BSCI Quick Reference Sheets

Advanced IP Addressing Issues

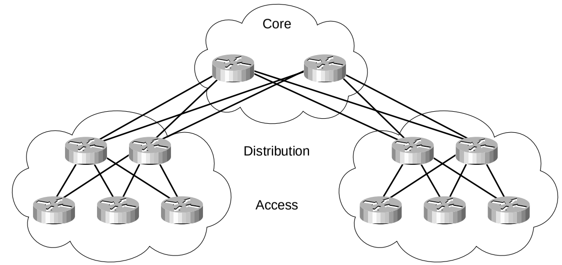
Scalable Network Design

A good network design is essential to a scalable network, so some basic design information is necessary. Networks can generally be broken into three functional layers:

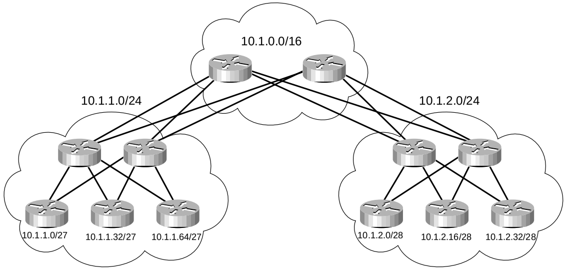
- **Access**—Where end users connect to the network. Switches (which can connect to access layer routers) are the typical network device at this layer, with VLANs, firewall, and access lists providing security and scalability. Host addressing is usually Dynamic Host Configuration Protocol (DHCP).
- **Distribution**—Access layer switches and routers aggregate to this layer. Routers and multilayer switches route between VLANs and to services that are used by many segments of the network (such as e-mail servers). This layer controls access to the core.
- **Core**—This is the network backbone. It connects the different parts of the network. You generally find fast switches here, with as little in the way of security policies as possible. Core devices can be either fully meshed or in a hub-and-spoke with redundancy, as shown in the following figure.

These components can be arranged in your network either by function or by geography:

- **Functional design**—The network is divided up by department, division, or some other type of functional group. The distribution layer bounds each group and connects it to a common core.
- **Geographical design**—A much more common design. The network is divided by location. The distribution layer allows access to groups that need to communicate and connects to a common core.



Hierarchical IP addressing is an IP addressing plan that purposefully allows points of summarization. This takes a network, breaks it into subnets by location, then further subnets those addresses by location. In the following figure, the access layer routers can summarize their networks to the distribution layer, who can then combine all the subnets into one summary to the core.



Scalable IP Addressing

A well-designed network along with a hierarchical IP addressing plan gives your network:

- **Scalability**—The network can grow to many users and many sites and still perform efficiently.
- **Predictability**—Traffic patterns are predictable and all routers choose the best paths.
- **Flexibility**—The impact of changes in the network are minimized, as the network has the ability to expand or contract.

Additionally, hierarchical IP addressing allows:

- Smaller routing tables because of summarization of networks
- More efficient use of IP addresses because addresses can be assigned contiguously

Variable-Length Subnet Masking

Variable-Length Subnet Masking (VLSM) involves using a subnet mask that gives you just the number of hosts needed. It requires the use of a classless routing protocol (one that has a field for subnet mask length in its updates.) In the previous figure, the networks on the right have a maximum of 14 hosts in each of them, while the networks on the left can have 30 hosts each. We have chosen the subnet mask based on the number of hosts in each network.

The formula to calculate the number of hosts allowed by a particular subnet mask is $\mathbf{2^n - 2}$, where **n** is the number of host bits in the subnet. The formula to calculate the number of networks gained by subnetting is simply $\mathbf{2^n}$, where **n** is the number of network bits in addition to the classful bits. The following table shows some common subnet masks, with the number of hosts and subnets.

Subnets and Hosts

Original Network: 172.16.0.0, Subnet Mask 255.255.0.0 (/16)		
Subnet Mask	**Additional Subnets**	**Hosts**
255.255.128.0 (/17)	2	32,766
255.255.192.0 (/18)	4	16,382
255.255.224.0 (/19)	8	8190
255.255.240.0 (/20)	16	4094
255.255.248.0 (/21)	32	2046
255.255.252.0 (/22)	64	1022
255.255.254.0 (/23)	128	510
255.255.255.0 (/24)	256	254
255.255.255.128 (/25)	512	126
255.255.255.192 (/26)	1,024	62
Original Network: 172.16.0.0, Subnet Mask 255.255.0.0 (/16)		

Subnets and Hosts (Continued)

Subnet Mask	Additional Subnets	Hosts
255.255.255.224 (/27)	2,048	30
255.255.255.240 (/28)	4,096	14
255.255.255.248 (/29)	8,192	6
255.255.255.252 (/30)	16,384	2

Calculating VLSM IP Addresses

To plan an IP addresses scheme using VLSM, follow these general steps:

1. Begin by looking at the network that you use to create your subnets and determine how many bits you have to work with.
2. Look at all the segments that must be assigned IP addresses and determine the most number of hosts any network require.
3. Find the subnet mask for the largest subnet first. To figure out how many bits you must allow for hosts, add 2 to the maximum number of hosts (for network and broadcast addresses) and then round up to the nearest power of 2. Calculate how many bits equal that number.
4. Find the subnet mask for the number of host bits needed by the largest network. Or use the previous table to help you determine the necessary subnet mask.
5. Assign the subnets obtained by using this subnet mask as necessary to the largest subnets.
6. Determine the number of host bits needed for the next largest subnet(s). Take an unused subnet from above and subnet that further.
7. Continue subnetting the subnets for the smaller networks, until you have IP addresses for all networks.

Typically, point-to-point links are assigned a subnet mask of 255.255.255.252 (a 30 bit mask, also written as /30).

It is recommended that you assign hosts within a VLAN to the same subnet.

Remember that you have a finite number of bits to work with; the more subnets you need, the fewer hosts you can have per subnet, and vice versa.

Route Summarization

Summarization is, in a sense, the opposite of subnetting. When subnetting, you move the subnet mask boundary to the right, creating more subnets. When summarizing, you move the subnet mask boundary to the left, thus combining subnets. *Route summarization* is announcing one route that encompasses many networks. For example, if you were in a room with 20 members of the Smith family, you could either introduce all 20 of them individually or just summarize all 20 as "the Smith family." Which would use less of your brain's resources?

Summarization uses less router resources because you have fewer networks to keep in the routing table and to announce to your neighbors.

Calculating a Summary Address

1. The routes to be summarized must share the same high-order bits.
2. Routes can be summarized in powers of 2. If you move the subnet mask 1 bit to the left, that summarizes 2 networks. Moving it 3 bits to the left summarizes 8 networks, etc.
3. Organize the networks to be summarized numerically and write each in binary to determine what high-order bits they have in common.
4. Be careful about including routes in the summary that are not assigned to you. Don't over summarize.
5. A good hierarchical IP addressing design allows for maximum summarization.
6. You must use a classless routing protocol.
7. The router must base its routing decisions on the entire 32-bit IP address and a prefix length of up to 32 bits.

Summarization Example

You have networks 10.1.24.0, 10.1.25.0, 10.1.26.0, and 10.1.27.0. The third octet is where you can summarize, because it is the one that varies. Take a look at the third octet in binary:

```
24  --  00011000
25  --  00011001
26  --  00011010
27  --  00011011
```

Notice that the first six bits are always "000110" but the last two bits vary between 0 and 1. Because the last two bits include all possible combinations of 1 and 0, it is safe to summarize these four networks into one summary route. Configure the router to advertise "10.1.24.0 255.255.252.0".

The network portion of an IP address is referred to as the *prefix*.

Classless Interdomain Routing (CIDR)

CIDR is basically route summarization done on Internet routes. It was created to decrease the size of Internet routing tables, and make better use of existing IP addresses. Service providers are assigned a block of IP addresses that they can then further subnet and assign to customers. These addresses could be several contiguous networks, or just a subnet of a larger network. CIDR allows the assignment of address blocks regardless of the classful network boundary of the blocks. The service provider then advertises to the Internet the summary route for its entire block, instead of each customer's subnet.

IP Version 6

IP version 6 (IPv6) was created to help alleviate the shortage of IP addresses, and to introduce into IP addressing the same multiple levels of hierarchy as found in telephone numbers. Benefits of IPv6 include the following:

- Larger address space than IPv4—128 versus 32 bits.
- Simpler, more efficient header than IPv4—40 bytes long, 64-bit aligned.
- Autoconfiguration options for IPv6 hosts.
- IP mobility capability built in.
- Broadcasts are not used, just multicast and anycast.
- IPSec available on all IPv6 nodes.
- Cisco IOS has methods for easing transition to IPv6.

IPv6 Address Format

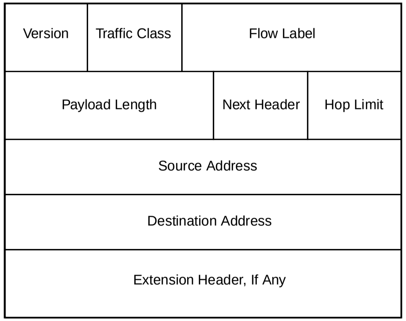
- Written in hexadecimal. Case insensitive.
- A series of eight 16-bit fields, separated by colons.
- The leading 0s in a field are optional (003c = 3c).
- Contiguous fields of all 0s can be written as "::" (two colons) once in an address.
- Allows multiple levels of aggregation.

Examples of IPv6 Addresses

```
1234:5678:90ab:cdef:1001:2202:2bad:babe

abcd::1001 (equals abcd:0000:0000:0000:0000:0000:0000:1001)

a1:beef:3add:212::1 (equals 00a1:beef:3add:0212:0000:0000:0000:0001)

:: (equals all zeros)
```

- **Anycast**—A new type of address available in IPv6. The same IP address is assigned to multiple devices that have the same function, such as a web server. Devices send a message to the anycast address and routers route it to the closest device with that address.
- **Autoconfiguration**—When a host boots up it sends a Router Solicitation (RS). The router responds with a Router Advertisement containing a 64-bit network prefix and default gateway. The host then appends its MAC address to the prefix. This eliminates the need for DHCP or manual host addressing.

IPv6 Header



- **Version**—4 bits—Contains the number 6 instead of 4.
- **Traffic Class**—8 bits—For setting quality of service options, similar to type of service (ToS) bits in IPv4.
- **Flow Label**—20 bits—Tags a flow to use with multilayer switching and for faster packet switching
- **Payload Length**—16 bits—The total length of the packet.
- **Next Header**—8 bits—Tells what header follows this one, in the data portion of the packet. Could be a Layer 4 header or an extension header.
- **Hop Limit**—8 bits—A Time-To-Live field. Each router decrements this value by 1, and the packet is dropped if it reaches 0.
- **Source and Destination Address**—128 bits each—These fields give the IP address of the source and destination device.
- **Extension Headers**—Size varies. Optional network layer information, usually read only by the destination device.

Transitioning to IPv6 from IPv4

- **Dual Stack method**—Run both versions on the routers interfaces.
- **Manual tunnel**—Use tunnels to connect areas of IPv6 separated by areas using IPv4. Tunnel endpoints must be dual stacked; tunnel is configured manually. Set up routing to go across the tunnel between the IPv6 networks.
- **6to4 tunnel**—Routers configure this automatically. Each 6 to 4 site is given a network address of 2002 concatenated with the hex equivalent of the IPv4 address of the edge router. IPv6 traffic is encapsulated in an IPv4 packet addressed to the other edge router. That router decapsulates the traffic and forwards it.
- **Routing IPv6**—IPv6 is supported in Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Integrated Intermediate System-to-Intermediate System (IS-IS), and Border Gateway Protocol (BGP).

