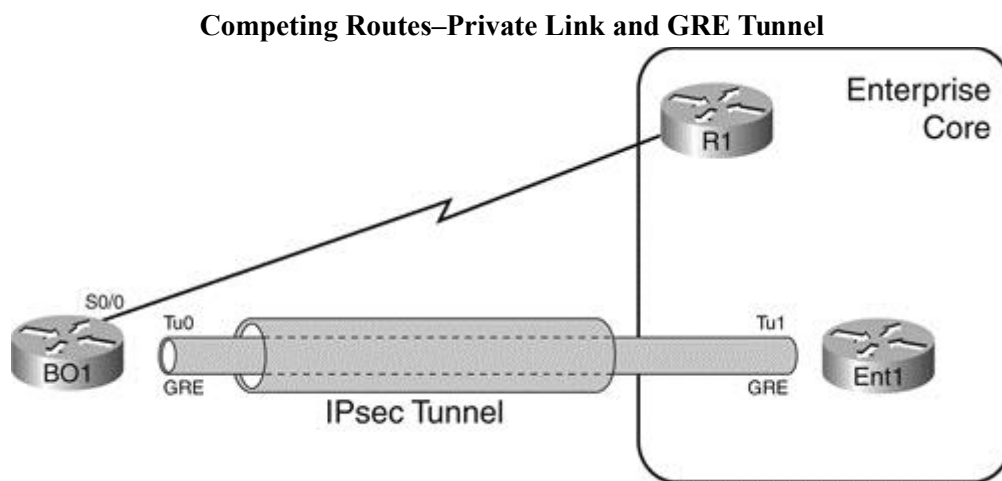


8.4.2

Medium and Large Branches

Focusing on the branch router's logic, two main options exist:

- using *static routes* : they can be used to forward traffic over the private connection and over the IPsec tunnel.
 - Note that to use the floating static route option (***modified AD***), a GRE tunnel would not be required, because the routers do not need to use an IGP through the Internet.
- using *an IGP* : *IPsec does not directly support IGP protocols*, because the IPsec tunnel cannot natively forward IPv4 multicasts. To overcome this restriction, you can use a *GRE tunnel* that actually runs over the IPsec tunnel. GRE supports multicasts by encapsulating them in unicast packets, so GRE supports IGPs. Summarizing, by using GRE, you get the following features:
 - A GRE tunnel acts like a point-to-point link from a Layer 3 perspective.
 - A GRE tunnel supports many passenger protocols, including IPv4.
 - A GRE tunnel encapsulates/forwards broadcasts and multicasts, therefore supporting IPv4 IGPs.
 - GRE tunnels can run through IPsec tunnels.
 - ***If a GRE Tunnel is implemented over IPsec to support an IGP, all traffic exchanged between the Tunnel endpoint routers will be routed inside the GRE Tunnel (not only IGP traffic).***



