

# CCNP BSCI Notes

30 Mar 2008

## Chapter 1: Network Design

Full mesh formula:  $n(n-1)/2$

Example: To create a mesh between 8 nodes,  $8 * (8-1) / 2 = 28$ ; 28 links are needed.

### Design concepts

#### Legacy Hierarchical Design Model

Core - fast L2-switched backbone

Distribution - L3 switches

Access - dense L2 switches

This failed to address issues such as redundancy, Internet and remote access, and locating services.

*Switch block* design was introduced to add redundancy; this included redundant core and distribution switches and links per switch block.

#### Enterprise Composite Network Model

This new model was developed to address modern design considerations.

Enterprise campus

Campus backbone (previously the "core")

Building distribution (previously "distribution")

Building access (previously "access")

Management

Server farm (for internal enterprise services)

Enterprise edge

E-commerce

Internet connectivity

Remote access

WAN (internal links)

Service provider edge

ISP

PSTN

Frame relay, ATM, PPP for private connectivity

## **Intelligent Information Network (IIN)**

Phase 1: Integrated transport - the shift toward the ECN model

Phase 2: Integrated services - service virtualization (disassociation of services from individual machines)

Phase 3: Integrated applications - recognizing and combining high-layer traffic properties (for example, Network Access Control)

## **Services-Oriented Network Architecture (SONA)**

SONA is the application of IIN ideas to enterprise networks.

Network infrastructure (IIN phase 1)

Interactive services (IIN phase 2)

Application (IIN phase 3)

## **Routing protocols**

Distance vector - a router will only exchange routes with a directly connected neighbor

Link-state - a router advertises a list of all its neighbors and its neighbors' networks; routers run SPF to determine the best path

## **Chapter 3: EIGRP Principles**

Supports routed protocols like IP and IPv6 via protocol-dependent modules

Uses Reliable Transport Protocol (RTP, Cisco proprietary) for some traffic (updates, queries, and replies)

Uses hellos to identify/monitor neighbors

Uses the Diffusing Update Algorithm (DUAL) to select routes

EIGRP is IP protocol 88.

EIGRP supports proportional unequal-cost load-balancing among feasible routes.

## **Packet types**

**Hello** - Identify neighbors, sent as periodic multicasts

**Update** - Advertises routes, only sent when there is a change, multicast to 224.0.0.10

**Ack** - Acknowledges receipt of an update

**Query** - Used to query routes from neighbors (multicast; unicast attempted up to 16 times if multicast gets no response)

**Reply** - Used to answer a query (unicast)

## Metric calculation

Metric =  $256 * (K_1 * \text{bandwidth} + ( (K_2 * \text{bandwidth}) / (256 - \text{load}) ) + K_3 * \text{delay} ) * ( K_5 / (\text{reliability} + K_4) )$

K values are used to distribute weight to different path aspects:

bandwidth - Defined as  $10^7$  divided by the speed of the slowest link in the path, in Kbps

load - 8-bit value, not considered by default

reliability - 8-bit value, not considered by default

delay - constant value associated with interface type; EIGRP uses the sum of all delays in the path

K defaults:  $K_1 = 1, K_2 = 0, K_3 = 1, K_4 = 0, K_5 = 0$

K values can be manipulated by an admin, but routers must have matching K values to become neighbors

## DUAL

Advertised distance - Cost advertised by a neighbor to get to a destination

Feasible distance - Advertised distance + cost get to that neighbor

The *feasibility requirement* states "if my neighbor's advertised distance is less than my feasible distance, the path will be loop free."

Successor - The neighbor with the best path

Feasible successor - All other neighbors which meet the feasibility requirement

Split-horizon - A network is not advertised on the link from which it was learned.

## Queries

When a router loses its successor and has no feasible successors, it will query all remaining neighbors for

a new route. Queries are recursive and will be forwarded to other neighbors until either a route is found, or a summarization boundary is reached.

Stuck in Active (SIA) - Queries which do not return a route before the active timer expires (usually 3 minutes), the router is considered stuck in active mode.

## EIGRP Tables

### Neighbor table

Stores information about neighboring EIGRP routers:

- Network address (IP)

- Connected interface

- Holdtime - how long the router will wait to receive another HELLO before dropping the neighbor; default = 3 \* hello timer

- Uptime - how long the neighborship has been established

- Sequence numbers

- Retransmission Timeout (RTO) - how long the router will wait for an ack before retransmitting the packet; calculated by SRTT

- Smooth Round Trip Time (SRTT) - time it takes for an ack to be received once a packet has been transmitted

- Queue count - number of packets waiting in queue; a high count indicates line congestion

### Topology table

Holds *all* routes received from neighbors, is built from updates, calculated by DUAL, and contains all the information required by the routing table

### Routing table

Route types:

- Internal - Paths directly within EIGRP

- Summary - Internal paths which have been summarized

- External - Routes redistributes into EIGRP

## Chapter 4: Scalable EIGRP

### Configuring EIGRP

Enabling EIGRP:

---

```
Router(config)# router eigrp <ASN>
```

---

Adding a network to EIGRP:

---

```
Router(config-router)# network <network> [wildcard mask]
```

---

All interfaces belonging to the specified network will become EIGRP capable and advertised via EIGRP.

EIGRP will summarize to classful boundaries by default.

An interface can be set to "passive" to prevent advertisement:

---

```
Router(config-router)# passive-interface <interface>
```

---

If an interface is included in the network configuration but set to passive, it will still be included in EIGRP advertisements. If an interface is not included in an EIGRP network, it will not be advertised in EIGRP.

## Summarization

Summarization is key to preventing looping route queries (which leads to SIA condition).

Disable automatic summarization to classful addresses:

---

```
Router(config-router)# no auto-summary
```

---

Configure manual summarization on an interface:

---

```
Router(config-if)# ip summary-address eigrp <ASN> <network> <subnet mask>
```

---

## Stub Routers

Stub routers participate minimally in EIGRP, which reduces utilization of memory and CPU on the router.

Often used in lieu of static routes in a hub-spoke topology to provide a standard configuration for all spoke routers.

---

```
Router(config-router)# eigrp stub [options]
```

---

receive-only - Prevents the router from advertising any routes

connected - Permits advertisement of connected networks (default)

static - Permits redistribution of static routes

summary - Advertises summary routes (default)

## Load Balancing

EIGRP automatically load balances across equal-cost links.

Unequal-cost load balancing is enabled by specifying a *variance*. Feasible successors with a feasible distance less than (best path FD \* variance) will be proportionally utilized.

---

```
Router(config-router)# variance <multiplier>
```

---

Variance can be between 1 and 128; default is 1 (equal-cost only).

## EIGRP Tuning

EIGRP timers do not have to match for two routers to form an adjacency.

Default hello timer:

High-bandwidth links ( $\geq T1$ ): 5 seconds

Low-bandwidth links ( $< T1$ ): 60 seconds

Hello timers are modified per interface:

---

```
Router(config-if)# ip hello-interval eigrp <ASN> <seconds>
```

---

Default hold timer = 3 \*

Reconfiguring the hold timer is done in seconds, not as a multiplier:

---

```
Router(config-if)# ip hold-time eigrp <ASN> <seconds>
```

---

## Authentication

Authentication types:

Plain-text password (insecure)

MD5 hash

Configuring MD5 authentication on an interface:

---

```
Router(config-if)# ip authentication mode eigrp <ASN> md5
Router(config-if)# ip authentication key-chain eigrp <ASN> <chain name>
Router(config)# key chain <chain name>
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string <password>
```

---

## Bandwidth Manipulation

Bandwidth of an interface can be administratively configured:

---

```
Router(config-if)# bandwidth <kilobits>
```

---

The bandwidth setting will not affect the actual interface speed, only the value used in metric calculation. This is useful for serial interfaces which connect to a DSU operating at a slower speed.

EIGRP assumes all virtual circuits on an interface receive an equal share of the total bandwidth.