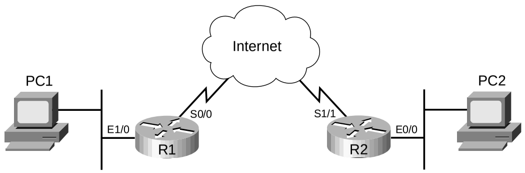
Network Address Translation (NAT)

NAT involves a network device swapping one IP address for another; it changes the source IP address of an outgoing packet and the destination IP address of an incoming one. This is done to hide IP addresses from the outside world and to enable multiple devices to use the same IP addresses.

NAT Terminology

- **Inside interface**—The router's interface that points to the inside of your company (E1/0 on R1).
- **Outside interface**—The router's interface that points to the rest of the world (S0/0 on R1).
- **Inside local IP address**—The actual IP address assigned to the host, usually a private address.
- **Inside global IP address**—What the inside local IP address has been translated to. This is what outside hosts use as the IP address for the inside host.
- **Outside global IP address**—The actual IP address of an outside host.
- **Outside local IP address**—What your network sees as the IP address of an outside host. This could be a translated address or the true address of the host.

Example: In the figure, PC1 has an address from the private addressing space. When it sends traffic through the Internet to PC2, R1 removes PC1's IP address (*inside local*) from the packet header and puts in a public address (*inside global*). PC2 replies to that public address. R1 then replaces the destination address with the original private address and sends it to PC1.



The address that PC1 has for PC2 is its outside local address. R2 might be doing NAT also, and PC2 might have a private (outside global) address that is never seen by PC1.

NAT can be done in one of three ways:

- Always translate the same inside address to the same outside address.
- Pick an outside address from a pool of addresses when an inside address needs translating.
- Translate all inside addresses to the same outside address (called *overloading* the outside address).

NAT Configuration with Access Lists and Multiple Pools

Identify what traffic to translate with a standard or extended access list. Multiple access lists can be used if different traffic must be translated to different addresses. Standard access lists translate traffic based on source address; extended access lists can translate traffic based on source, destination, and other information. P*ermit* traffic that should be translated.

Create the pools of outside addresses to use for translation. The command for this is as follows:

```
ip nat pool [pool name] [starting ip address] [ending ip address]
  [prefix-length <prefix length>| netmask <subnet mask>]
```

Next, link this pool to the access list using the following command:

```
ip nat inside source list [ACL number] pool [pool name]
```

Lastly, specify which interfaces are inside and which are outside ones. The command, at the interface config mode, is

```
ip nat inside
```

or

```
ip nat outside
```

NAT with Route Maps

Test the actions of NAT with the **show ip nat translations** command. When doing NAT with just access lists, this command shows only the source and destination addresses (*simple entries*). To see port and protocol information (*extended entries*) also, either use NAT with overloading or use a route map to specify traffic to be translated.

Route maps are described in detail in a later study sheet. Linking the access lists to a route map and using that to specify traffic to be translated give you an extended translation table.

NAT Translation Table Without Route Maps

Pro	Inside Global	Inside Local	Outside Local	Outside Global
---	63.1.1.1	10.1.1.1	---	---

NAT Translation Table with Route Maps

Pro	Inside Global	Inside Local	Outside Local	Outside Global
udp	63.1.1.1:2010	10.1.1.1:2010	82.3.1.2:69	82.3.1.2:69

Create access lists and address pools as before. Then, create route maps and refer to the access lists in the route maps. Next, link the route maps to the pools with the following command:

```
ip nat inside source route-map <route map name>  pool <pool name>
```

Understanding IP Routing

Static Routing

The easiest and most straightforward way to put a route into the routing table is to create a static route.

```
Router(config)# ip route {prefix} {mask} {next-hop} [distance] [permanent]
```

The next router along the path can be identified by an adjacent IP address or by a connecting point-to-point interface.

Static routes have a default administrative distance of 1. To create a route with a higher administrative distance (AD), include a value for distance. A static route with a very high AD, so high that the route is not used as long as a route is learned from a routing protocol, is called a *floating static route*.

The **permanent** keyword causes the route to remain in the routing table, even if the next hop is lost.

Static routing does not adjust to network changes, static routing requires time to maintain in a growing network, and static routing doesn't scale well (each new branch office results in a new static route on all other routers).

Static routes don't communicate information to neighboring routers. An important point to remember about static routes is that for routers to communicate, the neighbor router needs a reciprocal route back.

Static routes are appropriate when

- There is a single exit path from a router.
- The bandwidth required to support a routing protocol is unacceptable.
- The memory and central processing unit (CPU) resources required to support a routing protocol are unacceptable.

The Default Route

The route to 0.0.0.0/0 is a special route—the default route. A *default route* is a path to "everywhere else." If several paths should be considered, the router chooses the path with the longest prefix match, so the default is also known as the "path of last resort." A typical branch office configuration, with serial0 leading back to HQ, might look like the following:

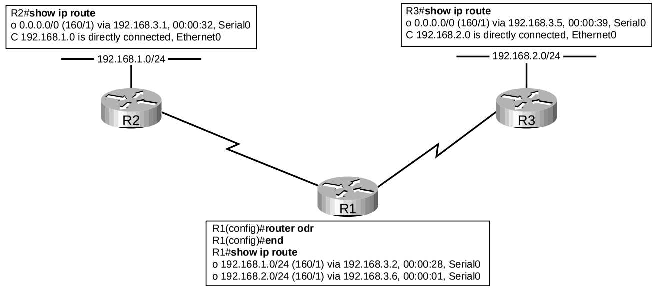
```
Router(config)# ip route 0.0.0.0 0.0.0.0 serial0
```

Dynamic Routing

Dynamic routing protocols automatically pick the best path of many and adjust to a new path as circumstances change, but they involve more advanced upkeep and overhead.

IP routing protocols include the following:

- RIP (v1 and v2)
- Interior Gateway Routing Protocol (IGRP)
- EIGRP
- OSPF
- IS-IS
- BGP



```
R2#show ip route
o 0.0.0.0/0 (160/1) via 192.168.3.1, 00:00:32, Serial0
C 192.168.1.0 is directly connected, Ethernet0
```

```
R3#show ip route
o 0.0.0.0/0 (160/1) via 192.168.3.5, 00:00:39, Serial0
C 192.168.2.0 is directly connected, Ethernet0
```

```
R1(config)#router odr
R1(config)#end
R1#show ip route
o 192.168.1.0/24 (160/1) via 192.168.3.2, 00:00:28, Serial0
o 192.168.2.0/24 (160/1) via 192.168.3.6, 00:00:01, Serial0
```

On Demand Routing

On Demand Routing (ODR) uses CDP information to build routing tables.

The Cisco Discovery Protocol (CDP) has the following characteristics:

- Runs automatically.
- Is globally disabled with the following command:

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

- Is disabled on an interface with the following:

  ```
  Router(config-if)#no cdp enable
  ```

- Sends out updates every 60 seconds and holds updates for 180 seconds. The timer is changed using the following command:

  ```
  Router(config)#cdp timer seconds
  ```

ODR has the following characteristics:

- Is appropriate only for hub and spoke.
- Spokes send classless connected routes to hub router.
- Hub router sends default route to each spoke router.
- No spoke configuration is required (CDP must be running between each spoke and the hub).
- Enabling any routing protocol on spoke router disables ODR.

- Is enabled on hub with the command:
  `Router(config)#`**`router odr`**

- Has an AD of 160; the metric is always 1.

Classful IP Routing

Classful Routing

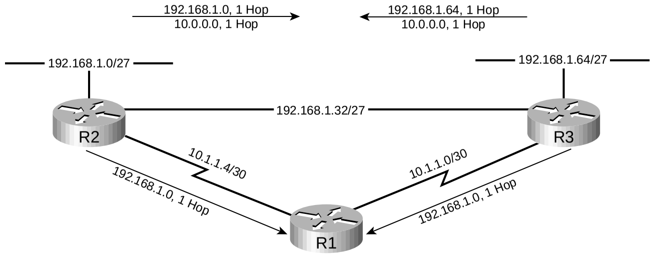
There are two classful IP routing protocols:

- RIPv1
- IGRP

Classful routing protocols do not include a subnet mask in routing updates. Therefore, the prefix length is assumed by the receiver to be either:

- The same as the receiving interface, if in the same classful network.
- Summarized to the classful mask, if in another classful network.

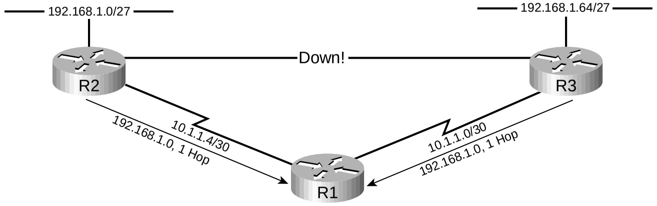
An example of RIP advertisements is shown to illustrate how subnet masks are assumed.



These assumptions lead to the two rules of classful network design:

- All subnets of a classful network must use the same mask.
- Each classful network must be contiguous.

An easy way to appreciate the issues with classful routing is to consider how the network in the following example functions when the interior link goes down. R1 receives a route to 192.168.1.0 from R2 and R3 and, therefore, load balances over two paths to reach 192.168.1.65.



The problem is that when the interior link goes down, R1 does not receive information about the changed topology because of the classful summarization. R1 continues to load balance, and therefore, only half the traffic reaches 192.168.1.65.

The ip classless Command

```
Router(config)# ip classless
```

The **ip classless** command became a default in version 12.0 of the IOS. Without the command, a router operating classfully and needing to route traffic to another location within the same classful network checks the routing table for specific routing information and, if not found, drops the traffic. This makes sense because classful networks must be built with all subnets contiguous.

With **ip classless**, the router considers routes inside the classful network, and if no route is found, the router can resort to less specific routes including the default.

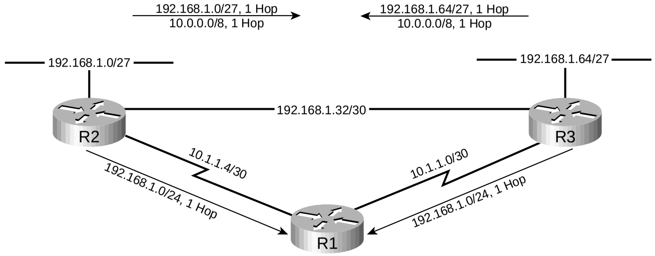
Classless IP Routing Protocols

Classless Routing

Classless IP routing protocols include RIPv2, EIGRP, OSPF, and IS-IS. Classless routing protocols include a subnet mask in routing updates, so they smoothly handle discontiguous subnets and situations of variable length subnet mask. Because of classless routing's ability to conserve address space and to handle any address space, classless routing is generally assumed to be taking place in modern IP networks.

Auto-Summary

Classless routing supports the use of arbitrary summary routes to reduce the routing table complexity. Summarization reduces the size and amount of routing protocol traffic and the amount of system resources used to maintain a routing protocol and hides (hopefully!) unimportant network details. Summarization is generally a good idea so RIPv2 and EIGRP attempt to automatically summarize at a logical place—the joining of classful networks. The problem is that this breaks discontiguous networks and ends up turned off in many current installations. The example shows the distribution of RIPv2 routes with automatic classful summarization in place. Notice that the subnet masks are included in each update, but that R1 still receives the same route to 192.168.1.0/24 from R2 and R3 and, therefore, load balances to reach an address in the Class C network. (All traffic eventually gets through, although not always along the shortest path.)