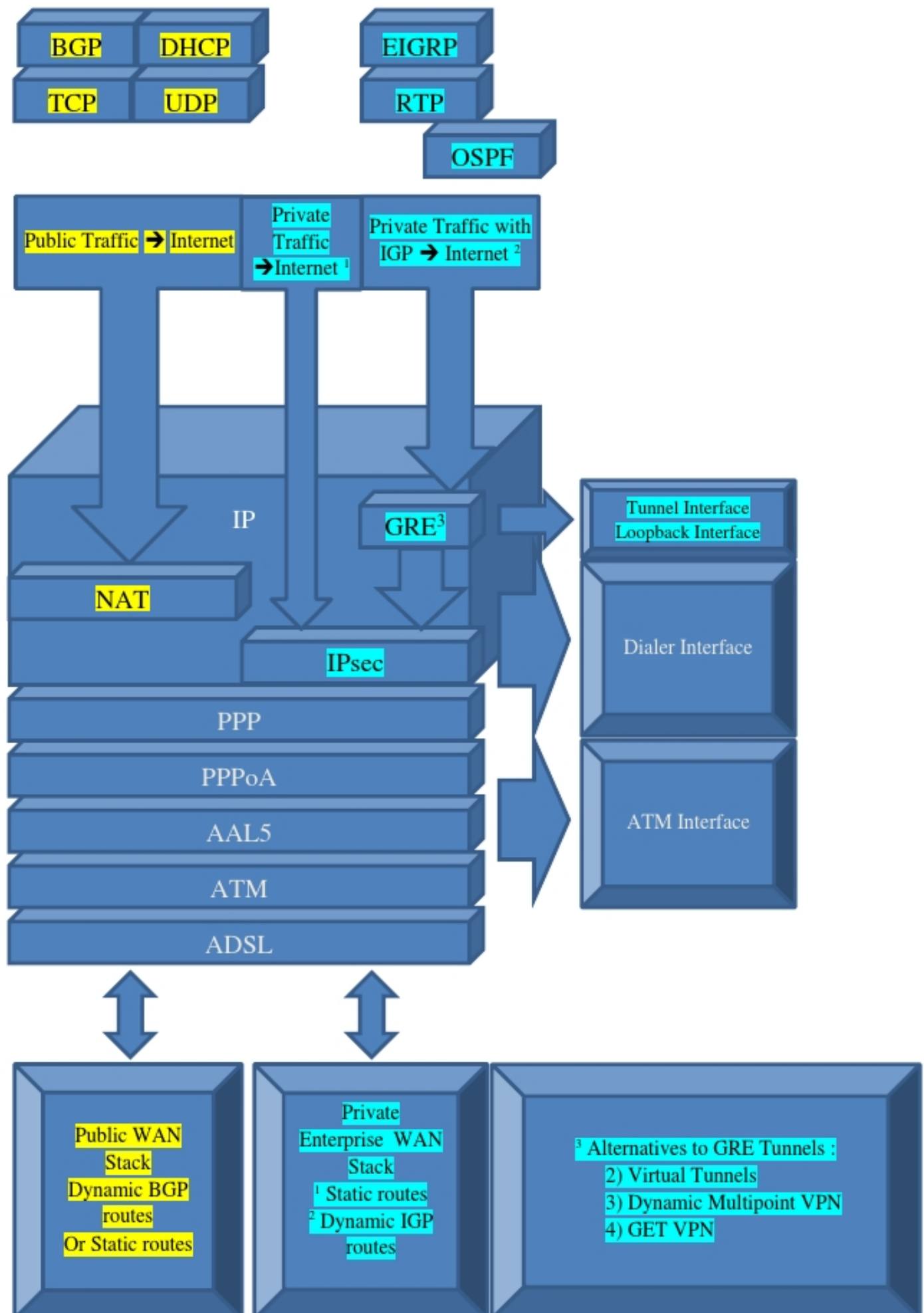
When the branch router receives a packet over the LAN interface, it has several options of how to process the packet. For instance, using the medium-sized branch, which has a leased line into the Enterprise plus a DSL Internet connection, the router could do the following:

- Forward the packet out the leased serial interface, unchanged, to the rest of the Enterprise network
 - ➔ **Private IP addresses**
- Or forward the packet out the tunnel (over the DSL dialer interface), changed somewhat (encrypted, encapsulated, and so on), to the rest of the Enterprise network:
 - ➔ **Public IP addresses encapsulating Encrypted Private addresses (IPsec Tunnel)**
 - Or ➔ **Public IP addresses encapsulating Encrypted Private addresses/GRE (Loopback) encapsulating Encrypted Private addresses (GRE over IPsec Tunnels)**
- Or forward the packet over the Internet link (the DSL dialer interface), after using NAT to change the source private address to a public address, to some public IP destination address
 - ➔ **Public IP addresses**

8.5 Sharing a few Public Addresses between Private Hosts to Access Internet: NAT/PAT

Only the third option requires NAT. Thankfully, NAT configuration easily supports the concept of performing NAT for traffic going to Internet destinations and not performing NAT for traffic in the tunnel.