By default, EIGRP will use no more than 50% of an interface's bandwidth. This can be adjusted as such:

---

```
Router(config-if)# ip bandwidth-percent eigrp <ASN> <percentage>
```

---

## Verifying EIGRP Operation

### Monitoring

```
show ip eigrp neighbors
show ip eigrp topology
show ip eigrp traffic
show ip eigrp interfaces
show ip route eigrp
```

## **Debugging**

```
debug eigrp packet
debug ip eigrp neighbors
debug ip eigrp
debug ip eigrp summary
```

## **Chapter 5: Understanding Simple Single-Area OSPF**

### **OSPF Fundamentals**

Link-state routing protocols utilize more internal resources in favor of reducing bandwidth consumption.

All OSPF routers in an area share the same Link State Database (LSDB).

Link State Advertisements (LSAs) are flooded to all neighboring routers.

OSPF tables:

- Neighbor table
- Topology database
- Routing table

### **Forming adjacencies**

Routers multicast hellos to 224.0.0.5 every 10 seconds on a broadcast link and every 30 seconds on a nonbroadcast link.

Once hellos are exchanged, neighboring routers add one another to their neighbor tables.

Contents of a hello packet:

- Router ID - 32-bit unique number (IP address)
- Hello/dead intervals - Timers
- Neighbor list - List of neighboring router IDs
- Area ID
- Priority - Used in electing the DR and BDR
- DR and BDR
- Authentication (if enabled)
- Stub Area Flag - On if this is a stub area

Neighbor states:

Down

Attempt - Used for manually configured neighbors on an NBMA link; unicast hellos sent to neighbor from which hellos have stopped being received

Init - Hello packet received from neighbor, but without the recipient's router ID

2-Way - Bi-directional communication has been established

Exstart - The DR and BDR have been elected, link-state exchange starting

Exchange - Exchange of database descriptor (DBD) packets

Loading - Exchange of link-state information

Full - Full adjacency established

### Example OSPF packet capture

Cisco OSPF will load balance over up to four equal-cost links; configurable up to six.

### Designated Routers

Neighbors on a broadcast segment elect a *designated router (DR)* and *backup designated router (BDR)*, which peer with all other routers on the segment. All non-designated routers peer only with the DR and BDR.

Multicast destinations:

224.0.0.5 - All OSPF routers

224.0.0.6 - All designated OSPF routers (DR and BDR only)

DRs are chosen based on priority (0 - 255). 1 is default; routers with 0 priority will never be elected. Priority ties are broken by choosing the higher router ID.

DRs are elected on point-to-point Ethernet links even though this is unnecessary (Ethernet is always seen as a broadcast medium). Interfaces can be configured to operate in point-to-point mode to prevent this.

(B)DRs are not preempted. New election will take place only when a current (B)DR goes offline or its OSPF process is administratively restarted.

### Areas

All routers in an area maintain an identical topological database.

Areas are defined to logically segment a network and reduce routing table size and complexity.

All areas connect to area 0 (the *backbone area*).

Router types:

Backbone routers - Routers in area 0

Area Border Routers (ABRs) - Routers in multiple areas

Autonomous System Boundary Routers (ASBR) - Routers which redistribute information from another AS

Internal - Routers which have all interfaces in a single area

Routers can fill multiple roles.

## **Packet Types**

OSPF is IP protocol 89.

Hello - Used to establish communication with directly connected neighbors

Database Descriptor (DBD) - Lists router IDs from which the router has an LSA and its current sequence number

Link State Request (LSR) - Request for an LSA

Link State Update (LSU) - Reply to an LSR with the requested information

Link State Acknowledgment (LSAck) - Used to confirm receipt of link-state information

## **Packet Fields**

Version - Version of OSPF being run

Type

Length

Router ID

Area ID

Checksum

Authentication type (none/plain/text/md5)

Authentication data

Data

## **Configuring OSPF in a Single Area**

Necessary information:

OSPF process ID (locally significant)

Participating interfaces

Area ID

Router ID

## Enable OSPF

---

```
Router(config)# router ospf <process number>
```

---

## Configure Included Networks

---

```
Router(config-router)# network <network> <wildcard mask> area <area ID>
```

---

A single interface can be specified by supplying its IP address and a null wildcard mask: `network 192.168.0.1 0.0.0.0 area 0`

## Router ID

If no router ID has been administratively declared, a router will choose the highest loopback IP address. If no loopback addresses are present, the highest IP address of the first active interface will be used.

A router ID can be manually specified:

---

```
Router(config-router)# router-id <IP address>
```

---

Best practice dictates the creation of a loopback address to be used as the router ID for stability and continuity:

---

```
Router(config)# interface loopback 0  
Router(config-if)# ip address <IP address> <subnet mask>
```

---

## Default Cost

Link cost is a 16-bit value (0-65535); default cost is calculated as `100Mbps/interface bandwidth`. (Interfaces 100Mbps and faster are assigned a cost of 1.)

OSPF cost can be manually specified per interface:

---

```
Router(config-if)# ip ospf cost <cost>
```

---

An alternative to defining static costs per interface is to change the numerator bandwidth (default 100Mbps):

---

```
Router(config-router)# ospf auto-cost reference-bandwidth <reference speed>
```

---

Reference speed is a 32-bit value (1 - 4294967). If reference speed is modified, the same modification should be performed on all routers within the area.

## Router Priority

Default DR election priority is 1, and a router with a priority of 0 will not become a DR. Priority range is 0 - 255.

---

```
Router(config-if)# ip ospf priority <priority>
```

---

## Verifying OSPF Configuration

show ip ospf - OSPF process details

show ip ospf database - Contents of the topology database

show ip ospf interface - Interfaces participating in OSPF

show ip ospf neighbor - Neighbor information

show ip protocols - Displays all active routing protocols

show ip route

debug ip ospf events

debug ip packet

## Chapter 6: OSPF Network Topologies

### OSPF Network Topology Options

OSPF assumes a subnet is broadcast-capable by default.

## OSPF Network Types

Broadcast multiaccess

Point-to-point

Point-to-multipoint broadcast

Point-to-multipoint nonbroadcast

Nonbroadcast multiaccess (NBMA)

NBMA and point-to-multipoint are standards-compliant (**RFC 2328**), whereas point-to-multipoint nonbroadcast, broadcast, and point-to-point implementations are Cisco proprietary.

NBMA networks utilize DRs like broadcast networks, however neighbors must be manually defined instead of being automatically discovered.

	<b>NBMA</b>	<b>Point-to-multipoint broadcast</b>	<b>Point-to-multipoint nonbroadcast</b>	<b>Broadcast</b>	<b>Point-to-point</b>
<b>DR/BDR</b>	Yes	No	No	Yes	No
<b>Identify neighbor?</b>	Yes	No	Yes	No	No
<b>Hello/dead timers</b>	30/120	30/120	30/120	10/40	10/40
<b>Standard</b>	RFC	RFC	Cisco	Cisco	Cisco
<b>Network supported</b>	Full mesh	Any	Any	Full mesh	Point-to-point

## Configuring OSPF in a Nonbroadcast Environment

### Nonbroadcast Network

Because NBMA is the default network type for a nonbroadcast interface, the only necessary configuration is to define neighbors.

DR priorities should be specified to ensure only candidates positioned well in the topology are elected DR and BDR.

---

```
Router(config-if)# neighbor <IP address> [priority <priority>] [poll-interval <seconds>] [cost <cost>]
```

---

priority - This can be used to specify a higher priority than what has been configured on the neighbor (but not lower)

`poll interval` - The rate at which hellos are sent to inactive neighbors (default 120 seconds)

`cost` - Cost to reach the neighbor

NBMA configuration:

---

```
Router(config-if)# ip address 10.0.0.60 255.255.255.0
Router(config-if)# encapsulation frame-relay
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.0.0.255 area 0
Router(config-router)# neighbor 10.0.0.40
Router(config-router)# neighbor 10.0.0.50
```

---

## Point-to-multipoint Network

Point-to-multipoint automatically establishes adjacencies along PVCs.

Point-to-multipoint assumes broadcast capability by default; nonbroadcast can be specified, and neighbors must then be defined manually.

