



The Bryant Advantage

WORLD-CLASS CISCO® TRAINING

The Bryant Advantage
Cisco Certified Network Associate®
Course Guide and Lab Workbook
PDF Version

Chris Bryant, CCIE™ # 12933

Copyright Information:

Cisco®, Cisco® Systems, CCIE™, and Cisco Certified Internetwork Expert are registered trademarks of Cisco® Systems, Inc., and/or its affiliates in the U.S. and certain countries.

All other products and company names are the trademarks, registered trademarks, and service marks of the respective owners. Throughout this Course Guide, The Bryant Advantage has used its best efforts to distinguish proprietary trademarks from descriptive names by following the capitalization styles used by the manufacturer.

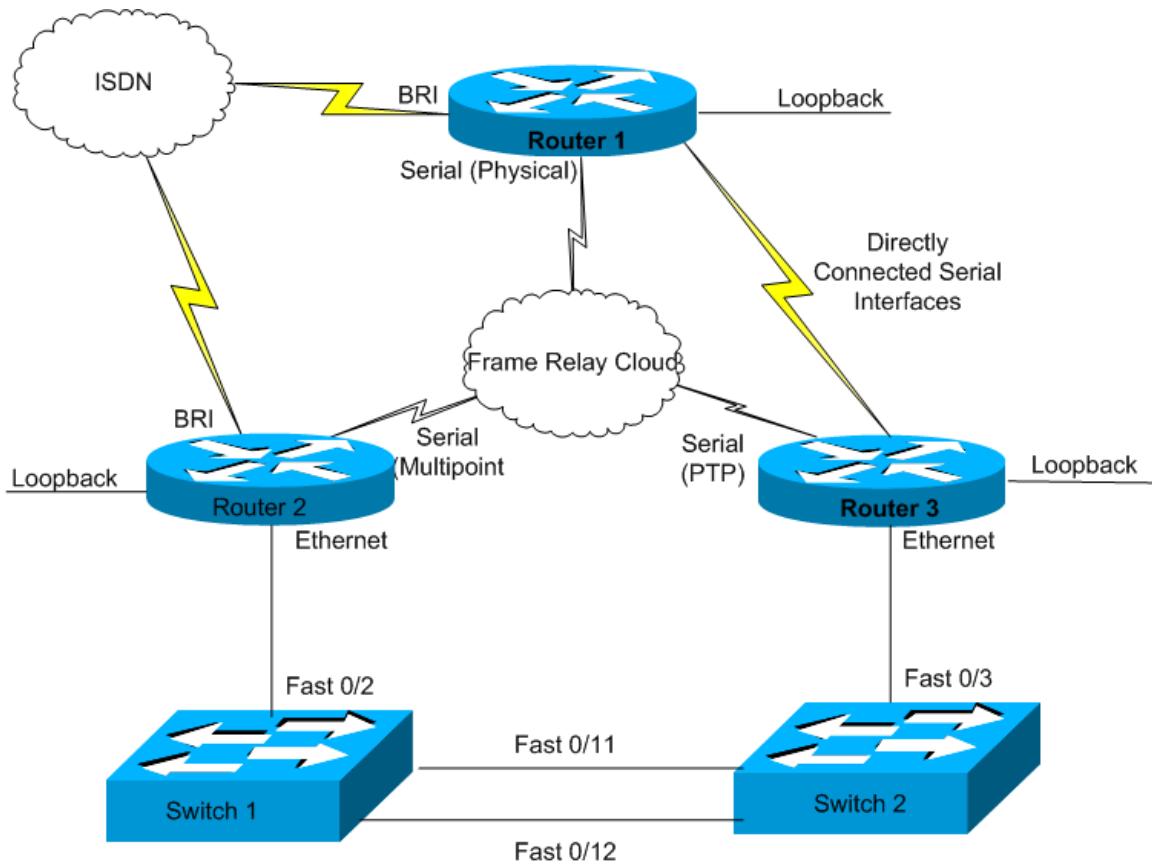
Disclaimer:

This publication, ***The Bryant Advantage CCNA ® Course Guide and Lab Workbook***, is designed and intended to assist candidates in preparation for the ICND exam for the Cisco Certified Network Associate ® certification. All efforts have been made by the author to make this book as accurate and complete as possible, but no guarantee, warranty, or fitness are implied, expressly or implicitly. The enclosed material is presented on an "as is" basis. Neither the author, Bryant Instructional Services, or the parent company assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This Course Guide is an original work by the Author. Any similarities between materials presented in this Study Guide and actual CCNA® exam questions are completely coincidental.

Copyright 2004 © The Bryant Advantage

The Lab Layout:



The labs in this book are run on this network. Students of **The Bryant Advantage CCNA® Accelerated Learning Course** have a minimum of three days of intensive hands-on experience configuring these labs on the above network, and the opportunity to spend this time with Chris Bryant, Cisco Certified Internetwork Expert™ #12933 in preparation for their Cisco exams and their career.