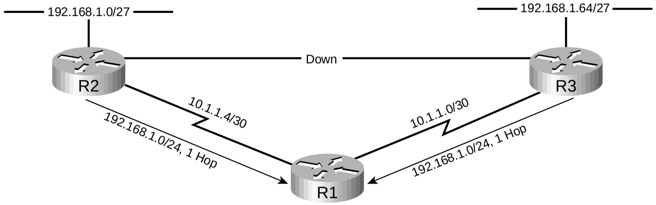
No Auto-Summary

Automatic summarization is disabled under the routing process with the **no auto-summary** command:

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

With auto-summary off, complete tables are now distributed. This means that R1 takes the shortest path to reach any destination.

The following diagram illustrates the issue with summarization.



In this case, RIP is automatically summarizing at classful network boundaries. The link between R2 and R3 is broken, but each device still advertises the classfully complete network. The result is that R1 load balances to 192.168.1.65 over the perceived two equal-cost paths, resulting in 50 percent packet loss.

When automatic summarization is turned off, this issue is repaired because each specific route is communicated.

The advantages of classless routing protocols are as follows:

- Support for Variable-Length Subnet Masking (VLSM), which is more efficient of address space
- Support for classless interdomain routing (CIDR), or summaries to blocks of classful networks

Comparing RIP and IGRP

IGRP

RIP was the original routing protocol, but was designed for a time when WAN links were commonly 56 K. By the late 1980s, it was recognized that a more advanced routing protocol was needed.

Interior Gateway Routing Protocol (IGRP) was not that protocol—OSPF was, but it wasn't coming fast enough. Cisco invented IGRP as an interim solution and carefully addressed the two largest problems with RIP:

- RIP puts out too much broadcast traffic.
- RIP assumes all links are the same speed.

Cisco addressed the issue of traffic volume by decreasing the frequency of routing updates—IGRP sends out a copy of it's routing table every 90 seconds instead of every 30 seconds like RIP. Unfortunately, the updates are broadcast just like RIP.

IGRP uses a complex metric that includes as variables the following:

- Bandwidth
- Delay
- Load (not used by default)
- Reliability (not used by default)

Maximum transmission unit (MTU) is tracked; this was meant to allow path MTU discovery, but it was never implemented. Hop count is also tracked, but strictly as a loop recognition technique.

IGRP is a Cisco proprietary classful IP routing protocol. By default, RIP will be enabled on all interfaces that fall within the networks specified in the **network** command.

Unequal cost load balancing is a unique feature of IGRP and Enhanced IGRP (EIGRP). When unequal cost load balancing is active, the best metric to a route is multiplied by a whole-number variance. Any path lower than the product is load-balanced proportionally.

RIP Version 1

RIP v1 is a standard-based (RFC 1058) classful routing protocol that uses hop count as its metric. RIP broadcasts a copy of its table every 30 seconds and, like all other IP routing protocols on a Cisco router, supports equal cost load balancing.

RIP Version 2

RIP v2 is a standards-based (RFC 1721, 1722, and 2453) classless routing protocol that uses hop count as its metric. Version 2 addresses several of the issues with Version 1:

- Version 2 is classless (VLSM).
- Version 2 supports route summarization and , by default, automatically summarizes at classful network boundaries. Automatic summarization can be disabled, and manual summarization configured.
- Version 2 uses multicasts instead of broadcasts
- Version 2 allows for authentication (plain text or MD5).

RIP is configured by entering the RIP router configuration mode and identifying the classful networks within which it should run. All interfaces on the router in the assigned networks run RIP.

```
Router(config)# router rip
Router(config-router)# network 192.168.1.0
```

By default, RIP sends version 1 and receives version 1 and 2 routes. To configure RIP to run only version 2 use the **version** command.

```
Router(config-router)# version 2
```

Configure specific interfaces to run either version 1 or 2 or both using the **ip rip send** or **ip rip receive** commands.

```
Router(config-router)# ip rip send version 2
Router(config-router)# ip rip receive version 2
```

Finally, configure a summary route out a particular interface using the **ip summary-address rip** command. Of course, automatic summarization is usually disabled also (**no auto-summary**). The following example advertises a default route out ethernet0 and a summary route 172.16.104.0/22 out serial0.

```
Router(config)# int e0
Router(config-if)# ip summary-address rip 0.0.0.0 0.0.0.0
Router(config)# int s0
Router(config-if)# ip summary-address rip 172.16.104.0 255.255.252.0
```

IP Routing Protocols

Administrative Distance

Cisco routers are capable of supporting several IP routing protocols concurrently. When identical prefixes are discovered from two or more separate protocols, administrative distance (AD) is used to discriminate between the paths. AD is really a poor choice of words—*trustworthiness* would be a better name. Routers use paths with the lower AD.

The following table lists the default values for various routing protocols. Of course, you have several ways to change AD for a routing protocol or for a specific route.

Information Source	AD
Connected	0
Static	1
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
ODR	160

Information Source	AD
External EIGRP	170
Internal BGP	200
Unknown	255

Building the Routing Table

The router builds its routing table by ruling out invalid routes and carefully considering the remaining advertisements. The procedure is as follows:

- For each route received, verify the next hop. If invalid, discard the route.
- If more than one specific valid route is advertised by a routing protocol, choose the path with the lowest metric.
- If more than one specific valid route is advertised by different routing protocols, choose the path with the lowest AD.
- Routes are considered identical if they advertise the same prefix and mask, so 192.168.0.0/16 and 192.168.0.0/24 are separate paths and are entered into the routing table separately.

Comparing Routing Protocols

Generically, two things should always be considered in choosing a routing protocol: fast convergence speed and support for VLSM. EIGRP, OSPF, and IS-IS meet these criteria.

EIGRP is Cisco proprietary, but simple to configure and support. OSPF is standards based, but difficult to implement and support. IS-IS is an OSI network layer protocol that can carry IP information. It is fairly simple to configure and support, but not as full featured as OSPF.

The following table compares critical parts of all the routing protocols on the BSCI test.

Property	RIPv2	EIGRP	OSPF	IS-IS	BGP
Algorithm	Distance Vector	Advanced Distance Vector	Link State	Link State	Path vector
Summarization	Auto and arbitrary	Auto and arbitrary	Arbitrary	Arbitrary	Auto and arbitrary
VLSM	Yes	Yes	Yes	Yes	Yes
Converge	Minutes	Seconds	Seconds	Seconds	Minutes
Timers: Update (hello/dead)	Update every 30 seconds	Triggered, (LAN 5/15, WAN 60/180)	Triggered, LSA refresh 30 minutes. (NBMA 30/120, LAN 10, 40)	Triggered (10/30)	Triggered (60/180)

EIGRP

EIGRP Overview

EIGRP is a proprietary classless routing protocol that uses a complex metric that is based on bandwidth and delay. EIGRP addresses several issues with IGRP. The following are some features of EIGRP:

- Quick convergence.
- Support for VLSM.
- Is conservative of network bandwidth.
- Support for IP, AppleTalk, and IPX.
- Support for unequal-cost proportional load-balancing.
- Classless.
- Supports route summarization by default and automatically summarizes at classful network boundaries. Manual summarization can also be done with EIGRP.
- Uses multicasts (and unicasts where appropriate) instead of broadcasts.
- EIGRP supports authentication.

Database Structure

EIGRP uses three tables:

- The neighbor table is built from EIGRP hellos and used for reliable delivery.
- The topology table contains EIGRP routing information for best paths and loop-free alternatives.
- The routing table is used by EIGRP and other routing protocols for all best paths.

Route Selection

An EIGRP router receives advertisements from each neighbor that list the advertised distance and feasible distance to a route. *Advertised distance* is the metric from the neighbor to the network. *Feasible distance* is the metric from this router, through the neighbor, to the network.

The EIGRP path with the lowest feasible distance is called the successor path. Any EIGRP paths that have a lower advertised distance than the metric of the successor are guaranteed loop free and called feasible successors.

EIGRP Metric

The EIGRP metric is as follows:

$$Metric = 256(k1 \times \frac{10^7}{BW_{\min}} + \frac{k2 \times BW_{\min}}{256 - load} + k3 \times \sum Delays)(\frac{k5}{Reliability + k4})$$

The k values are constants. The defaults are k1 = 1, k2 = 0, k3 = 1, k4 = 0, and k5 = 0. If k5 = 0, the final part of the equation (k5 / [rel + k4]) is ignored.

BW_{min} is the minimum bandwidth along the path—the choke point bandwidth.

Delay values are associated with each interface. The sum of the delays (in 10s of microseconds) is used in the equation.

Taking the constants into account, the equation becomes this:

$$Metric = 256(\frac{10^7}{BW_{\min}} + \sum Delays)$$

If default k values are used, this works out to be 256 (BW + cumulative delay).

Bandwidth is the largest contributor to the metric. The delay value allows us to choose a more direct path when bandwidth is equivalent.

The EIGRP metric is 256 times the IGRP metric. The two automatically redistribute and algorithmically adjust metrics if they are configured on the same router for the same autonomous system.

EIGRP Messages

Packets

EIGRP uses five packet types:

- **Hello**—Identifies neighbors and serves as keepalive
- **Update**—Reliably sends route information
- **Query**—Reliably request specific route information
- **Reply**—Reliable response to query
- **ACK**—Acknowledgement

EIGRP is reliable, but not all traffic requires an ACK. Hellos are not acknowledged, and the acknowledgement of a query is a reply.

When EIGRP first starts, it uses hellos to build a neighbor table. Neighbors are directly attached routers that agree on AS number and k values (timers don't have to agree). Subsequent traffic is sent with the expectation that each identified neighbor will respond.

If a reliable packet is not acknowledged, EIGRP periodically retransmits the packet to the non-responding neighbor as a unicast. EIGRP has a window size of one, so no other traffic is sent to this neighbor until it responds. After 16 retransmissions the neighbor is removed from the neighbor table.

Hellos also serve as keepalives. A neighbor is considered lost if no hello is received within three hello periods (called the hold time). The default timers are as follows:

- 5 seconds/15 seconds for multipoint circuits with bandwidth greater than T1 and for point-to-point media
- 60 seconds/180 seconds for other multipoint circuits with bandwidth less than or equal to T1

The neighbor table can be seen with the command **show ip eigrp neighbors**.

The process of route exchange between two EIGRP routers is as follows:

1. Router A sends out a hello.
2. Router B sends back a hello and an update. The update contains routing information.
3. Router A acknowledges the update.
4. Router A sends its update.
5. Router B acknowledges.

Hellos are used as keepalives from this point on. Additional route information is sent only if a route is lost or new route discovered.

The exchange process can be viewed using **debug ip eigrp packets**, and the update process can be seen using **debug ip eigrp**.

EIGRP DUAL

Diffusing Update Algorithm (DUAL)

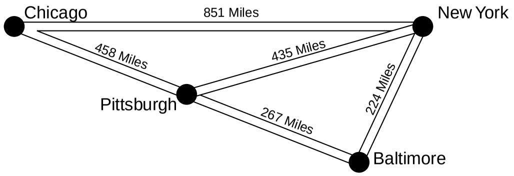
DUAL is the algorithm used by EIGRP to choose best paths by looking at advertised distance (AD) and feasible distance (FD). Advertised distance is the metric from the neighbor to the destination. Feasible distance is the metric from this router, through the neighbor, to the destination. The path with the lowest metric is called the *successor* path. EIGRP paths with a lower advertised distance than the feasible distance of the successor path are guaranteed loop free and called *feasible successors*. If the successor path is lost, the router might start using the feasible successor immediately without fear of loops.

Feasible successors are not always available. If a successor path is lost and no backup path is identified, the router sends out queries on all interfaces trying to identify an alternate path.