---

```
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# encapsulation frame-relay
Router(config-if)# ip ospf network point-to-multipoint [nonbroadcast]
Router(config)# router ip ospf 1
Router(config-router)# network 10.1.1.1 0.0.0.255 area 0
```

---

## Broadcast Network

---

```
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# ip ospf network broadcast
Router(config)# router ip ospf 1
Router(config-router)# network 10.1.1.1 0.0.0.255 area 0
```

---

## Point-to-point on Subinterfaces

---

```
Router(config)# interface serial0
Router(config-if)# no ip address
Router(config-if)# encapsulation frame-relay
```

```
Router(config)# interface serial0.1 point-to-point
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# frame-relay interface-dlci 51
Router(config)# interface serial0.2 point-to-point
Router(config-if)# ip address 10.1.2.1 255.255.255.0
Router(config-if)# frame-relay interface-dlci 52
Router(config)# router ip ospf 1
Router(config-router)# network 10.1.1.1 0.0.0.255 area 0
```

---

## Chapter 7: Using OSPF Across Multiple Areas

### Multi-area OSPF Features

Segmentation of an OSPF domain into areas reduces the utilization of resources on each router.

#### Router Types

Internal - Has all interfaces in one area

Backbone - Has one or more interfaces in the backbone area (area 0)

Area Border (ABR) - Connects two or more areas

Autonomous System Border (ASBR) - Connects to other routing domains; typically located in the backbone

#### Link-State Advertisements

Router link (type 1) - Lists a router's neighbors and its cost to each; flooded throughout the area

Network link (type 2) - Advertisement by the DR containing all routers on the segment it is adjacent to; flooded throughout the area

Network summary link (type 3) - ABRs generate this type of LSA to send between areas; it lists all prefixes available in an area

AS external ASBR summary link (type 4) - Router link LSA for ASBRs

External link (type 5) - Originated by an ASBR, contains a route external to OSPF

NSSA external (type 7) - Equivalent to a type 5 LSA, but generated by an ASBR in a not-so-stubby area (NSSA); converted to a type 5 by the ABR

#### Area Types

Standard area

Stub area - Will not accept external routes (type 5 LSAs); type 5 LSAs are replaced by a

default route

Totally stubby area - Will not accept LSAs of type 3, 4, or 5; routes are replaced by the ABR with a default route; Cisco proprietary

Not-so-stubby area (NSSA) - Stub areas which contain one or more ASBRs; ASBRs in a NSSA generate type 7 LSAs which are then converted to type 5 by the ABR

The greatest advantage of designating stub areas is decreased convergence time.

## Multi-area OSPF Operation

Adjacencies within an area are advertised with type 1 and 2 LSAs, which are passed to the backbone by an ABR as type 3 summaries. These summaries are then injected into other areas from the backbone.

OSPF will choose the path to a destination by the advertisement of the lowest LSA type, then by lowest cost.

### Path Calculation

Cost to another OSPF area = smallest cost to the ABR + smallest cost across the backbone

Cost to external routes can be determined two ways:

E1 = cost to ASBR + advertised external cost; internal routing influences path selection

E2 (default) = only advertised external cost is considered; internal routing is not considered in path selection

### Routing Table Codes

LSA Type	Table Code	Description
1 (Router)	O	Generated by all routers; lists neighbors and costs to them; propagated within an area
2 (Network)	O	Generated by the DR on a multiaccess network; propagated within an area
3 (Inter-area summary)	O IA	Advertises summaries from one area to another
4 (ASBR Summary)	O IA	Advertises the location of an ASBR
5 (External)	O E1 or O E2	Advertises a route external to the AS

### Design Considerations

## Capacity Planning

Main considerations:

- Type of area - normal, backbones, stub, or total stub

- CPU utilization on each member

- Link speed

- Stability

- NBMA - full mesh versus alternate solutions (point-to-point)

- External links - amount of external LSAs, summary to a default route

- Summarization - importance of hierarchical design

Best practice dictates manually defining the DR of a network, and ensuring a router is not the DR for more than one network.

## Summarization

- Inter-area summarization - Performed at the ABR; creates type 3 LSAs. Type 4 LSAs advertise ASBRs.

- External summarization - Performed at the ASBR; creates type 5 LSAs.

## Virtual Links

All OSPF areas must be connected to the backbone area (area 0) through an ABR. Virtual links are established when an area cannot be directly connected to the backbone.

Virtual links are not encouraged and should be used only as a temporary fix.

Virtual links cannot use a stub area for transit.

## Multi-area OSPF Configuration

### network

---

```
Router(config-router)# network <network> <mask> area <area>
```

---

### area range

Used on an ABR to summarize multiple networks into a single type 3 LSA.

---

```
Router(config-router)# area <area> range <network> <mask>
```

---

### **summary-address**

Used on an ASBR to define summarization boundaries for external routes redistributed into OSPF.

---

```
Router(config-router)# summary-address <address> <mask> [not-advertise] [tag <tag>]
```

---

### **area stub**

Areas are designated as stubs when the chosen exit ABR is unimportant (or there is only one).

---

```
Router(config-router)# area <area> stub
```

---

All routers within a stub area must be configured as such. A stub router will not form an adjacency with a non-stub router in the same area.

### **area stub no-summary**

no-summary further limits a stub area by creating a totally stubby area. Totally stubby areas do not receive type 3 or 5 LSAs from other areas.

---

```
Router(config-router)# area <area> stub no-summary
```

---

Only ABRs need to be configured with no-summary appended to the stub command.

To direct packets outside the stub area, routers rely on a default route advertised by the ABR(s).

The concept of totally stubby areas is Cisco proprietary.

### **area default-cost**

Used to define the cost of the default route injected by an ABR into a stub area. The default is (cost to the ABR + 1).

This can be used to prefer one stub exit over others.

---

```
Router(config-router)# area <area> default-cost <cost>
```

---

Configuration is done only on the ABR.

## **area virtual-link**

Virtual links are created when an outlying area does not have a direct connection to the backbone (area 0).

---

```
Router(config-router)#area <area> virtual-link <router ID>
```

---

Router ID specifies the remote endpoint of the virtual link.

## **Verifying OSPF Configuration**

```
show ip route
show ip ospf border-routers
show ip ospf virtual-links
show ip ospf database
```

## **Troubleshooting**

Adjacency changes can be logged; this is better suited than debugging for long-term monitoring.

---

```
Router(config-router)# log-adjacency-changes
```

---

Debugging:

```
debug ip packet
debug ip ospf events
```

When troubleshooting a failed adjacency, verify:

- Subnet
- MTU
- Interface HELLO timer
- OSPF HELLO interval
- OSPF dead interval
- Area ID
- Area type

## Chapter 8: OSPF Advanced Topics

### Stub Areas Overview

#### Stub area

ABR replaces all external routes with a default route

Must be configured on all routers in the area

Defining a stub area:

---

```
Router(config-router)# area <area> stub
```

---

#### Totally stubby area

ABR replaces all inter-area and external routes with a default route

Cisco proprietary

Configured on ABRs; internal routers configured as normal stubs

Defining a totally stub area on the ABR:

---

```
Router(config)# area <area> stub no-summary
```

---

#### Not-so-stubby area

Is a stub or totally stubby area with an ASBR

Advertises external routes with type 7 LSAs instead of the normal type 5

NSSA routes appear in the routing table as type N1 or N2 (instead of E1 or E2)

All routers within the area must be configured

Defining a not-so-stubby area:

---

```
Router(config)# area <area> nssa [no-summary]
```

---

### OSPF Authentication

Authentication types:

null (no authentication)  
plaintext  
MD5

Authentication is enabled per interface.

### **Plaintext**

---

```
Router(config-if)# ip ospf authentication-key <password>  
Router(config-if)# ip ospf authentication
```

---

### **MD5**

---

```
Router(config-if)# ip ospf message-digest-key <key number> md5 <password>  
Router(config-if)# ip ospf authentication message-digest
```

---

## **Chapter 9: Fundamentals of Integrated IS-IS**

### **Background**

IS-IS was developed by DEC as an international standard for the OSI to compete with TCP/IP in the 1980s.

*Integrated IS-IS* refers to IS-IS which supports IP.

IS - Intermediate System (router)

ES - End System (host)

Connectionless Network Protocol (CLNP) is the network layer protocol defined in OSI. It is used by the Connectionless Network Service (CLNS).

IS-IS routers use a CLNS address for a router ID.