This network can be replicated in a home lab. Besides the three routers shown above, you will also need two 2950 switches, a frame relay switch, and an access server. An ISDN simulator will also be needed.

Purchasers of this Study Guide have the opportunity to rent access to a pod of equipment preconfigured for use with the labs in the Study Guide. The Bryant Advantage is the first organization in the world to offer Cisco rack rentals especially designed and priced for CCNA and

CCNP candidates. Visit www.thebryantadvantage.com today for details on renting your own pod of Cisco routers and switches!

As always, your feedback is welcome. I'm always glad to hear from students and exam candidates at chris@thebryantadvantage.com.

For the most up-to-date Cisco certification news, subscribe to Cisco Certification Central, the #1 Cisco certification newsletter in the world. Every week, you get the latest news on changes and updates on Cisco certification, exam tips, and special offers that are extended ONLY to CCC subscribers. Visit www.thebryantadvantage.com to subscribe today!



The Bryant Advantage

WORLD-CLASS CISCO® TRAINING

Section One: The Internetworking Models

As networks grow, the traffic on them grows as well. A network where all traffic flows throughout the entire network quickly becomes inefficient and overloaded. The process of dividing a network into smaller, more manageable sections is *segmentation*. Cisco switches, routers, and bridges all work to do this in different ways.

Before analyzing how Cisco devices perform this task, it is important to understand the terms **broadcast domain** and **collision domain**.

A **broadcast domain** is the set of hosts (usually end user PCs) and network devices that will receive all broadcasts sent on that segment. A **broadcast** is a message that has no specific destination; every single device on the network will receive it. If too many hosts exist in a single broadcast domain, congestion can occur.

Poorly defined broadcast domains can also lead to a **broadcast storm**, occurring when one device sends a broadcast, and in response, all hosts receiving this broadcast answer with a broadcast of their own. The number of broadcasts continues to rise until they begin to block other network traffic.

Collision domains are found on Ethernet network segments. Ethernet uses a process called Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to communicate across the network. Using CSMA/CD, a node will not send out a packet unless the network is clear of traffic. If two nodes send out packets at the same time, a collision occurs and the packets are lost. Then both nodes wait a random amount of time and retransmit the packets. Any part of the network where there is a possibility that packets from two or more nodes will interfere with each other is considered to be part of the same **collision domain**. A network with a large number of nodes on the same segment will often have a lot of collisions and therefore a large collision domain.

A **router** breaks up a broadcast domain due to one simple rule: *routers do not forward broadcasts*. Routers also break up collision domains, but not by default.

Switches break up collision domains by logically placing hosts into their own, smaller collision domains. Consider an Ethernet network with 100 hosts. Using CSMA/CD, if one host is transmitting, 99 hosts

cannot transmit. Switches can be used to create virtual networks, each containing a smaller amount of hosts. If 10 such networks are created, each containing 10 hosts, one host transmitting data would only affect the other nine users in that virtual network. The chance of collisions decreases as well.

Bridges are also used to break up collision domains. A switch is basically a highly intelligent bridge. Switches also offer many more ports than a bridge.

Repeaters are used to regenerate an electrical signal to allow the signal to travel a greater distance than it would be able to on its own.

Hubs are basically multiport repeaters. Neither device can be used to break up a broadcast domain or a collision domain.

	Breaks Up Broadcast Domain?	Breaks Up Collision Domain ?
Routers	Yes, by default.	Yes, but not by default.
Bridges	No	Yes
Switches	No	Yes
Hubs	No	No
Repeaters	No	No

The OSI Reference Model

The OSI Reference Model is a structured set of guidelines for communications between two end users in a network. It's used in application development and understanding how an internetwork actually works. The OSI model is broken down into seven layers, examined here from the top level down to the bottom.

The Application Layer

The top layer of the OSI model, the Application Layer is the layer that the end user actually interacts with. The Application layer should not be confused with the application itself. The Application layer ensures that the remote communication partner is available, that the needed communication resources exist (a modem, for example), and that both ends of the communication agree on procedures involving data integrity, privacy, and error recovery.

The sender or receiver can also be authenticated at this level.

The Presentation Layer

The Presentation Layer makes sure that the communications the Application Layer will be presenting are in the appropriate format. There are three primary facets that the Presentation Layer is concerned with.

1. Compatibility with the operating system
2. Proper encapsulation of data for network transmission.
3. Data formatting (ascii, binary)

The Session Layer

The Session Layer handles the construction and teardown of the connection between the two end points involved in the communication. Some sessions last just long enough to send a unidirectional message, where other sessions will be of longer duration.