8.6 Protocol Stack on a router with Internet and VPN connections



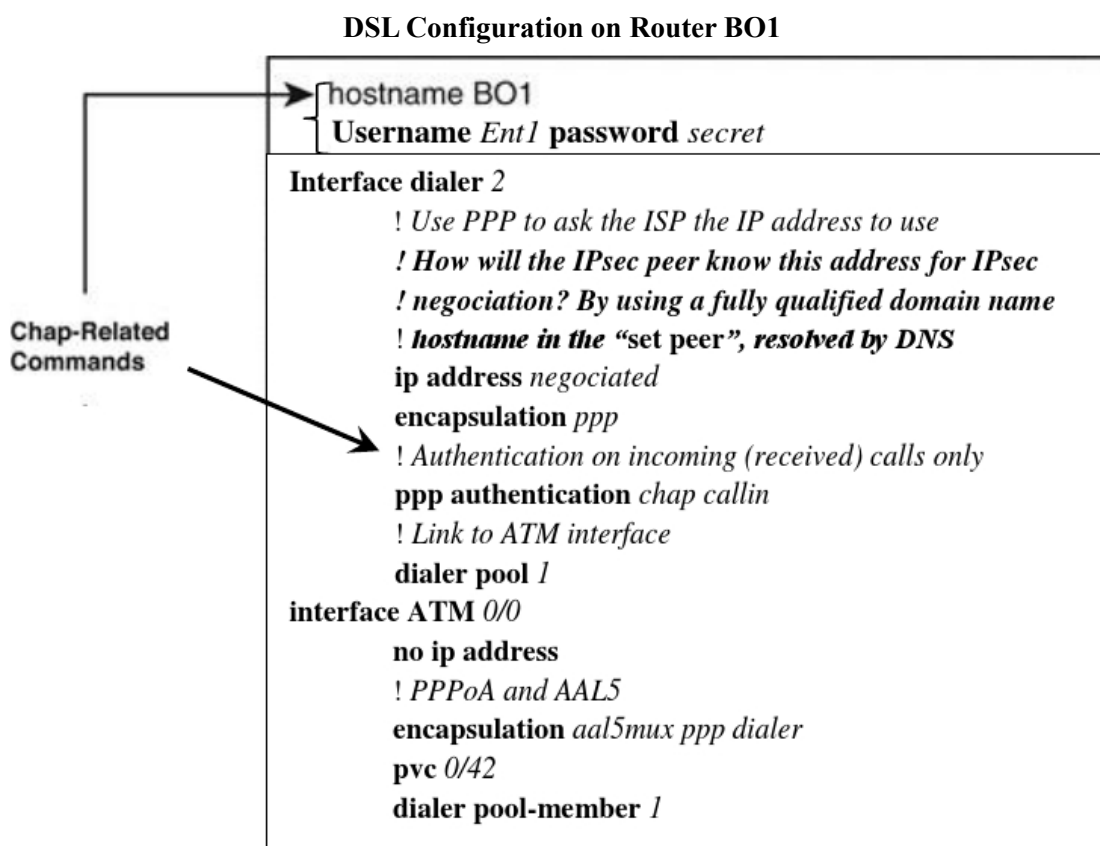
8.7 Configuration Commands

8.7.1 Branch Office Broadband Internet Access

8.7.1.1 PPP/ATM/DSL

The main pieces of the PPP/ATM/DSL configuration as shown in this section are as follows:

- The configuration creates a dialer interface.
- The Layer 3 and PPP configuration is applied to the dialer interface.
- The dialer interface is linked to the ATM interface.
- The ATM configuration is applied to the ATM interface.
- The ATM interface is linked to the dialer interface.
- An IP route forwards traffic out the dialer interface, which triggers the ATM/DSL encapsulation process.



- **ip nat outside** on interface dialer 2

8.7.2 Dynamic Address Allocation: DHCP Server

(config)#service dhcp

Launches the DHCP Server and relay agent

(config)#ip dhcp pool name

Name of DHCP Pool

(dhcp-config)#network ip-address [mask | /prefix-len]

Pool subnet and mask

(dhcp-config)#default-router ip-address

Default router (unallocated address)

(dhcp-config)#dns-server ip-address1 ip-address2 ...

DNS Servers

(config)#ip dhcp excluded-address ip-address [last-ip-address]

IP range of addresses which are excluded from the pool

(config-if)#ip helper-address ip-address

Relay agent: it relays from this interface all UDP broadcasts (DHCP, BOOTP, TFTP, DNS, ...) towards this ip-address server

(config)#ip forward-protocol {udp[port]}

UDP protocols to be forwarded by the relay agent

8.7.3 Sharing a few Public Addresses between Private Hosts to Access Internet: NAT/PAT

The dynamic NAT configuration can leverage *NAT overload*, using a single public IP address—namely, *dialer2*’s dynamically learned IP address, with:

```
(config)#ip nat inside source {list {acl-number | acl-name} | route-map name} {interface int | pool pool-name}
[overload]
```