OSI routing levels:

Level 0 - Used to locate end systems

Level 1 - Exchange of routes within an area (IS-IS)

Level 2 - Backbone between areas (IS-IS)

Level 3 - Between autonomous systems (Interdomain Routing Protocol (IDRP))

Like OSPF, IS-IS also relies on the Dijkstra algorithm for path selection.

## Packet Types

IS-IS packets use an 8-byte header with variable *Type Length Value (TLV)* fields for data. TLVs make IS-IS very flexible and easily extended to support other protocols.

### Hellos

Hello types:

End System Hello (ESH) - Used by ISO hosts to attach to routers; not used with IP networks

Intermediate System Hello (ISH) - Used by routers to announce themselves to end systems

Intermediate-to-Intermediate Hello (IIH) - Router to router; used separately at levels 1 and 2

Point-to-point hello packets are used over point-to-point links, and level 1 or 2 LAN hello packets are used on multiaccess links.

### Link-State Packet (LSP)

Level 1 LSPs list a router's adjacencies.

Level 2 LSPs list a router's adjacencies and the areas it can reach.

### Sequence Number Packet (SNP)

An advertisement containing one or more condensed LSPs.

SNPs are never flooded, only sent between neighbors.

Can be complete (CSNP) or partial (PSNP).

## Comparing IS-IS and OSPF

IS-IS is most similar to OSPF using only totally stubby areas.

Notable differences:

OSPF has a single backbone area; IS-IS has a backbone on top of its areas

OSPF has a DR and BDR; IS-IS has only one Designated Intermediate System (DIS)

OSPF has multiple LSA types; IS-IS uses a standard advertisement form (composed of TLV fields)

OSPF rides on IP; IS-IS is encapsulated directly into the layer 2 header

OSPF was designed for IP; IS-IS was intended to support multiple network layer protocols, namely CLNS and IP

IS-IS allows for the preemption of a DIS by a router with a higher priority (MAC address determines the DIS in the event of a priority tie).

All routers on the medium form adjacencies with one another, not just with the DIS.

IS-IS was better positioned to adopt IPv6 as it merely required new TLV types, whereas OSPF had to be rewritten (OSPFv3).

## Addressing

ISO address types:

- Network Service Access Point (NSAP)

- Network Entity Title (NET)

Addresses are 8-20 bytes long, composed of three parts:

- Area - Similar to an IP subnet

- ID - Identifies a particular host

- SEL - Identifies a process on the host (similar to TCP/UDP ports)

Address fields:

- Inter Domain Part (IDP) - Used for external routing

  - Authority and Format Identifier (AFI) - Identifies the authority that dictates the format of the address (usually "country code", "international code", or "private")

  - Initial Domain Identifier (IDI) - An organization in the AFI (optional)

- Domain Specific Part - Used for routing within the AS

  - High Order DSP (HODSP) - The area within the AS

  - System ID - Identifies the system, 6-8 bytes (Cisco supports only 6-byte length), could be MAC or static length IP (192.168.0.1 ## 1921.6800.0001)

  - NSEL - 1-byte, identifies a network layer service

An NSAP with the NSEL set to 0x00 becomes a NET.

## Addressing Rules

ISO addresses are assigned to the system, not a particular interface

One address per router is typical; limit is 3

If a router has multiple NETs, they must all have the same system ID

The area address must be identical for all routers in an area

All routers must have a unique system ID of the same length (6 bytes for Cisco devices) within their area

### **Example NET**

49.0005.AA00.0301.16CD.00

AFI (private) HODSP (area) System ID (MAC) SEL (null)

49            0005            AA00.0301.16CD 00

### **Adjacencies**

Requirements for an adjacency to form:

MTUs must match

Levels must match

If level 1, routers must be in the same area

System IDs must be unique

Authentication type (if any) must match

IS-IS has only two network types: broadcast and point-to-point.

It is recommended to configure NBMA networks as point-to-point with subinterfaces.

### **IS-IS Operation**

Routing process:

Update

Decision

Forwarding

Receiving

### **Update Process**

Triggers:

- An adjacency comes up or goes down
- An interface changes state or receives a new metric
- A route changes (for example, due to redistribution)

An LSP generated by an update is flooded throughout an area on its respective level.

Three fields in an LSP are inspected to determine whether its information is more recent than what exists in the recipient's database:

- Remaining Lifetime - Time to live in the database; 20 minutes by default, refresh timer is 15 minutes
- Sequence Number - 32-bit sequential counter
- Checksum - Error checking

## **Decision Process**

IS-IS will consider six routes to a destination (or more on newer IOSs).

Internal paths are preferred over external paths.

Level 1 paths are preferred over level 2 paths.

If no path exists, the packet is sent to the nearest level 2 router.

Integrated (IP) IS-IS metrics:

- Default - Required; Cisco default for all interfaces is 10
- Delay - Not supported
- Expense - Not supported
- Error - Not supported

By default, interfaces are given a 6-bit metric, while a 10-bit (*narrow*) metric defines the total path cost. Cisco's implementation increases the metric size to 24-bits (*wide*).

## **Design Considerations**

### **Area Design**

Typical designs:

Level 1 only - Sufficient for small networks but does not scale well

Level 2 only - Provides for expansion through the addition of level 1 areas

Level 1-2 only - Cisco default, allows for easy migration to a hierarchical design, hard on resources (all routers have two databases)

Hierarchy - Intended implementation of IS-IS but can produce suboptimal routing

## Route Summarization

Summarization reduces the need to flood LSPs.

Level 1-2 routers can summarize routes within their area. Summarization must be configured identically for all level 1-2 routers in an area.

Level 1 routes cannot be summarized within an area.

## Chapter 10: Configuring Integrated IS-IS

### Basic Configuration

Step 1: Enable the routing process with `router isis`

Step 2: Configure the NET address (specifies router ID and area) with `net <NET>`

Step 3: Enable IS-IS per interface with `ip router isis`

```
Router(config)# router isis Router(config-router)# net 49.0001.0000.0000.0001.00
Router(config-if)# ip router isis
```

### Optional Configuration

#### Changing the Router from Level 1-2

Cisco routers become level 1-2 routers by default.

A router can be configured to operate on only one level:

---

```
Router(config-router)# is-type {level-1 | level-2}
```

---

Routing level can alternatively be configured per interface:

---

```
Router(config-if)# isis circuit-type {level-1 | level-2-only}
```

---

## Configuring Summarization

Summarization is configured on level 1-2 routers.

All level 1-2 routers in an area must summarize equivalently to avoid suboptimal routing.

Routes cannot be summarized within an area.

---

```
Router(config-router)# summary-address <network> <subnet>
```

---

## Configuring NBMA

IS-IS only recognizes networks as either broadcast or point-to-point; if the interface is not a serial line to exactly one other endpoint, IS-IS assumes broadcast capability.

Point-to-point subinterfaces are recommended for IS-IS over NBMA.

---

```
Router(config)# interface s0
Router(config-if)# no ip address
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface s0.1 point-to-point
Router(config-subif)# ip address 10.10.10.1 255.255.255.252
Router(config-subif)# ip router isis
Router(config-subif)# frame-relay dlci 629
```

---

## Verifying and Troubleshooting IS-IS Operation

```
show clns neighbor
show clns interface
show isis database [detail]
show isis spf-log
debug isis [adjacencies-packets | spf-statistics | update-packets]
```

## Chapter 11: Implementing Redistribution and Controlling Routing Updates

### Redistribution Fundamentals

*Internal routes* are native to a routing process. *External routes* are redistributed into a routing process.

IGRP and EIGRP will automatically redistribute into one another if the autonomous system numbers (ASNs) match.

## Routing Decisions That Affect Redistribution

### Routing Metrics

A *seed metric* must be defined when redistributing routes between routing protocols with unlike metrics.

Some default seed metrics need to be changed to allow redistribution to take affect.

Default seed metrics:

EIGRP: Infinity (no routes enter the table)

IS-IS: 0

OSPF: 20 (type 2); BGP-learned routes are given 1 (type 2)

BGP: MED is given the IGP metric value

### Path Selection

If a path is known by multiple routing protocols, the path belonging to the routing protocol with the lowest *administrative distance* is chosen.