Sessions are related to certain *ports*, a number that is associated with an upper layer application. Port numbers that are static and often-used are referred to as *well-known port numbers*.

The Transport Layer

The Transport Layer provides data flow controls and error checking mechanisms, and the reliable arrival of messages.

At the Transport Layer, there are two methods for transporting data: “connection-oriented”, referring to TCP, and “connectionless”, referring to UDP.

TCP, the connection-oriented packet delivery method, provides several additional services to prevent lost data:

Flow Control

TCP uses a series of acknowledgements to enforce *flow control*. With flow control, when one router receives a packet, it sends an acknowledgement, or “ack”, back to the sender. If the sender does not receive an ack for a segment it sent, the segment will be resent and reassembled in the correct order at the receiver. This prevents the receiver’s buffer from being overburdened, since packets that are not received due to a full buffer are not acknowledged. (A *buffer* is a part of the router’s memory used to hold packets awaiting processing.)

The Three-Way Handshake

One term often heard when discussing TCP is the “three-way handshake”. The connection is actually built before data is transmitted, which is why TCP is referred to as “connection-oriented”, or “reliable”.

When a sender and receiver participate in a TCP connection, here are the steps of the “three-way handshake”:

1. The sender requests synchronization with the receiver.
2. The request is acknowledged, and the rules of the connection are agreed upon. The receiver then requests synchronization with the sender, resulting in a bidirectional (“two-way”) connection.
3. The sender acknowledges the connection agreement and that the connection now exists, and data can now be transferred over the newly constructed connection.

Windowing

If the sender waits for an ack from the receiver before sending another data segment, the transmission is going to be unacceptably slow. For this reason, TCP allows *windowing*. This term refers to data sent during the time after the sender sent a packet, but before it completes processing the acks it received.

The term “size of the window” refers to the number of packets the sender can transmit before it must wait for an ack. If the “size of the window” is three, the sender can transmit three data segments before it must wait for an ack.

User Datagram Protocol (UDP)

UDP is referred to as “connectionless” because there is no handshake before transmission, and the data channel does not exist before the data is sent; it’s just sent. No acknowledgements are sent or expected. UDP makes no guarantee of delivery, only a “best effort”. UDP does not use windowing or flow control.

Port Numbers

Both TCP and UDP use *port numbers* to keep the conversations involving different protocols separate. Many protocols use the same

port number for all their conversations; these protocols are said to have *well-known port numbers*. Other protocols will use a randomly selected port number.

The Network Layer

The Network Layer (often referred to as Layer 3; IP addresses are often referred to as “Layer 3 addresses”) is aware of the address of neighbor nodes in the network, and is responsible for selecting the best route to transmit data between devices that are not local to each other. The Network Layer is the layer at which *routing* takes place.

Packets at this layer are referred to as **data packets** or **routing update packets**. Data packets are just what they sound like; packets containing user data. Protocols that carry data are **routed protocols**, such as IP (Internet Protocol).

Routing Update Packets are just what they sound like, too; packets carrying routes or changes to the routing tables. **Routing protocols** such as RIP, OSPF, and EIGRP send routing update packets.

Routed Protocols vs. Routing Protocols

It's easy to get a little confused about these two terms, but just as easy to keep them straight. Routed Protocols actually *get routed*, like IP is routed.

Routing Protocols such as EIGRP or RIP *do the routing*. They discover and exchange routes that the routed protocols will be taking.

The Data Link Layer

The Data Link Layer delivers *data frames* using the hardware address, or *MAC Address*. The MAC address is a hexadecimal address unique to that particular device. The Data Link layer breaks a segment into frames and encapsulates the frame with a header that contains the source and destination MAC address.

The Data Link Layer is generally referred to as “Layer 2”, and MAC addresses as “Layer 2 addresses”. Error control and notification are performed on frames at this level.

Switches operate at this layer, as do bridges. ("Layer 3 Switches" do exist, but when operating at Layer 3, they're not switching or bridging. They're routing.)

The Physical Layer

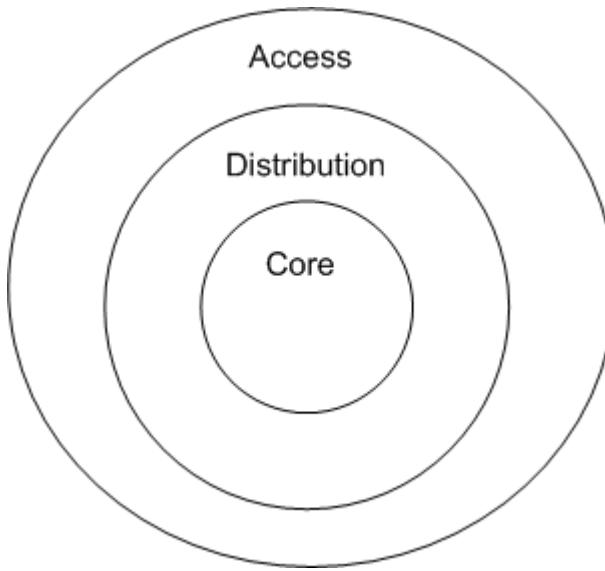
Also referred to as "the bit layer", this layer carries data in units called **bits**, which are in one of two values: zero or one. Hubs and repeaters operate at this layer, often called "Layer One". The electrical and physical specifications for moving data are set at this layer.