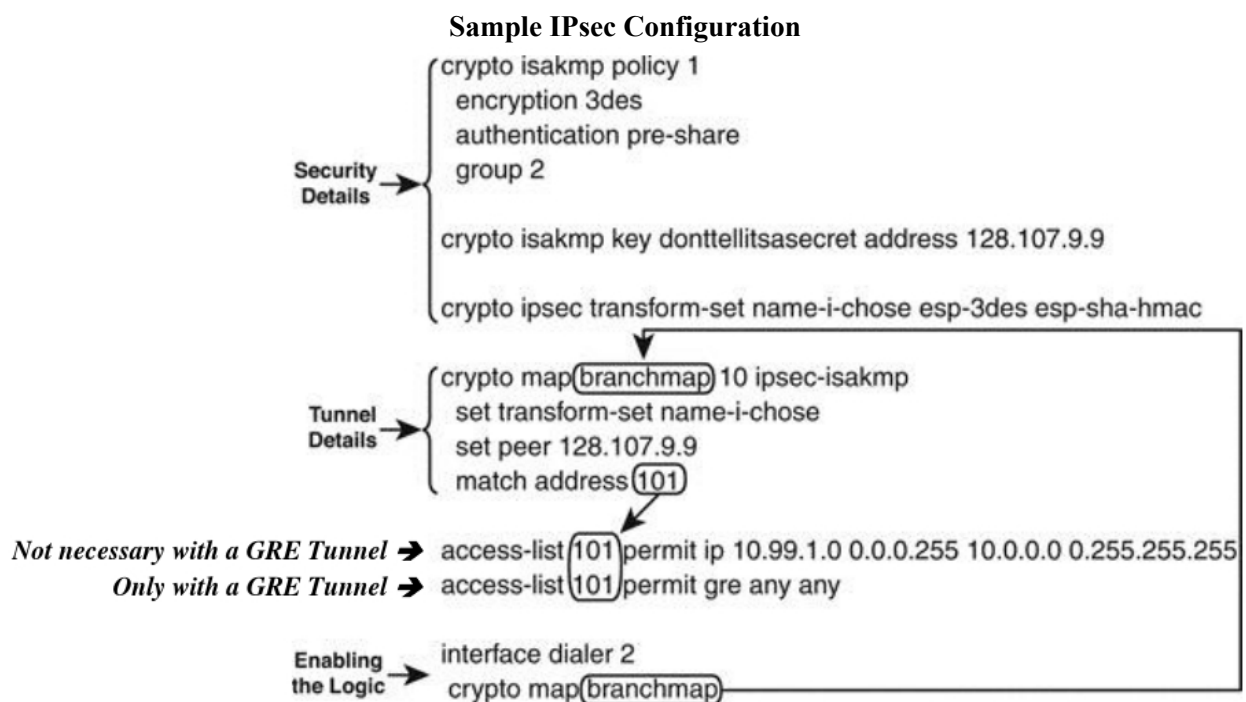
An ACL matches all packets whose source IP address is from the branch’s local LAN subnet *and not destined to the private enterprise core*. This ACL tells the router to “NAT” the traffic permitted by this ACL.

- *To have multiple outside interfaces or pools of public IP addresses, you can use multiple `ip nat inside source list` commands with each time a different ACL, or `ip nat inside source route-map` commands with each time the same ACL: inspired by this [Alain Cadet's post](#) and this other [Alain Cadet's post](#).*
- *Only standard ACL will work with NAT, whether as ‘list’ or inside a ‘route-map’; using extended ACL leads to unwarned failure to ‘nat’ any packet!*

The traffic going through the GRE tunnel will already be encapsulated in a new IP header, and no longer have a source address nor a source port number from the LAN subnet, so only traffic destined for Internet destinations will have NAT/PAT applied. Finally, the interface subcommands `ip nat inside` and `ip nat outside` tell *when* (related to the data stream) and on which interfaces to attempt the translation.

8.7.4 Securing Communications

8.7.4.1 IPsec VPN: ‘Policy-based VPN’



8.7.4.1.1 ISAKMP Policy

(config)#crypto isakmp policy priority

Defines an ISAKMP policy, a set of parameters to be used during the security association negotiation; when the negotiation begins, it tries to find a common policy configured on both peers, starting with the highest priority policies as specified on the remote peer.

This command invokes the Internet Security Association Key Management Protocol (ISAKMP) policy configuration (config-isakmp) command mode.

- *priority: Uniquely identifies the ISAKMP policy and assigns a priority to the policy. From 1 (the highest priority) to 10,000 (the lowest).*

(config-isakmp)# authentication {rsa-sig | rsa-encr | pre-share | ecdsa-sig}

Specifies the authentication method within an ISAKMP policy:

- *rsa-sig : RSA signatures (Default)*
- *rsa-encr : RSA encrypted nonces*
- *pre-share : PSK*
- *ecdsa-sig : Elliptic Curve Digital Signature Algorithm (ECDSA) signature*

(config-isakmp)#group {1 | 2 | 5 | 14 | 15 | 16 | 19 | 20 | 24}

Specifies one or more Diffie-Hellman (DH) group id(s) in an IKE policy :