### Potential Problems

Routing loops resulting from routing info being redistributed back into the routing protocol it was learned from (*route feedback*)

Suboptimal routing decisions based on incorrect metrics

Increased convergence time

Preventing route feedback:

Change the metric

Change the administrative distance

Use default routes

Use passive interfaces with static routes

Use distribute lists

Avoid overlapping routing protocols on a router; separate routing protocols into easily distinguished

domains.

Avoid two-way distribution.

A network will converge at the speed of the slowest routing protocol.

## **Controlling Routing Updates During Redistribution**

### **Passive Interfaces**

Passive interfaces do not actively participate in routing protocols, and can be implemented to avoid routing loops.

### **Static Routes**

Practical on stub networks.

Can be used to redistribute only a small number of routes rather than an entire database.

Can be used to alter the mask of a network, or to redistribute from classless into classful routing.

### **Default Routes**

Reduces complexity and can prevent routing loops.

### **Null Interface**

All traffic routed to a null interface is dropped.

Useful for catching traffic destined for unknown subnets within a summarized link.

### **Distribute Lists**

Access lists applied to the routing process, determining which networks are accepted into the routing table or advertised.

### **Route Maps**

More advanced than distribute lists; more granular and able to manipulate information.

Uses match to configure parameters, and set to manipulate data.

## **Configuring Redistribution**

Enable redistribution:

---

```
Router(config-router)# redistribute <protocol> [process ID] [metric <metric>]
[route-map <route map>] [...]
```

---

## Configuring the Default Metric

The default metric can be specified through either `redistribute` (above) or with `default-metric`.

`redistribute` takes precedence in the event both are configured.

`default-metric` is preferable when multiple protocols are being redistributed.

---

```
Router(config)# router eigrp 100
Router(config-router)# redistribute rip metric 10000 100 255 1 1500

Router(config)# router rip
Router(config-router)# redistribute static
Router(config-router)# redistribute ospf 25
Router(config-router)# default-metric 2
```

---

## Configuring the Administrative Distance

---

```
Router(config-router)# distance <weight> [<address> <mask>] [ACL]
```

---

## Controlling Routing Updates

Interfaces on which adjacencies should not be formed should be configured as passive.

---

```
Router(config-router)# passive-interface f0/1
```

---

All interfaces can be defined as passive by default with `passive-interface default`.

Static/default routes can be used in the absence of a routing protocol:

---

```
Router(config)# ip route <network> <mask> <address | interface>
[administrative distance]
```

---

Distribute lists can be applied to allow only certain routes to be received or advertised:

---

```
Router(config-router)# distribute-list [ACL number | ACL name] {in | out}
[interface | process | ASN]
```

---

## Verification and Troubleshooting

```
show ip protocol
show ip route [routing protocol]
show ip eigrp neighbors
show ip ospf database
traceroute and extended ping
```

## Chapter 12: Understanding Route Maps

### Characteristics

*Route maps* are an advanced alternative to simple ACLs, and are used for:

- Controlling route redistribution - Can permit or deny routes, or change metrics
- Policy-based routing - Allows for more complex routing decisions
- Additional granularity in NAT - Can define arbitrary address mappings opposed to the normal static mappings
- Defining network policy - Used by BGP

Route maps are structured and processed similar to ACLs.

Each route map statement has a sequence number and is configured to either permit or deny traffic.

If a match condition is met, zero or more set parameters are acted upon.

Like ACLs, route maps end with an implicit deny any.

Route map structure:

```
route-map <name> <permit | deny> <sequence number>
    {match | set} <condition>
    ...
...

```

## Configuration

---

```
Router(config)# route-map <name> [{permit | deny} [<sequence number>]]  
Router(config-route-map)# {match | set } <condition>
```

---

match conditions used in redistribution:

```
match interface  
match ip address [ACL]  
match ip next-hop  
match ip route-source  
match metric  
match route-type  
match tag
```

set operations used in redistribution:

```
set level {level-1 | level-2 | level-1-2 | stub-area | backbone} (OSPF/IS-IS)  
set metric  
set metric-type {internal | external | type-1 | type-2}  
set tag <value>
```

Implementing route maps for redistribution:

---

```
Router(config-router)# redistribute <protocol> [process ID] route-map <name>
```

---

## Monitoring

```
show ip protocol  
show ip route  
show route-map
```

## Chapter 13: Dynamic Host Configuration Protocol

### DHCP Operation

DHCP is the successor to BOOTP.

Five steps in a DHCP assignment:

Client broadcasts a DISCOVER message

Each DHCP server which receives the DISCOVER replies with an OFFER

Client picks an OFFER, responds to the issuing server with a REQUEST

Server responds with an ACKNOWLEDGEMENT

Client broadcasts gratuitous ARPs to ensure the address isn't already in use

## DHCP Configuration

### DHCP server

---

```
Router(config)# ip dhcp pool 1
Router(config-router)# network 192.168.0.0 /24
Router(config-router)# default-router 192.168.0.1
Router(config-router)# lease 12
Router(config-router)# dns-server 192.168.1.100
```

---

import all will replicate DHCP parameters from one pool to others.

### DHCP Relay

---

```
Router(config-if)# ip helper-address <DHCP server>
```

---

By default helpers forward UDP broadcasts on these ports:

37 (NTP)

49 (TACACS)

53 (DNS)

67 and 68 (DHCP)

69 (TFTP)

137 and 138 (NetBIOS)

### DHCP Client

---

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

---

## Verification

```
show ip dhcp bindings
show ip dhcp database
```

## Chapter 14: BGP Concepts

### Introduction to BGP

BGP is the only routing protocol in widespread use which facilitates interdomain routing (between autonomous systems).

BGP is *path-vector*; routes are tracked in terms of which autonomous systems they pass through.

BGP attributes allow granularity in path selection.

### Message types

- Open
- Keepalive
- Update
- Notification

Connections between BGP peers are maintained via *Open* and *Keepalive* messages on TCP/179.

### BGP tables

- Neighbor table
- BGP table
- IP Routing table

BGP doesn't use a route metric; it relies on a complex path selection process.

*eBGP* is used between autonomous systems; *iBGP* is used within an AS.

### Connecting to the Internet With BGP

Route reception options:

- Default route from provider(s) - Easy on resources, internal traffic routed to nearest BGP router

- Some routes + default route - Allows for selection of some paths with others falling back to a

default route

All routes (*full table*) - Hard on resources, but guarantees the most direct path is taken

## Synchronization

The *synchronization requirement* mandates that BGP must learn a route from an IGP for it to be propagated, to ensure consistency.

BGP synchronization is often disabled for autonomous systems which do not act as a transit AS.

## BGP States

Idle (active) - Searching for neighbors

Connect (active) - TCP connection established

Open Sent (active) - *Open* message sent

Open Confirm (active) - Response received

Established - BGP neighborhood established

BGP neighborhoods can be confirmed with `show ip bgp neighbors`.

Neighbors still displayed as "active" after some time has passed have not correctly peered.

## Chapter 15: BGP Neighbors

### BGP Configuration

#### Enabling BGP

---

```
Router(config)# router bgp <ASN>
```

---

Neighbors must be explicitly defined along with their AS number:

---

```
Router(config-router)# neighbor <IP address> remote-as <ASN>
```

---

iBGP connections are formed between same ASNs; eBGP connections are between different ASNs.

#### Peer groups

*Peer groups* are defined to efficiently apply policies to multiple neighbors:

---

```
Router(config-router)# neighbor <group name> peer-group
Router(config-router)# neighbor <group name> remote-as <ASN>
Router(config-router)# neighbor <IP address> peer-group <group name>
Router(config-router)# neighbor <IP address> peer-group <group name>
```

---

Neighbors can be temporarily disabled with `neighbor {<IP address> | <group name>} shutdown`.

## Source interfaces

The source interface from which to peer with a neighbor can be administratively set (ideally to a loopback):

---

```
Router(config-router)# neighbor 10.1.1.2 remote-as 100
Router(config-router)# neighbor 10.1.1.2 update-source loopback0
```

---

eBGP packets by default have a TTL of 1, requiring neighbors to be directly attached. This can be administratively overridden with `neighbor <IP address> ebgp-multihop <hop count>`.