	Basic Function	Tasks Performed / Notes
Application	File, print, database, application, and message services.	<p>Identifies remote party, ensures communication with that party.</p> <p>Ensures the needed communication resources exist (modems, etc.).</p> <p>Authentication performed if required.</p>
Presentation	Encrypts and compresses data as needed.	Ensures communications passing through this layer are in the appropriate form for the recipient.
Session	Uses port numbers to keep data from different apps separate.	Manages building and teardown of the connection between the end points.
Transport	Provides reliable or unreliable data delivery.	<p>Uses TCP for reliable, connection-oriented delivery.</p> <p>Uses UDP for unreliable, connectionless delivery.</p> <p>Errors are corrected before retransmission.</p>
Network	Routing.	Logical addressing is provided at this level, referred to as "Layer 3 addresses".
Data Link	Framing.	Access to media via MAC addresses, also called "hardware addresses" and "Layer 2 addresses".
Physical	The actual wiring.	<p>Data is moved in units called "bits";</p> <p>Voltage and wire speed are controlled.</p>

Cisco's Three-Layer Hierarchical Model

Cisco uses another model of its own to describe the design of a network. This model consists of three layers.

The **core layer** is the “center” of this model. Its sole purpose is to switch the network’s traffic as quickly as possible. The **distribution layer** is comprised of routers, and routing is the task this layer handles. The **access layer** is the layer closest to the end users, and controlling their access to network resources is the access layer’s primary function.



Ethernet Data Transmission And Cabling Types

Ethernet is a data transmission method where each host on the network share a link’s bandwidth equally. Ethernet utilizes *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*, meaning that before a host on an Ethernet network will transmit, it listens for a signal on the Ethernet wire first to see if another host is currently transmitting. If the wire is silent, the host will begin transmitting.

If another host attempts to transmit while the first host is still transmitting, the sending host will transmit a *jam signal* informing *all* hosts on the wire to stop transmitting. The nodes that receive that jam signal will not attempt to transmit for a certain period of time.

Ethernet Cabling Types

Devices on an Ethernet network are connected by either a straight-through cable or a crossover cable.

The straight-through cable describes the physical connectivity of the wires inside the cable; Pin 1 is connected to Pin 1, Pin 2 to Pin 2, Pin 3 to Pin 3, and Pin 6 to Pin 6. A router will use a straight-through cable to connect to a hub or switch, as will a host.

The crossover cable has wires that crossover between pins. Pin 1 connects to Pin 6, Pin 2 to Pin 3, Pin 3 to Pin 2, and Pin 6 to Pin 1. A crossover cable is used for the following connections: switch-switch, hub-hub, host-host, switch-hub, or if a router is connected directly to a host.

There is a third type of cable, a **rolled cable**. It's similar to a crossover cable, but eight wires are used instead of four. Pin 1 is connected to Pin 8, Pin 2 to Pin 7, Pin 3 to Pin 6, and Pin 4 to Pin 5. The important factor is that a rolled cable makes it possible for a host to connect directly to the COM port of a router.

Half-Duplex and Full-Duplex Ethernet: Theory vs. Reality

Half-duplex Ethernet connections contain one set of wires, basically meaning that a device can transmit or receive, but cannot do both at the same time. Half-duplex uses the CSMA/CD of transmitting data that was defined at the beginning of this section.

A 10MBPS (megabits per second) half-duplex port sounds like it would allow 10MBPS, but in reality, it will not. The inability to send and transmit data simultaneously inherently slows connection speed down, and the nature of CSMA/CD means that an Ethernet port will be told on occasion that it cannot transmit.

Full-duplex Ethernet contains two set of wires, allowing devices to transmit and receive simultaneously. Since the incoming data is traveling on a different set of wires than the outgoing data, there are no collisions. Theoretically, on a 100 MBPS full-duplex connection, the port should be able to transmit at 100 MBPS and send at 100 MBPS for an overall transmission of 200 MBPS.

The Data Encapsulation Process

As data is passed from the Application Layer to the Physical layer, a layer-specific header is added at each layer. The information contained in the header is specific to the protocol and the layer that added it. These headers are called *protocol data units* (PDUs).

Tracing the process from the top of the OSI model down reveals what PDUs are used at each layer.