- *1 : 768-bit DH group (Default)*
- *2 : 1024-bit DH group*
- *5 : 1536-bit DH group*
- *14 : 2048-bit DH group.*
- *15 : 3072-bit DH group.*
- *16 : 4096-bit DH group*
- *19 : 256-bit elliptic curve DH (ECDH) group*
- *20 : 384-bit ECDH group*
- *24 : 2048-bit DH/DSA group*

(config-isakmp)#encryption {des | 3des | aes | aes 192 | aes 256}

Specifies the encryption algorithm within an ISAKMP policy :

- *des : 56-bit Data Encryption Standard (Default)*
- *3des : 3x 56-bit Data Encryption Standard*
- *aes : 128-bit Advanced Encryption Standard*
- *aes 192 : 192-bit AES*
- *aes 256 : 256-bit AES*

There might be a Performance Issue!

(config-isakmp)#hash {sha | sha256 | sha384 | sha512 | md5}

Specifies the hash algorithm within an ISAKMP policy (code length must at least = 2 x encryption key length) :

- *sha : SHA-1 160-bit (HMAC variant), (Default)*
- *sha256 : SHA-2 family 256-bit (HMAC variant)*
- *sha384 : SHA-2 family 384-bit (HMAC variant)*
- *sha512 : SHA-5 family 512-bit (HMAC variant)*
- *md5 : MD5 128-bit (HMAC variant)*

(config-isakmp)#lifetime seconds

Specifies the lifetime of an ISAKMP security association (SA):

- *seconds : Number of seconds for each SA that should exist before expiring. Use an integer from 60 to 86,400 s, (Default, 1 day)*

(config)#crypto isakmp key enc-type-digit keystring {address public-peer-address [mask] | hostname hostname}

Configures a pre-shared authentication key (in conjunction with a previous “(config-isakmp)#authentication pre-share”):

- *enc-type-digit : whether the password to be used is encrypted or unencrypted.*
 - *0 : an unencrypted password follows.*
 - *6 : an encrypted password follows.*
- *keystring : the preshared key*
- *public-peer-address : the IP address of the remote peer*

8.7.4.1.2 IPsec Crypto Map

(config)#crypto map map-name seq-num [ipsec-manual | ipsec-isakmp]

<https://t.me/learningnets>

Enters crypto map configuration mode and creates or modify a crypto map entry :

- *ipsec-manual* : Indicates that ISAKMP will not be used to establish the IPsec security associations (SAs) for protecting the traffic specified by this crypto map entry.
- *ipsec-isakmp* : Indicates that ISAKMP will be used to establish the IPsec SA for protecting the traffic specified by this crypto map entry.

Crypto maps provide two functions:

- *filtering* : it affects the flow of traffic on an interface,
- *classifying the traffic to be protected and defining the policy to be applied to that traffic* : it affects the negotiation performed (via IKE) on behalf of that traffic.

IPsec crypto maps define the following:

- *What traffic should be protected*
- *To which IPsec peers the protected traffic can be forwarded--these are the peers with which an SA can be established*
- *Which transform sets (encryption and integrity types with which protocol) are acceptable for use with the protected traffic*
- *How keys and SAs should be used or managed (or what the keys are, if ISAKMP is not used)*

➤ *Two methods for applying a crypto map:*

1) *On the physical interface,*

- *for policy-based VPN (only IPsec), with Routing Protocols not supported:*
crypto-map-acl: permit ip local-subnet wildcard remote-subnet wildcard,
- *for route-based VPN (GRE encapsulated into IPsec), with Routing Protocols supported:*
crypto-map-acl: permit gre host GRE-Tunnel-src host GRE-Tunnel-dst,

2) *On the GRE tunnel interface,*

- *for route-based VPN (GRE encapsulated into IPsec), with Routing Protocols supported*
(config)#crypto ipsec profile profile-name
(ipsec-profile)#set security-association lifetime seconds number-of-seconds
(ipsec-profile)#set transform-set transform-set-name
(config)#interface Tunnel n
(config-if)#tunnel protection ipsec profile profile-name

(config-crypto-map)#match address [extended-access-list-id | extended-access-list-name]

Specifies an extended access list for a crypto map entry.

➤ *If the traffic does not match a permit entry in any crypto map entry, it will be forwarded without any IPsec security.*

(config-crypto-map)# set peer {host-name [dynamic] [default] | public-ip-address [default]}

Specifies an IPsec peer in a crypto map entry.