Route Selection

An easy way to understand this is to consider a driving example. Located in Pittsburgh, you are told of three ways to New York:

- A direct path (435 miles)
- A path through Baltimore (AD: 224 miles, FD: 491 miles)
- A path through Chicago (AD: 851 miles, FD: 1319 miles)



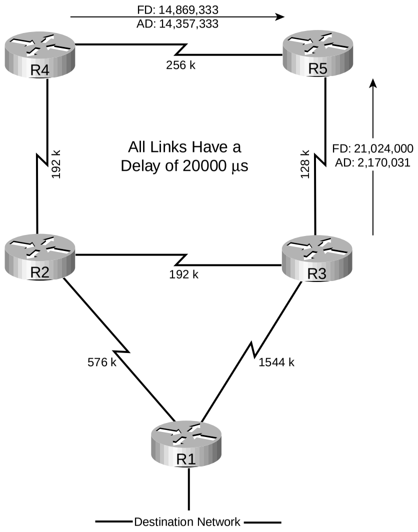
The direct path would be the successor path; it has the lowest metric.

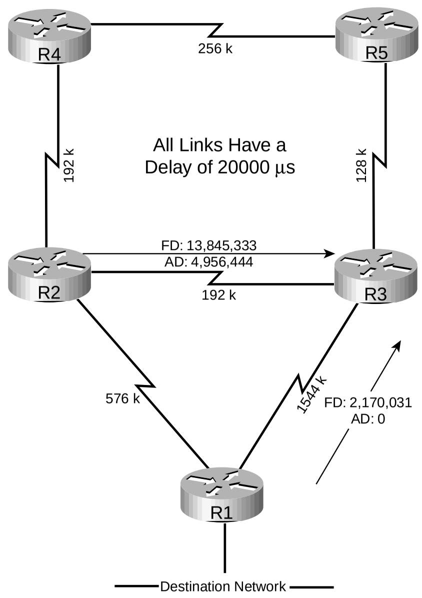
The path through Baltimore would be a feasible successor—Baltimore is closer to New York (224 miles) than Pittsburgh (435 miles), so we're sure that going through Baltimore to New York doesn't involve looping back through Pittsburgh.

The path through Chicago would not be a feasible successor because Chicago (851 miles) is further away from New York than Pittsburgh (435 miles) and we can't say with certainty that traveling through Chicago to New York doesn't involve coming back through Pittsburgh.

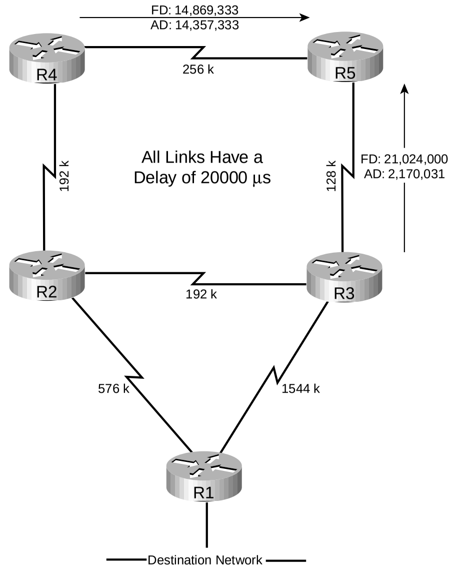
Network Example

The following diagrams show EIGRP advertisements from the destination network to R3 and R5. R5 chooses R4 as the successor path because it is offering the lowest Feasible Distance. The Advertised Distance from R3 indicates that passing traffic through R3 will not loop, so R3 is a feasible successor.





How does R3 choose its path? R1 will be the successor, but no feasible successor exists. If the direct path to R1 is lost then R3 has to query its neighbors to discover an alternative path. It waits to hear back from R2 and R5, but ultimately decides that R2 is the new successor.



Configuring EIGRP

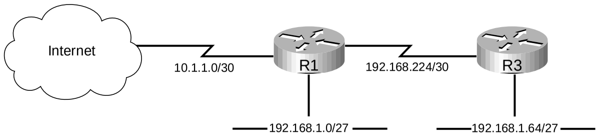
Configuration

EIGRP is configured by entering router configuration mode and identifying the classful networks within which it should run. When setting up EIGRP, an autonomous system number must be used (7 is used in the example). Autonomous system numbers for EIGRP are not globally assigned, so most installations just make up numbers. Autonomous system numbers must agree for two routers to form a neighbor relationship and to exchange routes.

```
Router(config)# router eigrp 7
Router(config-router)# network 192.168.1.0
```

If a router has two interfaces—e0/0 (192.168.1.1/27) and e0/1 (192.168.1.33/27)—and needs to run only EIGRP on e0/0, the wildcard mask option can be used with the network command:

```
Router(config-router)# network 192.168.1.0 0.0.0.31
```



Creating a Default Route

You can produce a default route in EIGRP in three ways:

- R1 can specify a default network:

  ```
  R1(config)# ip default-network 10.0.0.0
  ```

  R3 now sees a default network with a next hop of R1.

- Produce a summary route:

  ```
  R1(config)# interface s0
  R1(config-if)# ip summary-address eigrp 7 0.0.0.0 0.0.0.0
  ```

  This passes a default route from R1 out its serial0 interface toward R3.

- Create a static default route and then include network 0.0.0.0 in EIGRP:

  ```
  R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2
  R1(config)# router eigrp 7
  R1(config-router)# network 0.0.0.0
  ```

Troubleshooting EIGRP

The most straightforward way to troubleshoot EIGRP is to inspect the routing table—**show ip route**. To filter the routing table and show only the routes learned from EIGRP use **show ip route eigrp**.

show ip protocols is always the first place to check when investigating routing protocol issues. Use this command to verify autonomous system, timer values, identified networks, and EIGRP neighbors (routing information sources).

EIGRP specific issues can be investigated using show **ip eigrp topology**. This shows the EIGRP topology table and identifies successors and feasible successors.

Advanced EIGRP

Summarization

Just like RIPv2, EIGRP defaults to automatically summarizing at classful network boundaries. This is commonly disabled and arbitrary summarization is employed.

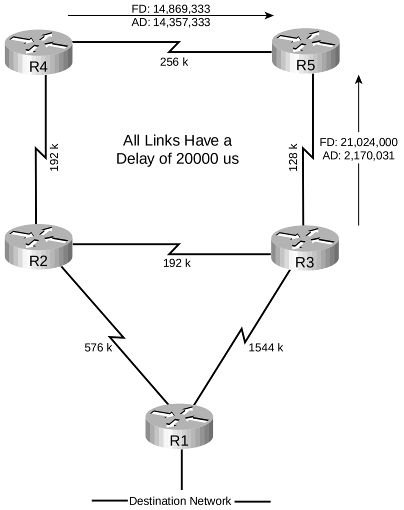
```
Router(config-router)# no auto-summary
```

Summaries can be produced manually on any interface. When a summary is produced, a matching route to null0 also becomes active as a loop prevention mechanism. Configure a summary route out a particular interface using the **ip summary-address eigrp** ***AS*** command. The following example uses EIGRP AS 7 and advertises a default route out ethernet0 and a summary route 172.16.104.0/22 out serial0.

```
Router(config)# int e0
Router(config-if)# ip summary-address eigrp 7 0.0.0.0 0.0.0.0
Router(config)# int s0
Router(config-if)# ip summary-address eigrp 7 172.16.104.0 255.255.252.0
```

Load Balancing

EIGRP, like most IP routing protocols on a Cisco router, automatically load balances over equal cost paths. What makes EIGRP unique is that it proportionally load balances over unequal cost paths. A variance is specified and load balancing is used for any path with a metric of less than the product of the variance and the best metric.



In this example, R5 uses the path through R4 since it offers the lowest metric (14,869,333). To set up unequal cost load balancing, assign a variance of 2, and now R5 uses all paths with a metric less than 29,738,666. This includes the path through R3.

```
R5(config-router)# variance 2
```

WAN Bandwidth

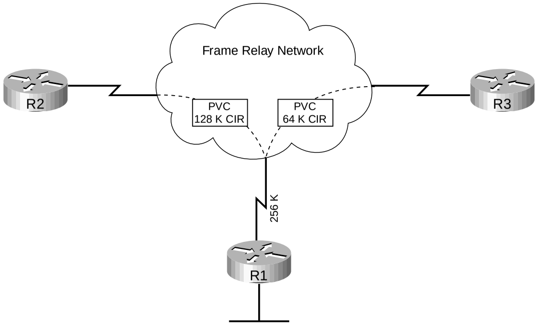
Other routing protocols burst to use all available link bandwidth. Because routing protocol traffic is treated as having a higher priority, this can sometimes lock out data traffic over WAN links. EIGRP is unique in offering a way to control this. By default, EIGRP limits itself to bursting to half the link bandwidth. This limit is configurable using the **ip bandwidth-percent** command. The following example assumes EIGRP AS 7 and limits EIGRP to a fourth of link bandwidth:

```
Router(config-if)# ip bandwidth-percent eigrp 7 25
```

The default value is acceptable in most cases. The real issue with WAN links is that the router assumes that each link has 1544 kbps bandwidth. If serial0 is attached to a 128 k fractional T1, EIGRP assumes it can burst to 768 k and could overwhelm the line. This is rectified by correctly identifying link bandwidth.

```
Router (config)# int serial 0
Router (config-if)# bandwidth 128
```

One situation suggests itself where all these techniques can be combined—Frame Relay.



In this configuration, R1 has a 256 K connection to the Frame Relay network and two permanent virtual circuits (PVCs) with committed information rates (CIR) (minimum guaranteed bandwidth) of 128 K and 64 K. What value should be used for the interface bandwidth in this case? The usual suggestion is to use the CIR, but the two PVCs have different CIRs. Specifying a bandwidth of 128 on the main interface leads EIGRP to assume a bandwidth of 64 for each PVC, but causes the router to incorrectly report utilization through Simple Network Management Protocol (SNMP). We could use the **bandwidth-percent** command to allow a true bandwidth value while adjusting the burst rate to 64 k for each PVC.

```
R1(config)# int serial 0
R1 (config-if)# bandwidth 256
R1 (config-if)# ip bandwidth-percent eigrp 7 25
```

A better solution is to use subinterfaces and identify bandwidth separately. In this case, s0.1 bursts to 64 k, and s0.2 bursts to 32 k.

```
R1(config)#int serial 0.1
R1 (config-if)# bandwidth 128
R1(config)# int serial 0.2
R1 (config-if)# bandwidth 10
R1 (config-if)# ip bandwidth-percent eigrp 7 320
```

Notice that an arbitrarily low value has been used for the second PVC.

OSPF Overview

Dijkstra Algorithm

OSPF is an open standard classless routing protocol that converges quickly and uses cost as a metric (cost is by default automatically associated with bandwidth by Cisco IOS).

OSPF is a link-state routing protocol and uses Dijkstra's Shortest Path First (SPF) algorithm to determine best paths. The first responsibility of a link-state router is to create a database that reflects the structure of the network.

The router exchanges hellos with each neighbor, learning Router ID (RID) and cost.

The router then constructs a Link State Advertisement (LSA) with the RIDs and cost to each neighbor.

Each router in the routing domain shares its LSA with all other routers. Each router keeps the complete set of LSAs in a table—the Link State Database.

Each router runs the SPF algorithm to compute best paths.

Description of the SPF Algorithm

The following section is helpful for understanding the operation of the SPF algorithm, but it is not tested in this depth on the BSCI exam. The SPF algorithm uses four tables:

- **Link State Database**—All LSAs
- **Tentative—**A scratch pad table for the algorithm
- **Path**—The best OSPF path
- **Routing Table**—The best path given all sources

The SPF algorithm operates on each router in five steps:

Step 1 Place myself in path table with a cost of zero.

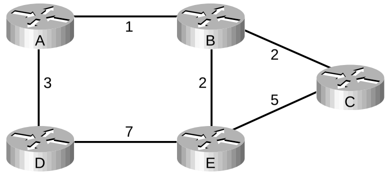
Step 2 For the new entry in the path table, look at its LSA in the Link State Database. Add the path cost to the node to the LSA cost. For each neighbor, if not already in the path or tentative table with a better cost, place in the tentative table.