## Forcing the next-hop address

The `next-hop-self` command allows a router to substitute its internal address as the next hop for a route to an external AS to ensure that its internal neighbors can reach it.

---

```
Router(config-router)# neighbor 10.1.1.2 next-hop-self
```

---

## Defining the networks to be advertised

The `network` command in BGP is used to define which networks to advertise (*not* which interfaces should run BGP).

---

```
Router(config-router)# network <network address> mask <subnet mask>
```

---

## Route summarization

BGP routes are summarized using an administratively defined *aggregate route*:

---

```
Router(config-router)# aggregate-address <network address> <subnet mask>
[summary-only] [as-set]
```

---

If `summary-only` is set, no more-specific routes will be advertised, just the summary. This is typical.

If `as-set` is used, all autonomous systems which the route traverses will be recorded in update messages.

## Authentication

A password can be applied to a neighbor statement to force MD5 authentication. This is very common between peers on the Internet.

---

```
Router(config-router)# neighbor <IP address> password <password>
```

---

## Verifying Operation

```
show ip bgp [summary]
show ip bgp neighbors
show processes cpu
debug ip bgp [dampening | events | keepalives | updates]
```

## Resetting neighbors

Configuration changes can necessitate a *hard reset* of neighbors:

---

```
Router(config-router)# clear ip bgp {* | <address>} [soft [in | out]]
```

---

Drawbacks of a hard reset include:

- The time taken to re-exchange routes and the interruption in the routing process

- Hard resets count as a link flap

- Re-exchange of routes could generate a large amount of traffic

*Soft resets* achieve the same goal without counting as a link flap, and can be applied inbound or outbound.

# Chapter 16: Controlling BGP Route Selection

## BGP Attributes

- Aggregator** - ID and AS of the advertising router

- AS\_Path** - List of the autonomous systems the route has traversed

**Atomic aggregate** - Summary includes multiple AS

**Cluster ID** - Originating cluster

**Community** - Route tag

**Local preference** - External path metric for internal neighbors (prefer highest)

**Multiple Exit Discriminator (MED)** - Informs external peers which path to take into the AS (prefer lowest)

**Next Hop** - External peer in neighboring AS

**Origin** - Lowest origin type preferred (IGP < EGP < unknown)

**Originator ID** - Identifies route reflector

**Weight** - Administrative Cisco attribute (prefer highest)

## Path Selection Process

Routes must meet the *synchronization requirement* (if enforced) to be considered.

1. **Weight** (highest) - Provided for administrative override
2. **Local Preference** (highest) - Used internally to select path out of AS
3. **Self-originated** (TRUE) - Prefer paths originated by self
4. **AS\_Path** (shortest) - Minimize AS hops
5. **Origin** (lowest type) - native < EGP < redistributed; stability
6. **MED** (lowest) - Used to determine entrance into an AS
7. **External** (eBGP < iBGP) - External path preferred over internal path
8. **IGP cost** (lowest) - Used to consider additional information about the route
9. **eBGP peering** (oldest) - Stability
0. **Router ID** (lowest) - Lowest router ID breaks a tie

## Controlling Route Selection

### Weight Attribute

The *weight* attribute is Cisco proprietary, and is considered before any other attribute.

Weight is local to the router and not propagated to other routers.

Weight is a 16-bit value; higher is preferable. Default is 0 if the route is learned from a peer, or 32,768 if sourced locally.

---

```
Router(config-router)# neighbor {<IP address> | <group name>} weight <weight>
```

---

## Local-Preference Attribute

Local preference is a 32-bit value; higher values are preferred. Default value is 100.

Configured as a default:

---

```
Router(config-router)# bgp default local-preference <value>
```

---

Configured per prefix (via a route-map):

---

```
Router(config-router)# neighbor {<IP address> | <group name>} route-map <map name> in
```

---

## MED Attribute

The *multi-exit discriminator* is used to influence path selection by external neighbors routing into the AS.

Default MED value is 0; lower is preferred.

Configured as a default:

---

```
Router(config-router)# default-metric <value>
```

---

MED can also be configured per prefix via route-maps.

## Verifying Attribute Configuration

show ip bgp displays:

Destination network

Next hop

Metric

Local Preference

Weight

AS Path

## Chapter 17: What is Multicasting?

IP packet types:

Unicast (one -> one)

Broadcast (one -> all)

Multicast (one -> many)

By default, routers do not forward multicasts, but switches flood multicasts out all ports.

### MAC Addressing

The 8th bit in a MAC address is a flag; if 1, the MAC is a multicast address.

Multicast MACs will have an OUI of 01-00-5E, and the 25th bit (the first bit in the second half) will be 0.

The remaining 23 bits in the MAC are taken from the rightmost 23 bits in the IP address.

Because only 23 bits of the IP address are ported to the MAC address, there is some potential for overlap, so a host must examine all frames it receives with the MAC and further verify the IP address within each.

### IP Addressing

Class D IP addresses are used for multicast destinations. (The first four bits will be 1110.)

Reserved multicast address ranges:

#### **Link-local (224.0.0.0/24)**

Packets are transmitted with a TTL of 1, ensuring traffic never leaves the local segment.

Some well-known examples are:

224.0.0.1 - All hosts

224.0.0.5 - All OSPF routers

224.0.0.6 - All OSPF DRs

224.0.0.9 - All RIPv2 routers

224.0.0.10 - All EIGRP routers

#### **Source-specific (232.0.0.0/8)**

Hosts are configured to only receive traffic from a specific server.

## **GLOP (233.0.0.0/8)**

Allocates 256 multicast IP addresses to each registered AS.

The ASN is used to fill the middle two octets. For example, AS1000 receives 233.3.232.0/24 ((3 \* 256) + 232).

## **Administratively scoped (239.0.0.0/8)**

Used within private multicast domains; similar to the private RFC 1918 address ranges.

Within this range, 239.252.0.0/14 is reserved for site-local multicast, and the rest of 239.192.0.0/10 is reserved for org-local scope.

## **Globally scoped (224.0.1.0-231.255.255.255 and 234.0.0.0-238.255.255.255)**

This is the bulk of the multicast range.

Global addresses may be routed across the Internet and are generally only temporarily assigned.

## **Chapter 18: IGMP**

### **Understanding IGMP**

Three methods exist to optimize multicast switched traffic:

- Static MAC table entries
- Cisco Group Management Protocol (CGMP)
- IGMP snooping

*Internet Group Management Protocol (IGMP)* is used by clients to identify themselves to a router and request multicast service.

### **IGMPv1**

Defined in **RFC 1112**.

A client sends a *Membership Report* message to its local router, requesting to be added to a multicast group (address).

Every 60 seconds the querier router on a segment multicasts a query to 224.0.0.1 (all hosts on the segment) to check if any host on the segment still wants to receive multicast traffic for that group. Interested hosts respond with a membership report.

IGMPv1 does not provide a mechanism for hosts to explicitly leave a group; requests expire after three query intervals pass without an answer.

## **IGMPv2**

Defined in **RFC 2236**.

Improvements over IGMPv1:

- Queries from the router can be sent to the original *all hosts* address (224.0.0.1), or to members of a specific group.

- Hosts can dynamically leave a group.

- Querier election

- Query-interval response time

A host no longer wanting to receive traffic for a multicast group sends a *Leave Group* multicast to 224.0.0.2 (all routers). The querier router responds with a group-specific query to check if any other hosts on the segment still want to receive multicast traffic for that group.

IGMPv1 routers cannot understand IGMPv2.

IGMPv2 routers are queriers by default; they transition to non-queriers upon hearing a query from a router with a higher IP address.

IGMPv2 queries include a query-interval response time, which tells members how long they have to respond to a query.

## **IGMPv3**

Defined in **RFC 3376**.

IGMPv3 introduces support for multicast source filtering. Along with a multicast group, a host can specify a list of sources from which it will accept multicast traffic.

## **Configuring IGMP**

`show ip igmp interface` is used to display all interfaces forwarding multicast traffic to clients.

`show ip igmp group` will display all active groups.

## **IGMP Snooping**

IGMP snooping can be enabled on switches to identify end systems which request multicast group membership and limit that traffic to only the necessary ports.