- *host-name* : Specifies the IPsec peer by its hostname. This is the peer's hostname concatenated with its domain name,
- *dynamic* : The hostname of the IPsec peer will be resolved via a domain name server (DNS) lookup right before the router establishes the IPsec tunnel,
- *default* : If there are multiple IPsec peers, designates that the first peer is the default peer,
- *public-ip-address* : Specifies the IPsec peer by its IP address.

(config-crypto-map)# set transform-set transform-set-name [transform-set2...transform-set6]

Specifies which transform sets can be used with the crypto map entry.

(config)#crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4]

<https://t.me/learningnets>

Defines a transform set--an acceptable combination of security protocols and algorithms :

- *transform-set-name : Name of the transform set to create*
- *transform1[transform2] [transform3] [transform4] : Type of transform set. You may specify up to four “transforms”: one Authentication Header (AH) integrity, one Encapsulating Security Payload (ESP) encryption, one ESP integrity, and one compression. These transforms define the IP Security (IPSec) security protocols and algorithms. Accepted transform values are described in the table below.*

After issuing this command, you are put into the (cfg-crypto-transform) mode. While in this mode, you can change the mode to tunnel or transport with the mode [tunnel | transport] command.

<i>Transform Type</i>	<i>Transform</i>	<i>Description</i>
<i>AH Integrity Transform</i> <i>>Pick only one</i>	<i>ah-md5-hmac</i> <i>ah-sha-hmac</i> <i>ah-sha256-hmac</i> <i>ah-sha384-hmac</i> <i>ah-sha512-hmac</i>	<i>AH with MD5 128 bits (Message Digest 5)</i> <i>AH with the SHA 160 bits (Secure Hash Algorithm)</i> <i>AH with the SHA 256 bits</i> <i>AH with the SHA 384 bits</i> <i>AH with the SHA 512 bits</i>
<i>ESP Integrity Transform</i> <i>>Pick only one</i>	<i>esp-md5-hmac</i> <i>esp-sha-hmac</i> <i>esp-sha256-hmac</i> <i>esp-sha384-hmac</i> <i>esp-sha512-hmac</i>	<i>ESP with MD5 128 bits (Message Digest 5)</i> <i>ESP with the SHA 160 bits (Secure Hash Algorithm)</i> <i>ESP with the SHA 256 bits</i> <i>ESP with the SHA 384 bits</i> <i>ESP with the SHA 512 bits</i>
<i>ESP Encryption Transform</i> <i>>Pick only one</i>	<i>esp-des</i> <i>esp-3des</i> <i>esp-aes</i> <i>esp-aes 192</i> <i>esp-aes 256</i> <i>esp-gcm 128 256</i> <i>esp-gmac 128 256</i> <i>esp-seal</i> <i>esp-null</i>	<i>ESP with 56-bit Data Encryption Standard (DES)</i> <i>ESP with 3x 56-bit DES</i> <i>ESP with the 128-bit Advanced Encryption Standard</i> <i>ESP with the 192-bit AES</i> <i>ESP with the 256-bit AES</i> <i>ESP with Galois/Counter Mode (Authentication and Encryption)</i> <i>ESP with Galois Message Authentication Code (Authentication)</i> <i>ESP with the 160-bit Software-Optimized Encryption Algorithm</i> <i>Null encryption algorithm</i>
<i>IP Compression Transform</i>	<i>comp-lzs</i>	<i>IP compression with the Lempel-Ziv-Stac (LZS) algorithm</i>

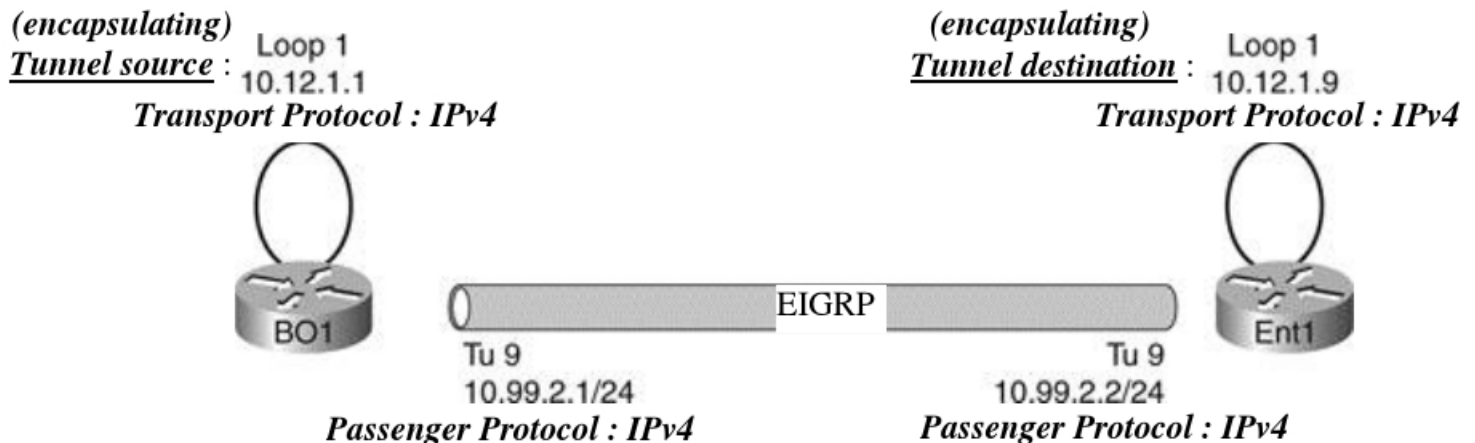
(config-if)#crypto map map-name *Applies a previously defined crypto map set to a physical interface.*