Step 3 If the tentative table is empty, stop.

Step 4 Find the lowest cost entry in the tentative table and move to it to the path table.

Step 5 Go to Step 2.

For example, consider the following network.



Router B builds an LSA that says, "My name is B, and I have a neighbor A with a cost of 1, neighbor C with a cost of 2, and neighbor E with a cost of 2."

The LSAs are distributed, and the Dijkstra algorithm is run.

Router B starts by placing itself at the root of the network with a cost of zero. B's LSA is added to the tent.

Topology	Path	Tent
B	B cost 0	A cost 1
		C cost 2
		E cost 2

The lowest cost entry in tent is A, so it is moved to the path and its LSA is incorporated in the tentative database. Because B is already in the path, an entry for B (via A) is not added to the tentative database.

Topology	Path	Tent
B	B cost 0	C cost 2
|	A cost 1	D via A cost 1 + 3 = 4
A		E cost 2

The lowest cost entry in tent is now C or E. Choosing C, it is moved to the path and its LSA is incorporated in the tentative database. Because E is already in the tentative database with a lower cost, the entry for E (via C) is not added to the tentative database.

Topology	Path	Tent
B /\\ A C	B cost 0 A cost 1 C cost 2	D via A cost 1 + 3 = 4 E cost 2

The lowest cost entry in tent is now E. It is moved to the path and its LSA is incorporated in the tentative database. Because D is already in the tentative database with a lower cost, the entry for D (via E) is not added to the tentative database.

Topology	Path	Tent
B /\\ \\ A C E	B cost 0 A cost 1 C cost 2 E cost 2	D via A cost 1 + 3 = 4

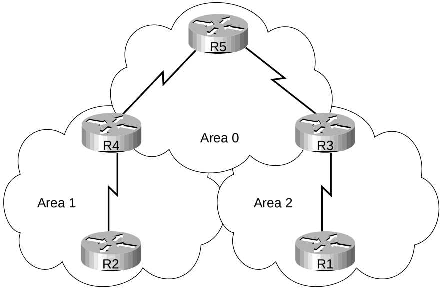
The lowest cost entry in tent is now D. It is moved to the path and its LSA is incorporated in the tentative database. Because E and A are already in the path database with a lower cost, the tentative database is now empty.

Topology	Path	Tent
B /\\ \\ A C E | D	B cost 0 A cost 1 C cost 2 E cost 2 D cost 4	

The algorithm is now complete, and the best paths from B to any other location are now incorporated from the path database into the routing table.

Network Structure

OSPF routing domains are broken up into areas. The SPF algorithm runs within an area, and inter-area routes are passed between areas. A two-level hierarchy to OSPF areas exists—other areas are always attached directly to area 0 and only to area 0.



OSPF areas do the following:

- Minimize the number of routing table entries and the number of times the SPF algorithm is run.
- Contain LSA flooding to a reasonable area.
- Are recommended to contain 50–100 routers.

Router roles are defined as well.

An internal router has all interfaces in one area (R5, R2, R1).

An Area Border Router (ABR) has interfaces in two or more areas (R3, R4).

An Autonomous System Boundary Router (ASBR) has interfaces inside and outside the OSPF routing domain.

LSA Update

Each router maintains a database of the latest received LSAs. Each LSA is numbered with a sequence number, and a timer is run to age out old LSAs.

When a LSA is received, it's compared to the LS database. If it is new then it is added to the database and the SPF algorithm is run. If it is from a Router ID that is already in the database, then the sequence number is compared, and older LSAs are discarded. If it is a new LSA, it is incorporated in the database, and the SPF algorithm is run. If it is an older LSA, the newer LSA in memory is sent back to whoever sent the old one.

Configuring Single Area OSPF

Configuration

OSPF is configured by entering router configuration mode and identifying the range of interface addresses on which it should run and the areas they are in. When setting up OSPF, a process ID must be used (8 is used in the example), but the process ID does not have to agree on different OSPF devices for them to exchange information. The network statement uses a wildcard mask and can specify any range from a single address to all addresses.

```
Router(config)# router ospf 8
Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
```

Troubleshooting OSPF

The first place OSPF issues are noticed is when inspecting the routing table—**show ip route**. To filter the routing table and only show the routes learned from OSPF use **show ip route ospf**.

show ip protocols should be used next. It offers a wealth of information for any routing protocol issue. Use this command to verify parameters, timer values, identified networks, and OSPF neighbors (“routing information sources”).

Because wildcard masks sometimes incorrectly group interfaces to areas, another good place to check is **show ip ospf interfaces**. This shows the interfaces on which OSPF is running and their current correct assigned area.

show ip ospf shows the RID, timers, and counters. **show ip ospf neighbors** shows the OSPF neighbor table, identifies adjacency status, and reveals the designated router and backup designated router.

Router ID

The SPF algorithm is used to map the shortest path between a series of nodes. The issue is that an IP router is not identified by a single number—its interfaces are. For reasons of compatibility with the protocol a single address is used as the “name” of the router—the Router ID (RID).

By default, the Router ID is the highest loopback IP address. If no loopback addresses are configured, the RID is the highest IP address on an active interface when the OSPF process is started. The router ID is selected when OSPF starts and—for reasons of stability—is not changed until OSPF restarts. Of course, the OSPF process can be restarted by rebooting or by using the command **clear ip ospf process**. Either choice affects routing in your network for a period of time and should be used only with caution.

A loopback interface is a virtual interface. A loopback address is configured by creating an interface and assigning an IP address.

```
Router(config)# interface loopback0
Router(config-if)# ip address 10.0.0.1 255.255.255.255
```

The loopback address does not have to be included in the OSPF routing process, but it often is to provide a location to ping or trace to.

A way to short-circuit the RID selection is to assign it using the OSPF **router-id** command.

```
Router(config)# router ospf 8
Router(config-router)# router-id 10.0.0.1
```

Low-Level OSPF

Packets

OSPF uses five packet types:

- **Hello**—Identifies neighbors and serves as keepalive.
- **Link State Request** (**LSR**)—A request for an LSU. Contains the type of LSU requested and the ID of the router requesting it.
- **Database Description** (**DBD**)—A summary of the Link State Database, including the RID and sequence number of each LSA in the Link State Database.
- **Link State Update** (**LSU**)—Contains a full LSA (link state advertisement) entry. An LSA includes topology information, for example the RID of this router and the RID and cost to each neighbor. One LSU can contain multiple LSAs.
- **Link State Acknowledgment** (**LSAck**)—Acknowledges all other OSPF packets (except hellos).

All five packet types are directly placed into an IP packet (IP protocol 89) using a common OSPF header.

Neighborship

OSPF routers send out periodic multicast packets to introduce themselves to other routers on a link. They become neighbors when they hear their own router ID included in the hello of another router, thus proving bidirectional communication.

OSPF traffic is multicast to either of two addresses:

224.0.0.5—All SPF Routers
224.0.0.6—All SPF DRs

OSPF routers, if certain parameters agree, become neighbors when they see themselves in the “neighbors” field of an incoming hello. The following must match:

- Timer values
- Area ID
- Password (if used)
- Common subnet
- Stub area flag (T/F)

OSPF routers can be neighbors without being adjacent. Only adjacent neighbors exchange routing updates and synchronize their databases. On a point-to-point link, an adjacency is established between the two routers when they can communicate. On a multiaccess link, each router establishes an adjacency only with the DR and the backup DR (BDR).

Hellos also serve as keepalives. A neighbor is considered lost if no Hello is received within four Hello periods (called the dead time). The default timers are as follows:

- 10 seconds/40 seconds for LAN and point-to-point interfaces
- 30 seconds/120 seconds for nonbroadcast multiaccess (NBMA) interfaces

The neighbor table can be seen with **show ip ospf neighbors**.

The process of route exchange between two OSPF routers is as follows:

1. **Down state:** Router A sends a hello that lists no neighbors.
2. **Init state:** Router B sends a hello that lists A as a neighbor.

   Two-way state: Router A sees Router B’s hello and adds it to the neighbor database. (If one router is not DR or BDR and this is a multiaccess link, process stops here.)
3. **Exstart state:** One router asserts itself as the lead in exchanging routes.
4. **Exchange state:** The lead router sends its DBD listing the LSAs in its LS database by RID and sequence number.
5. The other router replies with its DBD.
6. Each router sends an LSAck.
7. **Loading state:** Each router compares the DBD received to the contents of its Link State database. It then sends a Link State Request asking for missing or outdated LSAs.
8. The neighbor replies to the request.
9. The router acknowledges receipt.
10. **Full state:** Link State Databases are synchronized.

Sequence Numbers

OSPF sequence numbers are 32 bits. The first legal sequence number is 0x80000001. Larger numbers are more recent.

Normally, the sequence number changes only under two conditions:

- The LSA changes because a route is added or deleted.
- The LSA ages out (LSAs are updated every half hour, even if nothing changes).

The command **show ip ospf database** shows the age (in seconds) and sequence number for each RID.

Troubleshooting

The neighbor initialization process can be viewed using **debug ip ospf adjacencies**. All OSPF traffic can be seen from **debug ip ospf packet**.

OSPF Network Types

Expectations of Dijkstra’s SPF Algorithm

The SPF algorithm builds a directed graph—paths made up of a series of points connected by direct links. One of the consequences of this “directed graph” approach is that the algorithm has no way to handle a multiaccess network. The solution used is to elect one router to represent the entire segment—a designated router (DR).

Designated Routers

Point-to-point links fit the SPF model perfectly and don’t need any special modeling method. On a point-to-point link, no DR is elected and all traffic is multicast to 224.0.0.5.

On a multiaccess link one of the routers is elected as a designated router (DR) and another as a backup DR (BDR). All other routers on that link become adjacent only to the DR and BDR, not to each other (they stop at the *two-way state*). The DR is responsible for creating and flooding a network LSA (type 2) advertising the multiaccess link. Non-DR (DROTHER) routers communicate with the DRs using IP address 224.0.0.6. The DRs use IP address 224.0.0.5 to pass information to other routers.

The DR and BDR are elected as follows:

- A starting router listens for OSPF hellos. If none are heard within the dead time, it declares itself the DR.
- If one or more other routers are heard, the router with the highest OSPF priority is elected DR, and the election process starts again for BDR. Priority of zero removes a router from the election.
- If two or more routers have the same OSPF priority, the router with the highest RID is elected DR, and the election process starts again for BDR.

- DR election does not take place again unless the DR or BDR are lost. Because of this, the DR is sometimes the first device started with a non-zero priority.

As mentioned, the best way to specify a router to be the DR is to use the OSPF priority. The default priority is one. A priority of zero means that a router does not act as DR or BDR—it can be only a DROTHER. Priority can be set with the **ip ospf priority** command in interface configuration mode.

```
Router(config)# int ethernet 0
Router(config-if)# ip ospf priority 2
```

Nonbroadcast Multiaccess Networks

Routing protocols assume that multiaccess links support broadcast and have full connectivity (from any device to any device). In terms of OSPF, this means the following:

- All Frame Relay or ATM maps should include the broadcast attribute.
- DR and BDR should have full VC connectivity to all other devices.
- Hub-and-spoke environments should either have the DR as the hub or be configured using point-to-point subinterfaces.
- Partial mesh environments should be configured using point-to-point subinterfaces, especially when no single device has full connectivity to all other devices. If there is a subset of the topology with full connectivity, then that subset can use a multipoint subinterface.
- Full mesh environments can be configured on the main interface, but often logical interfaces are used to take advantage of the other benefits of subinterfaces.