1. At the top three layers (Application, Presentation, and Session), the data is referred to as *data*.
2. This data stream is broken up at the Transport layer. A Transport layer-specific header is added to each piece of the broken-up stream, and the result is a *data segment*.
3. The data segments are then sent to the Network layer for Layer 3 routing. A Layer 3-specific header is added, creating a *packet*.
4. The packets are sent to the Data Link layer. The Data Link layer will encapsulate each packet into a *frame*. The frame header identifies the source and destination hardware address, the MAC address.
5. The physical layer will convert these frames into bits that can then be transmitted on the physical wire.

ARP and RARP

Address Resolution Protocol (ARP) is used when a device knows the IP address of a remote device, but not its MAC address.

If the Layer 3 address (the IP address) of the remote device is known, why does the Layer 2 address need to be found? When the data is sent, the destination MAC address must be sent with it.

As network devices learn the MAC addresses of other devices on the network, they build *ARP caches*. This local cache is checked for the proper MAC address, and if the MAC address is not found here, ARP will send out a broadcast containing the known IP address. The remote device with the matching IP address will respond with its MAC address. All other devices will ignore the ARP request.

Reverse Address Resolution Protocol (RARP) is used in the opposite situation; the MAC address is known and the IP address is not. Used when a workstation is *diskless* (that is, the workstation does not have its own hard drive), because the diskless workstation has no way to know its own IP address. It will know its own hardware address, though.

The diskless workstation will send out a packet with its MAC address and a request for its own IP address. A device specifically configured to respond to this request, the *RARP server*, will send a packet back to the diskless workstation containing the desire IP address.

Notes

Internetworking Model Q&A

1. The term “broadcast domain” best describes what group of devices or users?

- A. **The routers that prevent broadcasts from being forwarded.**
- B. **The switches that allow broadcasts to be forwarded.**
- C. **The hosts and network devices that will receive all broadcasts sent on a given network segment.**
- D. **The segment of a network where there is a possibility that all devices will send broadcasts simultaneously.**

ANSWER: C. A broadcast domain is a set of hosts and other network devices that will receive all broadcasts on a given segment.

2. What term describes a situation where devices on a network segment receive a broadcast and respond with broadcasts of their own, resulting in a number of broadcasts blocking other network traffic?

- A. **Switching**
- B. **Routing**
- C. **Multicasting**
- D. **Broadcast Storm**
- E. **Collision Domain**

ANSWER: D. This situation is a broadcast storm.

3. What statement best describes how hosts send packets onto the network when CSMA/CD is in effect?

- A. **The hosts send packets at any time, without listening to the wire, and will resend if there is a problem.**
- B. **The hosts listen to the wire for traffic, and if the line is clear of traffic, packets are sent. Since the host listened for traffic, no retransmission is ever needed.**
- C. **The hosts listen to the wire for traffic, and if the line is clear of traffic, packets are sent. Packets can be retransmitted if necessary.**
- D. **The hosts listen to the wire for traffic, and only send packets if there is existing traffic on the wire. CSMA/CD will sort the packets out if necessary.**

ANSWER: C. Even though the hosts listen for preexisting traffic, and will not send packets unless there is no traffic, two hosts on the segment can still send traffic simultaneously, resulting in a collision. If this happens, both hosts will wait a random amount of time and then will retransmit.

4. How do routers handle broadcasts?

- A. **Routers forward broadcasts by default.**
- B. **Routers forward broadcasts only to local hosts.**
- C. **Routers forward broadcasts if configured with the “ip forward-broadcast” command.**
- D. **Routers do not forward broadcasts.**

ANSWER: Routers do not forward broadcasts. Period. There is no “ip forward-broadcast” command.

5. Which of the following are typically used to break up a collision domain?

- A. **Routers**
- B. **Bridges**
- C. **Switches**
- D. **Hubs**
- E. **Repeaters**

ANSWER: B, C. Bridges and switches are typically used to break up a collision domain. Routers can do so, but generally are not used for this purpose.

Don't let the word "typically" throw you on an exam question. If you see that word, follow your instincts and choose the best answer.

6. What device regenerates an electrical signal, allowing the signal to travel farther than it would be able to without the regeneration?

- A. **Router**
- B. **Switch**
- C. **Hub**
- D. **Repeater**
- E. **Bridge**

ANSWER: D. A repeater regenerates electrical signals.

7. Which of the following four statements best describe the Presentation Layer of the OSI?

- A. Concerned with the operating system
- B. Typically handles user authentication
- C. Ensures the remote communication partner is available
- D. Proper data encapsulation for network transmission
- E. Builds and tears down the end points of the communication
- F. Handles data formatting
- G. Data is referred to at this level as "data"
- H. Data is referred to at this level as "segments"
- I. Data is referred to at this level as "packets"

ANSWER: A, D, F, G. The Presentation layer is concerned with the OS, ensures proper data encapsulation, handles the formatting of data, and data at this level is referred to as just that – data.