➤ *The Tunnel remains DOWN as long as the IPSec flow has not yet been processed.*

8.7.4.2 GRE Tunnel: 'Route-based VPN'

In this case, the GRE tunnel carries IPv4 as the passenger protocol, inside an IPv4 packet. *Be careful with neighborhood issues with different subnet numbers when running an IPv4 IGP inside the tunnel → the GRE tunnel interfaces on both routers must be in the same subnet.*

GRE Tunnel Topology and Addresses



The configuration also requires two main branches of logic for routing to work correctly:

- First, for the tunnel to function, the **tunnel destination** must be reachable; in our situation, a **static route** out **interface dialer 2 (the one with the IPsec Tunnel)** should be added for this purpose (*and it triggers the ISAKMP SA opening*): **do not rely on the IGP to exchange a route for the distant loopback interface through the tunnel : this would cause a recursive routing loop problem ! The IGP multicasts are encapsulated inside unicast; to be able to send these unicasts, they both need a route... that they don't have. So they cannot form neighborhood nor send updates to each other. So the static route is necessary to avoid an oscillation for the tunnel interface status between UP and DOWN!**
- Additionally, the routers need to exchange routes that will list the tunnel interface as the outgoing interface, which in turn directs packets through the tunnel. In our example, EIGRP is enabled on tunnel 9 just as a reminder that one of the primary motivations for bothering with the GRE tunnel is to support IGP routing protocols.
- **Supported only on Cisco IOS.**
- **There's a trap regarding traffic going through a GRE/VPN Tunnel on a NAT outside interface: if GRE traffic is sent on an interface - where NAT is active - to be then encapsulated into the IPsec Tunnel, we must be cautious about the fact that the NAT ACL is processed before the crypto ACL; you certainly don't want the GRE traffic to have its source IP address changed into a public one!**