OSPF supports five network types:

- **NBMA**—Default for multipoint serial interfaces. RFC-compliant mode that uses DRs and requires manual neighbor configuration.
- **Point-to–multipoint** (**P2MP**)—Doesn't use DRs so adjacencies increase logarithmically with routers. Resilient RFC compliant mode that automatically discovers neighbors.
- **Point-to-multipoint nonbroadcast** (**P2MNB**)—Proprietary mode that is used on Layer 2 facilities where dynamic neighbor discovery is not supported. Requires manual neighbor configuration.
- **Broadcast**—Default mode for LANs. Uses DRs and automatic neighbor discovery. Proprietary when used on WAN interface.
- **Point-to–point** (**P2P**)—Proprietary mode that discovers neighbors and doesn't use DR.

Mode	RFC or Cisco	DR	Adjacencies	Neighbor Discovery	Timers Hello/Dead	Topo
NBMA	RFC	Yes	2n–3	Manual	30/120	Full
P2MP	RFC	No	n(n–1)/2	Automatic	30/120	Any
P2MPNB	Cisco	No	n(n–1)/2	Manual	30/120	Any
Broadcast	Cisco	Yes	2n–3	Automatic	10/40	Full
P2P	Cisco	No	1	Automatic	10/40	P2P

The interface type is selected—assuming the default is unsatisfactory—using the command **ip ospf network**:

```
Router(config-if)# ip ospf network point-to-multipoint
```

When using the NBMA or P2MP non-broadcast mode, neighbors must be manually defined under the routing process:

```
Router(config-router)# neighbor 172.16.0.1
```

OSPF Advertisements and Cost

LSA Types

OSPF advertises many different things in different ways. The type of advertisement establishes context for its contents.

Type	Description	Routing Table Symbol
1	Intra-area LSA of a router	O
2	Intra-area LSA of DR	O
3	Inter-area route passed by ABR	O IA
4	Inter-area route to an ASBR	O IA
5	External type 2 route from ASBR. Metric doesn't accrue in OSPF (default).	O E2
5	External type 1 route from ASBR. Metric accumulates in OSPF.	O E1
6	Multicast (not used by Cisco)	N/A
7	Not-so-stubby area (NSSA) external route passed through stubby area	O N2 O N1

Cost

By default, Cisco assigns a cost to each interface that is inversely proportional to 100 Mbps.

$$Cost = \frac{100\ Mbps}{Bandwidth}$$

The default formula doesn't differentiate between fast ethernet and gigabit ethernet, for example. In such cases, the cost formula can be adjusted using the **auto-cost** command. Values for bandwidth (in Kb) up to 4,294,967 are permitted (1 Gb is shown in the example).

```
Router(config-router)# auto-cost reference-bandwidth 1000000
```

The cost can also be manually assigned under interface configuration mode. Cost is a 16-bit number, so can be any value up to 65,535.

```
Router(config-router)# ip ospf cost 27
```

OSPF Summarization

OSPF Benefits of Summarization

All routing protocols benefit from summarization, but OSPF especially benefits because it is already sensitive to the memory and CPU speed of the routers. Summarization prevents topology changes from being passed outside an area and thus saves routers in other areas from having to run the SPF algorithm. OSPF can produce summaries within a classful network (VLSM) or summaries of blocks of classful networks (CIDR).

Creating a Summary

Inter-area summarizations are created on the ABR under the router process using the **area range** command. The following command advertises 172.16.0.0/12 from area zero.

```
Router(config-router)# area 0 range 172.16.0.0 255.240.0.0
```

Summarizing external routes is done on an ASBR with the **summary-address** command under the routing process. The following example summarizes a range of external routes to 192.168.0.0/16 and injects a single route into OSPF.

```
Router(config-router)# summary-address 192.168.0.0 255.255.0.0
```

Creating a Default Route

The default route is a special type of summarization—it summarizes everything. This provides the ultimate benefit of summarization by reducing the routing information to a minimum. Several different ways to use the router IOS to place a default route into OSPF.

The best-known way to produce an OSPF default is to use the **default-information** command:

```
Router(config-router)# default-information originate [always]
```

This command, without the keyword **always**, readvertises a default learned from another source into OSPF. If the **always** keyword is present, OSPF advertises a default even if one does not already exist.

A default summary can also be produced using the **summary-address** command or the **area range** command (to just produce a default into an area). Reducing routing information in non-backbone areas is a common requirement because these routers are the most vulnerable, in terms of processor and speed, and the links that connect them have the least bandwidth. A specific concern is that an area will be overwhelmed by external routing information.

Making an area a **stub** area forces its ABR to drop all external (type 5) routes and replaces them with a default route. In some cases an area can be made totally stubby (**stub no-summary**), in which case all intra-area and external routes are replaced by a default.

```
Router(config-router)# area 7 stub [no-summary]
```

Stub areas are attractive because of their low overhead. They do have some limitations, including the following:

- Stub areas can't include a virtual link.
- Stub areas can't include an ASBR.
- Stubbiness must be configured on all routers in the area.

A third kind of stubby area is a not-so-stubby area (NSSA). NSSA is like a stub or totally stub area, but allows an ASBR within the area. External routes are advertised as type 7 routes and sent to the ABR, which converts them to traditional type 5 external routes. The NSSA area allows the network to adapt to your company's physical topology and has advantages in terms of filtering and summarization of external routes. An NSSA area is identified with the **area nssa** command:

```
Router(config-router)# area 7 nssa [no-summary]
```

IS-IS

Intermediate System-to-Intermediate System (IS-IS) is a link state routing protocol that is part of the OSI family of protocols. Integrated IS-IS can carry IP network information, but does not use IP as its transport protocol. It uses CLNS and CLNP to deliver its updates. IS-IS is a classless interior gateway protocol that uses router resources efficiently, and scales to very large networks, such as large Internet service providers (ISPs).

IS-IS Acronyms

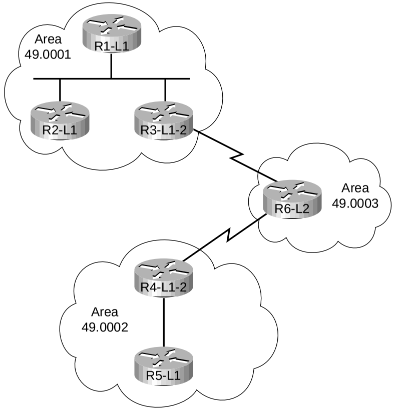
Term	Acronym	Description
Intermediate System	IS	The OSI name for a router.
End system	ES	A host, such as a computer.
Connectionless Network Services	CNLS	An OSI data delivery service that provides best-effort delivery.
Connectionless Network Protocol	CLNP	The OSI protocol used to provide the connectionless services.
End system hello	ESH	Sent by hosts to routers.
Intermediate System hello	ISH	Sent by routers to hosts.
IS to IS hello	IIH	Hellos exchanged between routers. Separate level 1 and level 2 IIHs exist.
Type Length Value	TLV	Fields in the IS-IS updates that contain IP subnet, authentication, and end system information.
Network Service Access Point	NSAP	The address of a CLNS device. Addresses are assigned per device, not per interface as with IP.

Term	Acronym	Description
Network Entity Title	NET	A router's NSAP. The last byte of a NET is always zero.
NSAP Selector	NSEL	The last byte of a NSAP address. Identifies the process on the device, such as routing.
Link State PDU	LSP	A routing update.
Subnetwork Point of Attachment	SNPA	Layer 2 identification for a router's interface, such as MAC address or DLCI.
Circuit ID		Identifies a physical interface on the router.
Link State Database	LSDB	
Sequence Number Protocol Data Unit	SNP	An IS-IS packet that is sequenced and must be acknowledged. The sequence number helps a router maintain the most recent link state information.
Complete Sequence Number PDU	CSNP	A summary of a router's complete Link State Database.
Partial Sequence Number PDU	PSNP	Used to acknowledge receipt of a CSNP and to request more information about a network contained in a CSNP.
Partial route calculation	PRC	Used to determine end system and IP subnet reachability.

Types of IS-IS Routers

An IS-IS network is divided into areas. Within an area, routers can be one of three types:

- **Level 1 (L1) router**—Routes to networks only within the local area (intra-area routing). Uses a default route to the nearest Level 2 router for traffic bound outside the area. Routing is based on System ID. Keeps one LSDB for the local area. (R1, R2, and R5 in this figure.)



- **Level 2 (L2) router**—Routes to networks in other areas (interarea routing). The routing is based on area ID. Keeps one LSDB for routing to other areas. (R6 in this figure.)
- **Level 1-2 (L1-2) router**—Acts as a gateway into and out of an area. Does Level 1 routing within the area and Level 2 routing between areas. Keeps two LSDB, one for the local area and one for interarea routing. (R3 and R4 in this figure.)

The IS-IS backbone is not a specific area, as in OSPF, but an unbroken chain of routers doing Level 2 routing. R3, R6, and R4 are the backbone in this figure.

NSAP Address Structure

Area ID—1 to 13 bytes long	**System ID**—Must be exactly 6 bytes long	**NSEL**—1 byte

In Cisco implementation of integrated IS-IS, NSAP addresses have three parts. A router always has a NSEL of 00. An area ID that begins with 49 designates private area addressing. MAC addresses or IP addresses padded with 0s are often used as system IDs.

Adjacency Formation in IS-IS

IS-IS routers form adjacencies based on the level of IS routing they are doing and their area number. This is a CLNS adjacency and can be formed even if IP addresses don't match.

- Level 1 routers form adjacencies only with L1 and L1-2 devices in their own area. (In the previous figure, R1 with R2 and R3.)
- Level 2 routers form adjacencies only with Level 2 capable devices (either L2 or L1-2 routers). These can be in the local area or in other areas. (In the previous figure, R6 with R3 and R4
- Level 1-2 routers form Level 1 adjacencies with L1 routers in their own area, and Level 2 adjacencies with routers in other areas. (In the previous figure, R4 has a L1 adjacency with R5 and a L2 adjacency with R6.)

IS-IS in a Broadcast, Multiaccess Network

On a network such as Ethernet, IS-IS routers elect a Designated Intermediate System (DIS). The DIS is elected based on priority, with MAC address as the tiebreaker. Routers form adjacencies with all routers on the LAN as well as the DIS. The DIS creates a *pseudonode* to represent the network and sends out an advertisement to represent the LAN. All routers advertise only an adjacency to the pseudonode. If the DIS fails, another is elected; no backup DIS exists. The DIS sends Hellos every 3.3 seconds; other routers send them every 10 seconds. The DIS also multicasts a CSNP every 10 seconds.

IS-IS on a Point-to-Point Link

No DIS exists on a point-to-point link. When an adjacency is first formed over the link, the routers exchange CSNPs. If one of the routers needs more information about a specific network, it sends a PSNP requesting that. After the initial exchange, LSPs are sent to describe link changes, and they are acknowledged with PSNPs.

IS-IS Configuration Tasks

- Enable IS-IS on the router.
- Configure each router's NET.
- Enable IS-IS on the router's interfaces.

Basic IS-IS Commands

Command	Description
router isis	Enables IS-IS on the router
net	Assigns the router's NSAP address
ip router isis	Enables IS-IS on an interface
is-type	Sets the IS level for the whole router
isis circuit-type	Sets the IS level for an interface
isis metric	Changes the metric for an interface
summary-address	Summarizes IP networks for IS-IS
show isis topology	Displays the topology database and least cost paths
show clns route	Displays the L2 routing table
show isis route	Displays the L1 routing table—requires CLNS routing enabled
show clns protocol	Displays the router's IS type, system ID, area ID, interfaces running IS-IS, and any redistribution
show clns neighbors	Displays the adjacent neighbors and their IS level
show clns interface	Displays IS-IS details for each interface, such as circuit type, metric, and priority
show ip protocols	Displays the integrated IS-IS settings

Tuning IS-IS

You need to do three basic types of tuning to IS-IS routers:

- Set the IS level.
- Set the circuit type on L1-2 routers.
- Summarize addresses.