8. Which of the following four statements best describe the Application Layer of the OSI?

- A. Ensures the needed communication resources exist, such as modems.
- B. If authentication is needed, it is performed at this level.
- C. Handles data formatting.
- D. Ensures the remote communication partner is available.
- E. Builds and tears down the end points of the communication
- F. Data is referred to at this level as "data"
- G. Data is referred to at this level as "packets"
- H. Data is referred to at this level as "frames".

ANSWER: A, B, D, F. The Application layer, the top layer of the OSI model, ensures that the resources exist to communicate with the remote partner, as well as that the remote partner is available. Authentication is performed at this level, and data is referred to as "data".

9. Which of the following four statements best describe the Session Layer of the OSI?

- A. Handles data formatting.
- B. Handles the building and teardown of the connection.
- C. Uses IP addresses to keep data from different applications separate.
- D. Uses MAC addresses to keep data from different applications separate.
- E. Uses port numbers to keep data from different applications separate.
- F. Referred to as "Layer 5".
- G. Referred to as "Layer 4".
- H. Referred to as "Layer 6".
- I. Data is referred to at this level as "data".
- J. Data is referred to at this level as "frames".
- K. Data is referred to at this level as "packets".

ANSWER: B, E, F, I. The Session Layer, also referred to as Layer 5, handles the building and teardown of the connection. Port numbers are used to keep data separate from other apps' data. And again, data is referred to here as "data".

10. Which of the following five statements are true of the Transport Layer of the OSI model?

- A. Provides data flow controls and error checking mechanisms.
- B. Uses TCP for connectionless data transfer.
- C. Uses TCP for connection-oriented data transfer.
- D. Uses UDP for connectionless data transfer.
- E. Uses UDP for connection-oriented data transfer.
- F. Referred to as "Layer 3".
- G. Referred to as "Layer 2".
- H. Referred to as "Layer 4".
- I. Data is in segments at this layer.
- J. Data is in packets at this layer.
- K. Data is in frames at this layer.

ANSWER: A, C, D, H, I. The Transport layer does provide data flow controls and error checking; TCP provides connection-oriented data transfer, where UDP provides connectionless transfer; the Transport layer is Layer 4, and data is carried in segments at this layer.

11. Which of the following four statements are true of the Network Layer of the OSI model?

- A. Routing takes place at this layer.
- B. Switching takes place at this layer.
- C. Referred to as "Layer 3".
- D. Referred to as "Layer 2".
- E. Referred to as "Layer 4".
- F. Data is carried in segments.
- G. Data is carried in packets.
- H. Data is carried in frames.
- I. This layer is most concerned with IP addresses.
- J. This layer is most concerned with MAC addresses.

ANSWER: A, C, G, I. The Network Layer is Layer 3, and routing takes place at this layer. Layer 3 carried data in packets, and IP addresses are often referred to as "Layer 3 addresses".

12. Which of the following five statements are true of the Data Link layer of the OSI model?

- A. Routing takes place at this layer.
- B. Switching takes place at this layer.
- C. Referred to as "Layer 2".
- D. Referred to as "Layer 3".
- E. Referred to as "Layer 1".
- F. Data is carried in frames.
- G. Data is carried in bits.
- H. This layer concerns itself with MAC addresses, which are expressed in decimal format.
- I. This layer concerns itself with MAC addresses, which are expressed in binary.
- J. This layer concerns itself with MAC addresses, which are expressed in hexadecimal.
- K. Bridges run at Layer 2.
- L. Repeaters run at Layer 2.

ANSWER: B, C, F, J, K. Switching takes place at Layer 2, where data is carried in frames. MAC addresses are expressed in hexadecimal. Bridges are Layer 2 devices.

13. What five statements are true of the Physical Layer of the OSI model?

- A. Data is carried in bits.
- B. Data is carried in bytes.
- C. Hubs run at this level.
- D. Repeaters run at this level.
- E. Bridges run at this level.
- F. The Physical layer is the lowest layer in the OSI model.
- G. The Physical layer is technically located to the side of the Data Link layer in the OSI model, not below it.
- H. The values in the data unit at this layer are zero and one.
- I. The values in the data unit at this layer are carried in hexadecimal, so any value is possible.
- J. The values in the data unit at this layer are carried in binary, so any value is possible.

ANSWER: A, C, D, F, H. Data is carried in bits at the Physical layer; hubs and repeaters are both Layer 1 devices; in the OSI model, the Physical layer is indeed at the bottom; values in bits are zero or one.

14. Cisco has a three-layer model for describing the design of a network. Which of the following three are layers of that model?