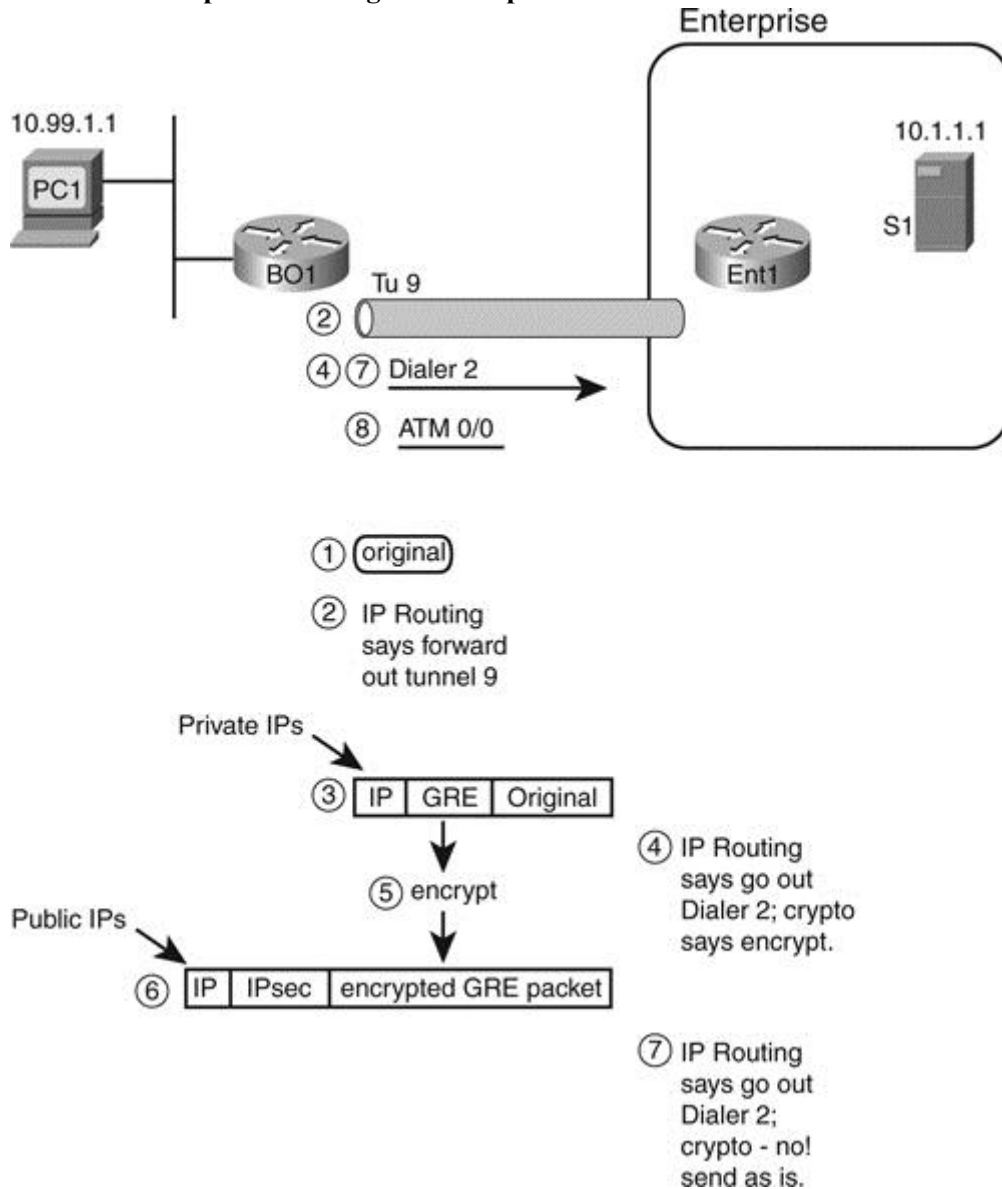
8.7.4.3 IPsec DMVPN

Out of CCNP ROUTE scope (at the time of last edition): Configuration on [Configuring Dynamic Multipoint VPN \(DMVPN\)](#) using GRE over IPSec between Multiple Routers (phase II)

- **The command "ip nhrp map multicast dynamic" is needed only on the hub.**

8.7.5 Summary

Example of Routing and Encapsulation with GRE over IPsec



➤ Traffic from the LAN is routed out on GRE Tunnel Interface, like IGP traffic.

8.8 Verification Commands

VPN

#show crypto session [**brief** | **detail**] | [**local** | **remote**] [*ip-address* | *ipv6-address*]

Displays status information for active crypto sessions.

#show crypto isakmp policy

Displays the parameters for each ISAKMP policy.

#show crypto isakmp sa [**detail**]

Displays current ISAKMP security associations (SAs).

#show crypto map

Displays the crypto map configuration.

#show crypto ipsec transform-set

Displays the configured transform sets or active default transform sets.

#show crypto ipsec sa [**map** *map-name* | **interface** *type number* | **peer** *address* | [**detail**] [**address**] [**active** | **standby**]

Displays the *settings* and various *counters* used by current *IPsec* security associations (SAs):

- **map** : Displays any existing SAs *settings* that were created for the crypto map set with the value for the *map-name* argument
- **interface** : Displays all existing SAs *settings* created for the interface value provided in the *interface* argument
- **peer** : Displays all existing SAs settings with the peer address
- **detail** : Detailed error counters.
- **address** : Displays all existing SAs, sorted by the destination address (either the local address or the address of the IP security (IPsec) remote peer) and then by protocol (Authentication Header [AH] or Encapsulation Security Protocol [ESP])
- **active** : Displays high availability (HA) (???) - enabled IPsec SAs that are in the active state
- **standby** : Displays HA-enabled IPsec SAs that are in the standby state

#clear crypto isakmp [*connection-id*]

Clears active ISAKMP connections.

#clear crypto sa

Clears IPSec Security Associations (SAs).

#clear crypto session [*local ip-address*] [*remote ip-address*]

Clears active ISAKMP and IPsec SAs

#debug crypto isakmp

Displays messages about ISAKMP events.

#debug crypto ipsec

Shows the source and destination of IPsec tunnel endpoints.

NAT

#debug ip nat

Can be very useful to discover which traffic is natted and which is not ...