Setting the IS Level

Cisco routers are L1-2 by default. If the router is completely an internal area router, set the IS level to L1. If the router routes only to other areas and has no internal area interfaces, set the IS level to L2. If the router has both internal and external area interfaces, leave the IS level at L1-2.

Setting the Circuit Type

On L1-2 routers, all interfaces send out both L1 and L2 hellos, trying to establish both types of adjacencies. This can waste bandwidth. If only an L1 router is attached to an interface, then change the circuit type for that interface to L1, so that only L1 hellos are sent. If there is only a L2 router attached to an interface, change the circuit type for that interface to L2.

Summarizing Addresses

Although IS-IS does CLNS routing, it can summarize the IP addresses that it carries. Summarized routes can be designated as Level 1, Level 2, or Level 1-2 routes. The default is Level 2.

Optimizing Routing

Migrating IP Addressing

Most networks operate more efficiently when using a modern routing protocol. When migrating to a new routing protocol, you might also need to change IP address schemes. You often change IP addressing when moving from a FLSM protocol to a VLSM one, or to create route summary points within the network. Here are some areas to consider when planning this:

- **Host addresses**—Change or add DHCP.
- **NAT**—Translate the new addresses.
- **DNS servers**—If server have new addresses, new DNS mappings made need to be configured.
- **Access lists and firewalls**—Update traffic filters to work for the new addresses.
- **Routing**—Update routers to route for the new networks.
- **Secondary addressing**—You might need to have both old and new addresses on the router interfaces during transition.
- **Timing of the transition**—Who gets converted when. Considers the day of the week and time of day.
- **Transition strategy**—Which parts of the network are changed first, second, etc. How to avoid disrupting user traffic during the transition.

Migrating Routing Protocols

The steps in migrating to a new routing protocol are as follows:

Step 1 Decide on a timeline for the migration.

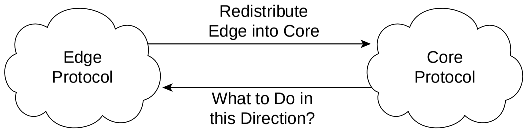
Step 2 Identify boundary routers that are to run both protocols.

Step 3 Decide which protocol is to be the core and which is to be the edge protocol.

Step 4 Decide where and in what direction to redistribute routes.

Step 5 Test the plan in a lab.

Step 6 Back up all device configurations before changing them.



Route redistribution can cause problems because of route feedback, incompatible metrics, and inconsistent convergence times. The safest way to run multiple protocols is to redistribute the edge protocol into the core, and send a default route or use static routing back to the edge. The next best way is to redistribute both ways and filter to avoid route feedback distance, or change the administrative distance of redistributed routes.

Each protocol has some unique characteristics when redistributing, as shown in the following table.

Route Redistribution Characteristics

RIP	Metric must be set except when redistributing static or connected routes, which have a metric of 1.
OSPF	Default metric is 20. Can specify the metric type; the default is E2. Must use **subnets** keyword or only classful networks are redistributed.
EIGRP	Metric must be set, except when redistributing static or connected routes, which get their metric from the interface. Metric value is "bandwidth, delay, reliability, load, MTU." Redistributed routes have a higher administrative distance than internal ones.
IS-IS	Default metric is 0. Can specify route level; default is L2. Can choose to redistribute only external or internal routes into IS-IS from OSPF and into OSPF from IS-IS.
Static/Connected	Only routes that are in the routing table and learned via the specified protocol are redistributed. To include local networks, you must redistribute connected interfaces. You can also redistribute static routes into a dynamic protocol.

Seed Metric

Redistribution involves configuring a routing protocol to advertise routes learned by another routing process. Protocols use incompatible metrics, so the redistributed routes must be assigned a new metric compatible with the new protocol. Normally, metrics are based on an interface value, such as bandwidth, but no interface for a redistributed route exists. A route's original metric is called its *seed metric*.

Set the metric for all redistributed routes with the **default-metric** [*metric*] command. To set the metric for specific routes, either use the **metric** keyword when redistributing or use the **route-map** keyword to link a route map to the redistribution.

Configuring Route Redistribution

You can redistribute between protocols that use the same protocol stack, e.g., IP protocols cannot advertise IPX routes. Configuring redistribution is simple; issue this command under the routing process that is to receive the new routes:

```
redistribute [route source]
```

Tools for Controlling/Preventing Routing Updates

- Passive interface
- Default and/or static routes
- Distribute list
- Route map
- Change administrative distance

Passive Interface

Passive interface prevents RIP and IGRP from sending updates out an interface. It prevents other routing protocols from sending hellos out of an interface; thus, they don't discover neighbors or form an adjacency out that interface. RIP, IGRP, and older versions of EIGRP require a classful network in the **network** statement. To avoid running the protocol all interfaces within that network, use the **passive-interface** command and specify an interface that doesn't run the protocol. To turn off the protocol on all interfaces, use **passive-interface default**, then **no passive-interface** for the ones that should run the protocol.

Distribute Lists

A distribute list allows you to filter routing updates through an access list. Configure an access list and then link that access list to the routing process with the **distribute-list** command.

The **distribute-list** command has two options:

- **distribute-list** [*ACL*] **in**—Filters updates as they come in an interface. For OSPF, this controls routes placed in the routing table but not the database. For other protocols, this controls the routes the protocol knows about.
- **distribute-list** [*ACL*] **out**—Filters updates going out of an interface and also updates being redistributed in from another protocol.

Route Maps

Route maps *match* conditions, and then *set* options for traffic that matches. Each statement has a sequence number, statements are read from the lowest number to highest, and the router stops reading when it gets a match. The sequence number can be used to insert or delete statements. Here are some uses for route maps:

- **Filtering redistributed routes**—Use the **route-map** keyword in the **redistribute** command.
- **NAT**—To specify the private addresses to be translated.
- **Policy-based routing**—To specify which traffic should be policy routed, based on very granular controls.
- **BGP policy**—To control routing updates and to manipulation path attributes.

Route Map Syntax

Route maps are created with the command:

route-map [***name***] **permit** | **deny** [***sequence no.***]

Each statement in a route map begins this same way, with the same route map name but different sequence numbers, and with *match* and/or *set* conditions below it. *Permit* means that any traffic matching the *match* conditions is used. *Deny* means that any traffic matching the *match* conditions is not used.

Match and Set Conditions

Each route map statement can have from none to multiple **match** and **set** lines. If no **match** line exists, the statement matches anything, similar to a "permit any" in an access list. If there is no **set** line, the matching traffic is either permitted or denied, with no other conditions being set.

Multiple **match** conditions on the same line use a logical OR. For example, the router interprets **match a b c** as "match a or b or c." Multiple **match** conditions on different lines use a logical AND. For example, the router interprets the following as "match a and b and c":

```
match a
match b
match c
```

In route redistribution, here are some common things to match:

- **ip address**—Refers the router to an access list that permits or denies networks.
- **ip next-hop**—Refers the router to an access list that permits or denies next-hop IP addresses.
- **ip route-source**—Refers the router to an access list that permits or denies advertising router IP addresses.
- **metric**—Permits or denies routes with the specified metric from being redistributed.
- **route-type**—Permits or denies redistribution of the route type listed, such as internal or external.
- **tag**—Routes can be labeled (tagged) with a number, and route maps can look for that number.

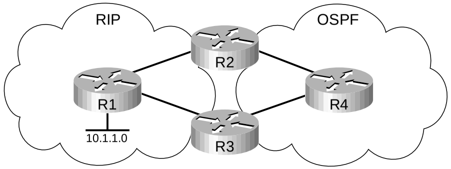
In route redistribution, some common things to set are

- **metric**—Sets the metric for redistributed routes
- **metric-type**—Sets the type, such as E1 for OSPF
- **tag**—Tags a route with a number that can be matched on later by other route maps
- **level**—For IS-IS, sets the IS level for this route

Route Map Example

```
route-map Demo permit 10
match ip address 23
set metric 550
route-map Demo permit 20
```

Manipulating Administrative Distance



When a router receives routes to the same destination network from more than one routing process, it decides which to put in the routing table by looking at the administrative distance (AD) value assigned to the routing process. The route with the lowest AD is chosen. AD can be changed for all routes of a process, or only for specific routes within a process. The command is as follows:

distance {**admin distance**} [**address mask** [**ACL**]]

Using the **address/mask** keywords in the command changes the AD of routes learned from the neighbor with that IP address. Specifying an access list number or name changes the AD only on networks permitted in the ACL. EIGRP and BGP have different AD values for internal and external routes, so you have to list those separately when using the command with those protocols.

In the previous figure, look at the path to the 10.1.1.0 network. Redistribute RIP into OSPF on R2. These routes inherit OSPF's AD when they are advertised to R4. R4 then advertises them to R3 as OSPF routes. We have route feedback. R3 now knows about the 10.1.1.0 network from two routing processes: RIP, with an AD of 120, and OSPF, with an AD of 110. The path for RIP is through R1, the shortest path. The path for OSPF is through R4 and R2, then to R1—a much longer path. But the router puts the OSPF path in the routing table, based on AD.

To prevent this, increase the AD of the redistributed RIP routes when OSPF advertises them. Note—this doesn't change all OSPF routes, just the ones learned from RIP. The commands, given on R2, are

```
access-list 10 permit 10.1.1.0
router ospf  1
     redistribute rip subnets
     distance 125 0.0.0.0 255.255.255.255 10
```

Now R3 hears about the 10.1.1.0 network from RIP with an AD of 120, and from OSPF with an AD of 125. The RIP route is, therefore, put into the routing table.

Policy-Based Routing

Routers normally route traffic based on destination network. Policy-based routing (PBR) overrides this and causes them to choose a path based on the following:

- Port number of the application used
- Protocol number
- Packet size

PBR lets you load share based on characteristics of the traffic, rather than bandwidth of the links, and you also can set quality of service (QoS) values. It is applied to packets as they enter the router, and is configured by using route maps.

PBR Basics

1. Configure a route map to match traffic against either a standard or extended access list, or packet length.
2. If traffic matches the conditions and the route map statement is a *permit*, policy route the traffic.
3. If traffic matches the conditions and the route map statement is a *deny*, do normal routing.
4. If all the statements in the route map have been checked and no match exists, route the traffic normally (based on destination).

Configuring PBR

You can set the following options in the route map:

- **ip next-hop**—The IP address of the next hop router. The router checks the routing table to make sure this next hop is reachable.
- **interface**—The outbound interface for the traffic. An explicit route must already be in the routing table for the destination network, but this statement overrides it.
- **ip default next-hop**—The IP address of a next hop router to send traffic if there is no explicit route for the destination network in the routing table.
- **default interface**—The outbound interface for traffic for which there is no explicit route in the routing table.
- **ip tos**—Sets the type of service bits in the IP header; used for QoS purposes.
- **ip precedence**—Sets the precedence value in the IP header; used for QoS purposes.

The policy must be applied to the incoming interface with the command **ip policy route-map <*route map name*>**. Verify the policy with the following commands:

- **show ip policy**
- **ping**
- **traceroute**
- **show route-map <*name*>**
- **debug ip policy**

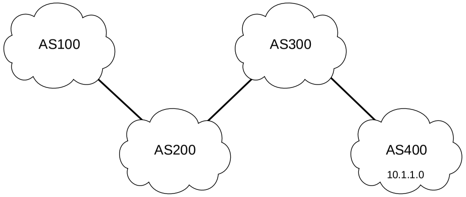
BGP

BGP Basics

- BGP stands for Border Gateway Protocol.
- BGP uses the concept of autonomous systems. An autonomous system is a group of networks under a common administration.
- Autonomous systems run Interior Gateway Protocols (IGPs) within the system. They run an Exterior Gateway Protocols (EGP) between them.
- BGP version 4 is the only EGP currently in use.