IGMP snooping is enabled by default, or with `ip igmp snooping`.

`show ip igmp snooping` will display IGMP snooping settings.

`show multicast router` displays associated router ports.

`show multicast group` displays active multicast groups and their ports.

## Chapter 19: Configuring Multicast

### Routing Multicast Traffic

Unicast IP traffic is always destined for a single interface. Special considerations must be taken for multicast traffic, which can have many destinations.

#### Reverse Path Forwarding

Routers perform an RPF test on every multicast packet they receive.

When a multicast packet is received, the source IP address and interface are checked to ensure the packet arrived on the same interface that would be used to send traffic to the source address.

This is the *reverse* of normal packet forwarding, in which the *destination* address is looked up.

#### Multicast Trees

Multicast traffic must be routed in a loop-free tree, away from the source. These are *distribution trees*.

Multicast routes are display in the form  $(S, G)$ , displaying the source and group IP addresses.

Two types of distribution trees:

- Shared - A common set of links over which all multicast traffic flows; administratively pre-defined

- Source-rooted - A separate route exists for each source

Shared trees can be extended with source-rooted trees at rendezvous points (RPs).

#### Multicast Routing Protocols

Some multicast routing protocols are:

- Multicast OSPF (MOSPF)

- Distance Vector Multicast Routing Protocol (DVMRP)

Center-based trees

Core-based trees

Protocol Independent Multicast (PIM)

Cisco IOS only supports PIM.

*Dense-mode* multicast routing protocols assume all hosts are interested in receiving multicast traffic, and prune out only hosts which explicitly ask not to receive it.

*Sparse-mode* protocols work the opposite way, only forwarding multicast traffic to hosts who explicitly request it.

## **Protocol Independent Multicast (PIM)**

PIM operates based on a router's IP routing table.

PIM can operate in one of three modes:

Dense mode

Sparse mode

Sparse-dense mode

Two versions of PIM are available (v1 and v2); PIMv2 is default.

### **PIM Dense Mode (PIM-DM)**

PIM routers become neighbors by exchanging hello messages.

Initially, multicast traffic is flooded to all PIM-DM routers, allowing the tree to be built.

If no hosts have registered with a router via IGMP to receive the traffic, that router can be pruned from the tree.

After the tree has been built, new hosts can join a group, and branches are rebuilt as necessary.

### **PIM Sparse Mode (PIM-SM)**

A PIM-SM tree is built from the hosts to the root.

The tree's root is a PIM-SM router centrally located on the network, called a rendezvous point (RP).

This is considered a *shared tree* because the multicast source must also join the tree.

Once the source has been learned, PIM-SM routers switch to a shortest-path tree rooted at the multicast source.

## **PIM Sparse-Dense Mode**

Sparse-dense mode allows a router to operate in either sparse or dense mode per group.

If a group has an RP defined, the router operates in sparse mode; otherwise, dense mode is used.

### **PIMv1**

The range of supported multicast groups can be limited with an ACL.

RPs can be configured manually, or dynamically via the *auto-RP* process.

If defined manually, the address of the RP must be configured on all routers, including the RP itself.

#### **Auto-RP**

*Auto-RP* is Cisco proprietary and provides a method of informing PIM-SM routers of the RP for a group.

A central router is identified as the *mapping agent*. The mapping agent learns of all candidate RPs by listening for multicasts to the *Cisco-RP-Announce* group (224.0.1.39).

A router must be explicitly defined as a candidate RP before it will announce itself.

### **PIMv2**

PIMv2 provides a standards-compliant equivalent to Auto-RP, called the *bootstrap router method*.

In this method, a *bootstrap router (BSR)* is identified. The BSR then advertises RP information to each group.

## **Enabling PIM Sparse-Dense Mode**

1. Enable multicast routing
2. Turn on PIM in the appropriate mode on selected interfaces
3. Set up rendezvous points (RPs)

### **Enable Multicast Routing**

---

```
Router(config)# ip multicast-routing
```

---

## Turning on PIM

PIM is enabled at the interface level.

Enabling PIM on an interface automatically enables IGMP.

---

```
Router(config-if)# ip pim {dense-mode | sparse-mode | sparse-dense-mode}
Router(config-if)# ip pim version {1 | 2}
```

---

Changing the PIM version is optional.

## Configuring RPs

### Manual configuration

---

```
Router(config)# ip pim rp-address <IP address> [<ACL>] [override]
```

---

### Auto-RP (PIMv1)

Configure a router to act as the mapping agent:

---

```
Router(config)# ip pim send-rp-discovery <interface> scope <TTL>
```

---

The TTL specified determines the scope of the RP (in hop counts).

Configure candidate RPs:

---

```
Router(config)# ip pim send-rp-announce <interface> scope <TTL> [group-list <ACL>]
[interval <seconds>]
```

---

### Bootstrap Router Method (PIMv2)

Define a BSR:

---

```
Router(config)# ip pim bsr-candidate <interface> <hash mask length> [<priority>]
```

---

Configure candidate RPs:

---

```
Router(config)# ip pim rp-candidate <interface> <tll> [group-list <ACL>]
```

---

By default, bootstrap messages flood throughout the PIM domain. Border routers can be defined to stop forwarding the messages:

---

```
Router(config)# ip pim border
```

---

## Verifying Operation

`show ip mroute` displays the multicast routing table.

`show ip pim interface` lists all interfaces participating in PIM.

`show ip pim neighbors` lists all PIM neighbors.

`show ip pim rp` displays the RP for each multicast group.

`show ip rpf <address>` inspects reverse-path forwarding information for an address.

## Chapter 20: Introduction to IPv6 and IPv6 Addressing

### The IPv6 Header

**Version** (4 bits) - 6

**Traffic class** (8 bits) - Priority used for QoS

**Flow label** (20 bits) - Allows arbitrary tagging of individual traffic flows

**Payload length** (16 bits) - Length of data in packet

**Next header** (8 bits) - Indicates the type of the next header

**Hop limit** (8 bits) - Functions like IPv4's TTL

**Source address** (128 bits)

**Destination address** (128 bits)

**Extension headers** (optional) - Zero or more extension headers follow the IPv6 header (specified by the *next header* field), such as:

**Hop-by-hop options** - Options communicated to intermediate routers along the path

**Destination options** - Options intended for the end node

**Routing** - Used to define a predetermined routing path

**Fragment** - Replaces the fragmentation fields of IPv4

**Authentication and ESP** - Authentication Header (AH) and Encapsulating Security Payload (ESP) header used by IPSec

## Checksum

IPv6 does not include a checksum, relying instead on upper-layer protocols to handle error checking.

This is more efficient as routers no longer have to recalculate a packet's checksum after modifying it (for example, by decrementing the TTL).

## Fragmentation

Fragmentation information has been moved to an optional extension header.

IPv6 routers do not fragment packets; any necessary fragmentation must be detected and performed by the source node.

Because an IPv6 device is able to determine the path MTU (via ICMP), fragmentation should only occur when upper-layer protocols don't understand communications from the IPv6 stack.

## Flow Label

Flow labeling allows similar traffic to be tagged upon entering a network, and quickly switched to the destination.

## IPv6 Addressing

Address types:

**Unicast** - Packet destined for a single interface; can be global or link-local

**Multicast** - Packet destined for all interfaces in a multicast group

**Anycast** - Packet destined for the *nearest* interface in a group

An IPv6 interface may have many addresses.

Addresses are in the format 2001:0:1:2::ABCD (shorthand notation)

::1/128 is the IPv6 loopback address.

## Interface Identifiers

Interface IDs make up the low-order 64 bits of an IPv6 address and are based on the data-link layer

address.

IPv6 addresses for Ethernet interfaces create the interface ID from the MAC address by converting it to EUI-64 (Extended Universal Identifier 64-bit):

First three bytes of MAC (OUI)

Seventh bit (Universal/Local bit) of OUI set to 1

'FFFE'

Last three bytes of MAC

The U/L bit determines whether the address is considered globally or locally unique; MAC addresses are assumed to be globally unique, thus the bit is set to 1 (universal).