- A. Routing
- B. Core
- C. Switching
- D. Physical
- E. Distribution
- F. Access
- G. Wiring Closet

ANSWER: B, E, F. The inner layer is the Core, the middle layer is the Distribution layer, and the outer layer is the Access layer.

15. With Carrier Sense Multiple Access / Collision Detect, a host will listen to an Ethernet segment before sending traffic. If there's no traffic, the host will send traffic. What happens if another host attempts to send traffic while the first host is still sending?
- A. This activity triggers a broadcast storm.
 - B. The first host will send a unicast jam signal directly to the host attempting to send packets, informing only that host to not send traffic.
 - C. The first host will send a jam signal to all hosts on the wire, informing all hosts to not send traffic.
 - D. If half-duplex Ethernet is in use, the traffic damaged by the resulting collisions will be retransmitted; if full-duplex Ethernet is in use, this behavior is desirable, since all hosts can send traffic simultaneously on full-duplex Ethernet.

ANSWER: C. The first host will send a jam signal, but not just to the host attempting to send traffic; the jam signal will go to all hosts on the segment.

16. When connecting a switch to a router, what cable is typically used?
- A. crossover
 - B. straight-through
 - C. rolled
 - D. streamlined

ANSWER: B. Straight-through cables are used to connect a router to a switch or hub.

17. When connecting a switch to another switch, what cable is typically used?
- A. crossover
 - B. straight-through
 - C. rolled
 - D. streamlined

ANSWER: A. A crossover cable is used to connect two switches, two hubs, and to allow a router to be directly connected to a host.

18. You want to connect your laptop directly to a router. What cable type and router port should be used?

- A. Use a rolled cable to connect your laptop straight to an unused Ethernet port on the router.
- B. Use a rolled cable to connect your laptop straight to the COM port on the router.
- C. Use a crossover cable to connect your laptop straight to an unused Ethernet port on the router.
- D. Use a crossover cable to connect your laptop straight to the COM port on the router.

ANSWER: B. Use a rolled cable for this purpose, and connect to the COM port on the router.

19. A workstation knows the Layer 3 address of a destination, but not its Layer 2 address. Which of the following statements are true? (Choose two.)

- A. The workstation knows the destination's IP address, but not its MAC address.
- B. The workstation knows the destination's MAC address, but not its IP address.
- C. The workstation will use ARP to retrieve this Layer 2 address.
- D. The workstation will use IARP to retrieve this Layer 2 address.
- E. The workstation will use RARP to retrieve this Layer 2 address.

ANSWER: A, C. Layer 3 addresses are IP addresses, and Layer 2 addresses are MAC addresses. ARP is used to get a Layer 2 address when the Layer 3 address is known.

20. A workstation knows its own Layer 2 address, but not its own Layer 3 address. Which of the following statements are true?
- A. **The workstation knows its MAC address, but not its IP address.**
 - B. **The workstation knows its hardware address, but not its IP address.**
 - C. **The workstation knows its IP address, but not its MAC address.**
 - D. **The workstation will use ARP to get its own IP address.**
 - E. **The workstation will use IARP to get its own IP address.**
 - F. **The workstation will use RARP to get its own IP address.**
 - G. **The workstation can dynamically map its MAC address to a randomly selected IP address.**

ANSWER: A, B, F. Remember that MAC addresses are also referred to as hardware addresses. The workstation will use RARP (Reverse ARP) to get its IP address from a RARP server.

Remember: ARP is used when the IP address of a remote host is known, but the MAC address is not. RARP is used in the opposite situation – the local MAC address is known, but the IP address is not.

Notes

Section Two: Switching Theory and Configuration

LAN Switching occurs at Layer 2 of the OSI model. Before taking the CCNA exams, you must know how root bridges are elected, how to influence the election, the various stages of STP, how to configure a VLAN, VTP, and an Etherchannel.

If you don't have access to 2950 switches in your home lab, I do offer remote rack access to pods of Cisco equipment from my website, and they each contain two 2950 switches. I do recommend you have some hands-on experience with these technologies before taking the CCNA exams.

*Chris Bryant
CCIE #12933*

Commands used in this chapter and the accompanying labs include:

Show mac-address-table: Displays Layer 2 addresses known by the switch.

Show spanning-tree vlan <VLAN_NUMBER> : Displays STP information of the VLAN, including the root bridge, bridge ID of the switch, and STP timers.

Show vtp status: Displays information regarding the switch's VLAN Trunking Protocol configuration, including the VTP domain name, password, and pruning status.

Show vlan: Displays information regarding VLANs known about by the switch, including ports that are a member of each VLAN.

Vlan database: Command used to enter vlan database mode. VLANs can be configured in this mode, but care must be taken when exiting this mode. Only the commands **exit** and **apply** exit this mode while saving changes.