- BGP neighbors are called *peers* and must be statically configured.
- Uses TCP port 179.
- BGP is a path-vector protocol. Its route to a network consists of a list of autonomous systems on the path to that network.
- BGP's loop prevention mechanism is autonomous system number. When an update about a network leaves an autonomous system, that autonomous system's number is prepended to the list of autonomous systems that have handled that update. When an autonomous system receives an update, it examines the autonomous system list. If it finds its own autonomous number in that list, the update is discarded.

Example: In the following figure, BGP routers in AS100 see network 10.1.1.0 as having an autonomous system path of 200 300 400.



BGP Databases

- **Neighbor database**—List of all configured BGP neighbors. To view, type **show ip bgp summary**.
- **BGP database, or RIB (Routing Information Base)**—List of networks known by BGP, along with their paths and attributes. To view, type "**show ip bgp**".
- **Routing table**—List of the paths to each network used by the router, and the next hop for each network. To view, type **show ip route**.

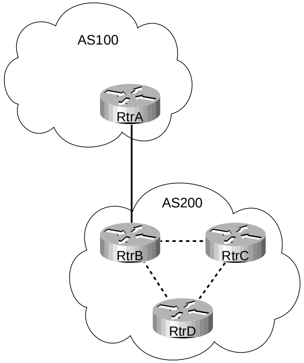
BGP Message Types

- **Open message**—After a neighbor is configured, BGP sends an open message to try to establish a peering with that neighbor. Includes information such as autonomous system number, router ID, and hold time.
- **Update message**—Message used to transfer routing information between peers.
- **Keepalive message**—BGP peers exchange keepalive messages every 60 seconds by default. These keep the peering session active.
- **Notification message**—If a problem occurs that causes the BGP peer to be ended, a notification message is sent to the BGP neighbor, and the connection is closed.

Internal and External BGP

Internal BGP (iBGP) is a BGP peering between routers in the same autonomous system. External BGP (eBGP) is a BGP peering between routers in different autonomous systems. BGP treats updates from internal peers differently than updates from external peers.

In this figure, routers A and B are eBGP peers. Routers B, C and D are iBGP peers.



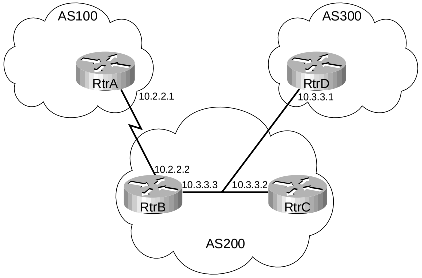
BGP Split Horizon Rule

Routes learned from iBGP neighbors are not forwarded to other iBGP neighbors. BGP assumes that all internal BGP routers are fully meshed, so if one internal router has gotten an update from an iBGP peer, all internal routers should have received it. Without BGP split horizon, routing loops and black holes could be introduced within an autonomous system.

For example, in the previous figure, if RtrB receives an update from RtrA, it forwards the update to RtrC and RtrD. But C and D do not send it back to B, or to each other.

iBGP Next Hop

When a BGP router receives an update from an eBGP neighbor, it must pass that update to its iBGP neighbors without changing the next-hop attribute. The next-hop IP address is the IP address of an edge router belonging to the next-hop autonomous system. Therefore, iBGP routers must have a route to the network connecting their autonomous system to that edge router. For example, in the figure below, RtrA sends an update to RtrB, listing a next hop of 10.2.2.1—it's serial interface. When RtrB forwards that update to RtrC, the next-hop IP address will still be 10.2.2.1. RtrC needs to have a route to the 10.2.2.0 network to have a valid next hop.



BGP Next Hop on a Multiaccess Network

On a multiaccess network, BGP can adjust the next-hop attribute to avoid an extra hop. In the previous figure, RtrC and RtrD are eBGP peers, and RtrC is an iBGP peer with RtrB. When C sends an update to D about network 10.2.2.0, it normally gives its interface IP address as the next hop for D to use. But since B, C, and D are all on the same multiaccess network, it adds an extra hop for D to send traffic to C, and C to send it on to B. So RtrC advertises a next hop of 10.3.3.3 (RtrB's interface) for the 10.2.2.0 network. To change this behavior, use the **neighbor** [*ip address*] **next-hop-self** command.

BGP Synchronization Rule

When a BGP router receives information about a network from an iBGP neighbor, it does not use that information until a matching route is learned via an IGP or static route. It also does not advertise that route to an eBGP neighbor. In the preceding figure, if RtrB advertises a route to RtrC, C does not submit it to the routing table or advertise it to RtrD unless it also learns the route from some other IGP source. It is usually safe to turn off synchronization when all routers in the autonomous system are running BGP. To turn it off, use the command **no synchronization** under BGP router config mode.

Basic BGP Commands

Command	Description
router bgp [*AS number*]	Starts the BGP routing process on the router.
neighbor [*ip address*] **remote-as** [*AS number*]	Sets up a peering between BGP routers.
neighbor [*peer group name*] **peer-group**	Creates a peer group, which you can then put neighbors in.
neighbor [*ip address*] **peer-group** [*peer group name*]	Assigns a neighbor to a peer group.
neighbor [*ip address*] **next-hop-self**	Configures a router to advertise its connected interface as the next hop for all routes to this neighbor.
neighbor [*ip address*] **update-source loopback 0**	Configures a router to use the IP address of its loopback 0 interface as the source for its advertisements to this neighbor.
no synchronization	Turns off BGP synchronization.
network [*prefix*> [**mask** [*subnet mask*]]	Initiates the advertisement of a network in BGP.
no auto-summary	Turns off automatic summarization of routes to the classful network boundary.

The BGP network Command

In most IGPs, the **network** command starts the routing process on an interface. In BGP, the command tells the router to originate an advertisement for that network. The network does not have to be connected to the router; it just has to be in the routing table. In theory, it could even be a network in a different autonomous system (not usually recommended). When advertising a network, BGP assumes you are using the default classful subnet mask. If you want to advertise a subnet, you must use the optional keyword **mask** and specify the subnet mask to use.

BGP Peering

BGP assumes that external neighbors are directly connected and that they are peering with the IP address of the directly connected interface of their neighbor. If not, you must tell BGP to look more than one hop away for its neighbor, with the **neighbor** [*ip address*] **ebgp-multihop** [*no. of hops*] command. You might use this command if you are peering with loopback interface IP addresses. BGP assumes that internal neighbors might not be directly connected, so this command is not needed with iBGP.

BGP Peering States

- **Idle**—No peering; router is looking for neighbor. **Idle** (**admin**) means that the neighbor relationship has been administratively shut down.
- **Connect**—TCP handshake completed.
- **OpenSent**—An open message was sent to try to establish the peering (also called **Active** state).
- **OpenConfirm**—Router has received a reply to the open message.
- **Established**—Routers have a BGP peering session. This is the desired state.

Summarizing BGP Routes

You can perform route summarization in BGP in several ways:

- Leave auto-summarization on and allow the router to advertise the default classful route.
- Use the **network** command along with a static route pointing to Null0. In the following figure, AS200 has the entire 10.2.0.0 subnet. First, configure a static route for the summary, pointing to Null0 (remember that a route must be in the routing table before BGP originates an advertisement for that network):

  ```
  ip route 10.2.0.0 255.255.0.0 null0
  ```

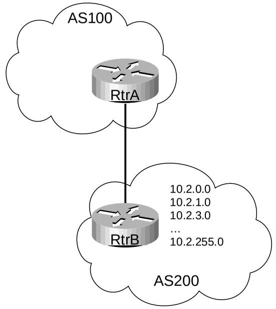
  Then, put a network statement for the summary network into BGP:

  ```
  network 10.2.0.0 mask 255.255.0.0
  ```

- Use the **aggregate-address** command—the preferred method. At least one more specific network of the summarized routes must be in the BGP table, either from the **network** command, or via an advertisement from another router.

In the following figure, RtrB uses the following command to summarize its subnets:

```
aggregate-address 10.2.0.0 255.255.0.0
```



BGP advertises the summary route and the more specific routes. To suppress advertisement of the specific routes, add the summary-only keyword to the preceding command. If the routes come from other autonomous systems, you should add the as-set keyword. This tells BGP to advertise a list of the autonomous system numbers associated with the specific routes. BGP's default is to advertise the route with only the autonomous system number of the summarizing router. The aggregate-address command causes BGP to automatically create a route for the summary, pointing to Null0, to prevent loop formation.

BGP Attributes

BGP chooses a route to network based on the attributes of its path. Four categories of attributes exist:

- **Well-known mandatory**—Must be recognized by all BGP routers, present in all BGP updates, and passed on to other BGP routers. Example: autonomous system path, origin, next hop.
- **Well-known discretionary**—Must be recognized by all BGP routers and passed on to other BGP routers, but need not be present in an update. Example: Local preference.
- **Optional transitive**—Might or might not be recognized by a BGP router, but is passed on to other BGP routers. If not recognized, it is marked as partial. Example: aggregator, community.
- **Optional nontransitive**—Might or might not be recognized by a BGP router and is not passed on to other routers. Example: Multi-Exit Discriminator (MED), originator ID.

Autonomous system path	An ordered list of all the autonomous systems through which this update has passed.
Origin	How BGP learned of this network. i = by **network** command, e = from EGP, ? = redistributed from other source.
Next hop	The IP address of the next-hop router.
Local preference	A value telling iBGP peers which path to select for traffic leaving the AS.
Weight	Cisco proprietary, to tell a router which of multiple local paths to select for traffic leaving the AS. Only has local significant.
Multi-Exit Discriminator (MED)	Suggests to a neighbor autonomous system which of multiple paths to select for traffic bound for your autonomous system.

Applying BGP Policies

These attributes are usually manipulated using route maps. You can set a default local preference by using the command **bgp default local-preference** and a default MED for redistributed routes with the **default-metric** command under the BGP routing process.

But by using route maps, you can change attributes for certain neighbors only, or for certain routes only.

When attributes are changed, you must tell BGP to apply the changes. Either clear the BGP session (**clear ip bgp ***) or do a soft reset (**clear ip bgp * soft in | out**).

BGP Path Selection Criteria

BGP tries to narrow its path selection down to one best path; it does not load balance by default. To do so, it examines the path attributes in the following order:

1. Is the route synchronized (if iBGP) with a valid next hop and no autonomous system loops?
2. Choose the route with the highest weight.
3. If weight is not set, choose the route with the highest local preference.
4. Choose a route that was originated locally over one that was advertised to you.
5. Choose the path with the shortest autonomous system path.
6. Choose the path with the lowest origin code (i is lowest, e is next, ? is last).
7. Choose the route with the lowest MED, if the same autonomous system advertises the possible routes.
8. Choose an eBGP route over an iBGP route.
9. Choose the route through the nearest IGP neighbor.
10. Choose the oldest route.
11. Choose a path through the neighbor with the lowest router ID.

To enable BGP to load balance over more than one path, you must enter the command **maximum-paths** [***no. of paths***]. BGP can load balance over a maximum of 6 paths.

Multihoming

Multihoming means connecting to more than one ISP at the same time.

Multihoming is done for redundancy and backup in case one ISP fails—and for better performance—if one ISP provides a better path to often used networks. Three ways exist to receive routes from each ISP:

- Default routes from each provider. Low use of bandwidth and router resources, IGP metric determines path chosen for routes outside the autonomous system.
- Default routes plus some more specific routes. Medium use of bandwidth and router resources. Can manipulate path for specific routes, IGP metric chooses path for default routes.
- All routes from all providers. Highest use of bandwidth and router resources. Typically done only by ISPs. Path selection can be controlled via BGP policy routing tools.