## Unicast Addressing

### Global Aggregatable

Global unicast structure:

First 48 bits: Global prefix

Next 16 bits: Subnet ID

Remaining 64 bits: Interface ID

Currently IANA is only assigning addresses from 2000::/3 (one eighth of the available space).

Registries receive a range (typically a /12) within 2000::/3. (**Current assignments**)

### Link-local

Link-local addresses are used within a local network and are autoconfigured with a FE80::/10 prefix and an EUI-64 format interface ID.

1111 1110 10 (FE80/10)

54 0's

EUI-64

## Anycast Addressing

An IPv6 anycast address is a global unicast address assigned to multiple devices.

Routers forward packets destined for an anycast address to the closest interface (determined by the

routing protocol's metric).

There is no reserved address space for anycast addresses; a global address becomes an anycast address when it is applied to multiple nodes.

Nodes must be appropriately configured to recognize that they have been assigned an anycast address.

All IPv6 routers must support the *subnet-router anycast address* (the subnet address with a zeroed interface ID) for the subnets connected to it.

## Multicast Addressing

Every IPv6 interface should recognize several default multicast addresses, such as *all-nodes*.

Address format:

FF

Flags (4 bits)

Scope (4 bits)

Remainder: Group ID (112 bits)

Multicast addresses are prefixed with FF00::/8.

Flags:

1. Undefined; 0
2. R-bit; 1 if the RP address is embedded in the multicast address
3. P-bit; 1 if the multicast address is assigned based on the unicast prefix
4. T-bit; 1 if the address is permanently assigned

The scope determines how far the multicast traffic may travel.

1 = Interface-local (loopback)

2 = Link-local

4 = Admin-local

5 = Site-local

8 = Organization-local

E = Global

All nodes automatically belong to the *all-nodes* group (FF01::1 for interface-local, FF02::1 for

link-local).

*Solicited-node* multicasts serve as a replacement for ARP; FF02::1:FF00:/104 + last 24 bits of the unicast address.

Routers belong to the *all-routers* group (FF01::2 for interface-local, FF02::2 for link-local, FF05::2 for site-local).

## Address Assignment

Addresses may be manually assigned, or assigned dynamically via DHCPv6 or stateless autoconfiguration.

### Stateless Autoconfiguration

End systems receive network information advertised by a local router and automatically append a generated EUI-64 to the 64-bit network prefix.

*Duplicate Address Detection (DAD)* detects and avoids duplicate addresses.

### DHCPv6 and Stateless DHCPv6

DHCP may still be used in the interest of security and/or privacy. (EUI-64's taken from a MAC expose a potentially sensitive address.)

Stateless DHCPv6 is defined in [RFC 3736](#).

### IPv6 Mobility

Defined in [RFC 3775](#).

An IPv6 mobile node has a *home address* on its home network and a *care-of address* on its current network.

When roaming, an IPv6 node sends a *binding update* to a router on its home network.

## Chapter 21: IPv6 Routing Protocols, Configuration, and Transitioning from IPv4

### IPv6 Routing Protocols

#### Static Routes

Static IPv6 routes are configured in the same manner as IPv4:

---

```
Router(config)# ipv6 route <prefix>/<prefix length> {<address> | <interface>} ...
```

---

::/0 specifies a default route.

## **RIPng**

Operates like RIPv2, but on UDP/521 (RIPv2 uses UDP/520; RIPng is not backward compatible).

Multicasts updates to FF02::9.

## **EIGRP for IPv6**

Operates like IPv4 EIGRP.

## **IS-IS for IPv6**

The same IS-IS protocol used to route IPv4 was simply extended to support IPv6 by creating a new protocol identifier and two new TLVs.

## **Multiprotocol BGP4 (MP-BGP4) for IPv6**

MPBGP extends BGP to allow it to carry additional protocols like IPv6 and MPLS.

## **OSPFv3**

Successor of OSPFv2.

OSPFv3 routes and runs on IPv6.

OSPFv2 and OSPFv3 are independent and do not interact with each other.

OSPFv3 multicasts to FF02::5 for *AllOSPF Routers* and FF02::6 for *AllOSPF DRs*.

OSPFv3 routers send packets from their link-local addresses, not from global addresses.

OSPFv3 relies on IPv6 to provide authentication and encryption (via IPsec extension headers).

OSPFv3 supports multiple instances across a single link; the instance is defined in the *Instance ID* header field.

As in OSPFv2, OSPFv3 still has a 32-bit router ID. DRs and BDRs are identified by this router ID, not their IPv6 address(es).

## **OSPFv3 LSA Types**

1. **Router LSA** - Advertise router IDs within an area, from a router
2. **Network LSA** - Advertise router IDs within an area, from a DR
3. **Inter-Area Prefix LSA** - Advertise prefix from one area to another
4. **Inter-Area Router LSA** - Advertise location of an ASBR
5. **AS External LSA** - Advertise routes redistributed into OSPF
6. **Group Membership LSA** - Advertise multicast information
7. **Type 7 LSA** - Pass external routes through an NSSA
8. **Link LSA** - Advertise a router's link-local address to directly attached neighbors
9. **Intra-Area Prefix LSA** - Advertise prefixes associated with a router ID

LSA types are concatenated to 0x200 when inserted into a packet header (LSA type 1 becomes 0x2001).

## Configuring IPv6

Enable IPv6 routing and CEF:

---

```
Router(config)# ipv6 unicast-routing
Router(config)# ipv6 cef
```

---

Configure interfaces with IPv6 addresses:

---

```
Router(config-if)# ipv6 address <address>/<prefix length> [eui-64]
```

---

The eui-64 parameter causes the router to complete the last 64 bits of the address using an EUI-64 format interface ID.

## Configuring OSPFv3

OSPFv3 configuration commands follow the ipv6 keyword.

OSPFv3 interfaces are enabled under interface configuration, rather than with the network command.

---

```
Router(config)# ipv6 router ospf <process ID>
Router(config-rtr)# router-id <router ID>
Router(config-rtr)# area <area ID> range <summary range>/<prefix length>
[advertise | not-advertise] [cost <cost>]
```

```
Router(config-if)# ipv6 ospf <process ID> area <area ID> [instance <instance ID>]
Router(config-if)# ipv6 ospf priority <priority>
Router(config-if)# ipv6 ospf cost <interface cost>
```

---

## Verifying IPv6 and OSPFv3 Configuration

```
show ipv6 route

show ipv6 interface

ping ipv6

show ipv6 ospf

show ipv6 ospf interface

show ipv6 ospf neighbor

show ipv6 ospf database

clear ipv6 ospf
```

## Transitioning from IPv4 to IPv6

### Dual Stack

*Dual stack* refers to running IPv4 and IPv6 in parallel, with no communication directly between the two.

This can be implemented on routers simply by assigning an interface both an IPv4 address and an IPv6 address.

### Tunneling

Tunneling encapsulates IPv6 packets inside IPv4 packets (therefore decreasing the MTU by 20 bytes).

Tunneling requires dual stack.

Several types of tunneling may be implemented:

#### Manual Tunnels

Example configuration:

---

```
Router(config)# interface tunnel0
```

```
Router(config-if)# ipv6 address 2001:0:1:5::1/64
Router(config-if)# tunnel source 192.168.1.1
Router(config-if)# tunnel destination 192.168.2.1
Router(config-if)# tunnel mode ipv6ip
```

---

### **6-to-4 Tunnels**

6-to-4 tunnels work similar to manual tunnels but are set up automatically.

6-to-4 tunnels concatenate 2002::/16 with the 32-bit IPv4 address of the edge router, creating a 48-bit prefix.

### **Teredo**

Defined in [RFC 4380](#).

Teredo encapsulates IPv6 packets as IPv4 UDP segments, giving it the advantage of working through NAT.

### **ISATAP**

Intra-Site Automatic Tunnel Addressing Protocol; defined in [RFC 5214](#).

Treats an IPv4 network as an NBMA network, allows incremental upgrades to IPv6.

## **Translation**

Translation does not require end hosts to be dual stack.

*Stateless IP/ICMP Translation (SIIT)* translates IP header fields, and *NAT Protocol Translation (NAT-PT)* maps IPv6 addresses to IPv4 addresses.