General Switching Theory

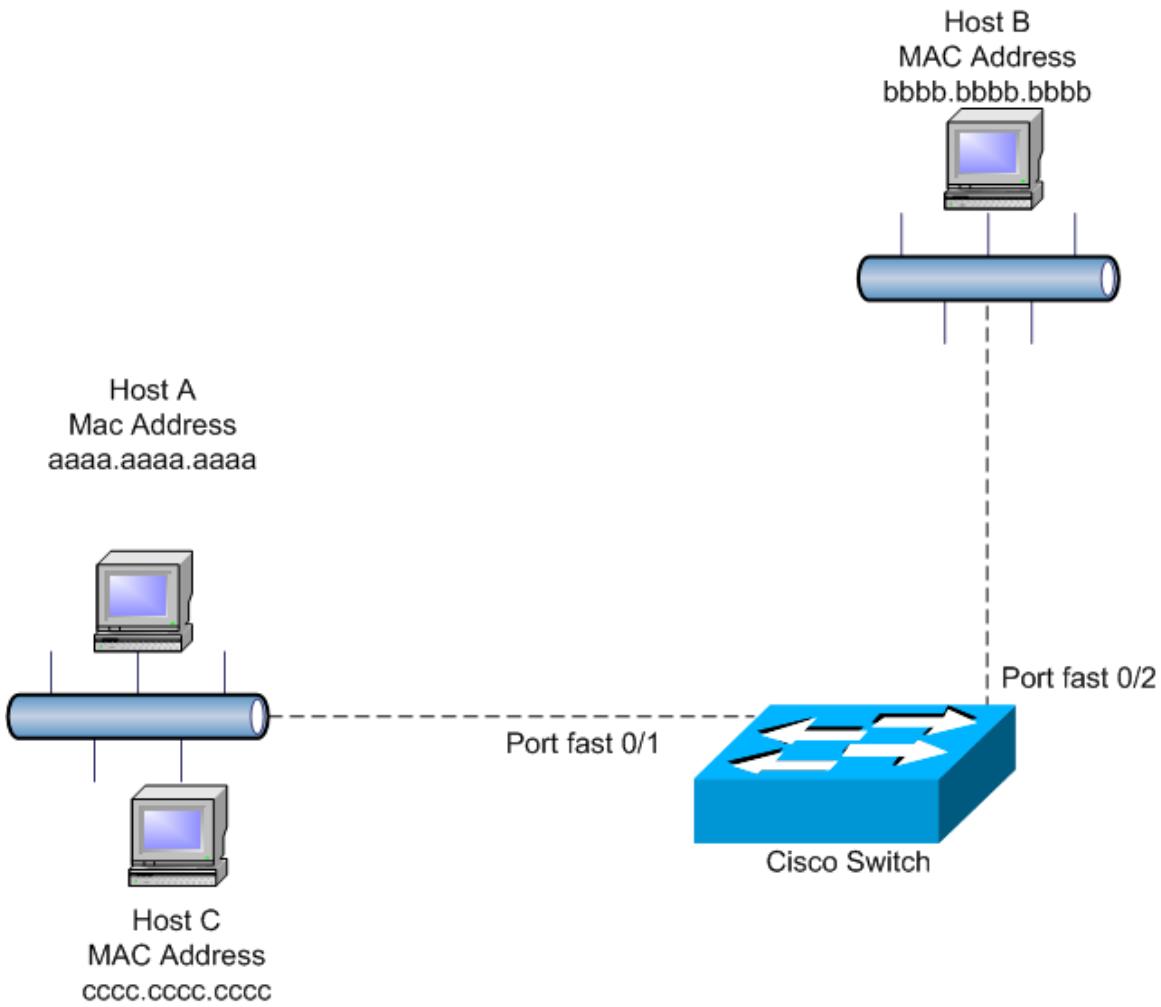
The basic purpose of a switch is to receive a Layer 2 frame, examine the source and destination MAC address, and perform one of three operations:

- A. If the destination is a unicast but the destination address is unknown, forward the frame out all ports except the port on which the frame was received.
- B. If the destination is a unicast and the destination is known, and the exit port for that destination is not the same port on which the frame was received, forward the frame out the correct port and no others. (If the port to reach the destination is the same port on which the source resides, the frame is said to be *filtered* – it is not forwarded out any port.)
- C. If the destination is a broadcast or multicast, forward the frame out all ports except the frame on which it was received.

How The MAC Address Table Is Built

The basic operation of the switch revolves around the MAC address table, sometimes referred to as the “bridging table”. The switch is constantly referring to this table to decide whether to forward or filter a frame.

When a frame is received, the switch does not just look at the destination. The switch also looks at the source to see whether the MAC table has an entry for the source. If it does not, the switch will make an entry in its MAC table noting upon which port that source address can be reached. If there is already an entry for the source MAC address, the source then examines the destination address. This method of examining source addresses is the main fashion in which the MAC table is built.



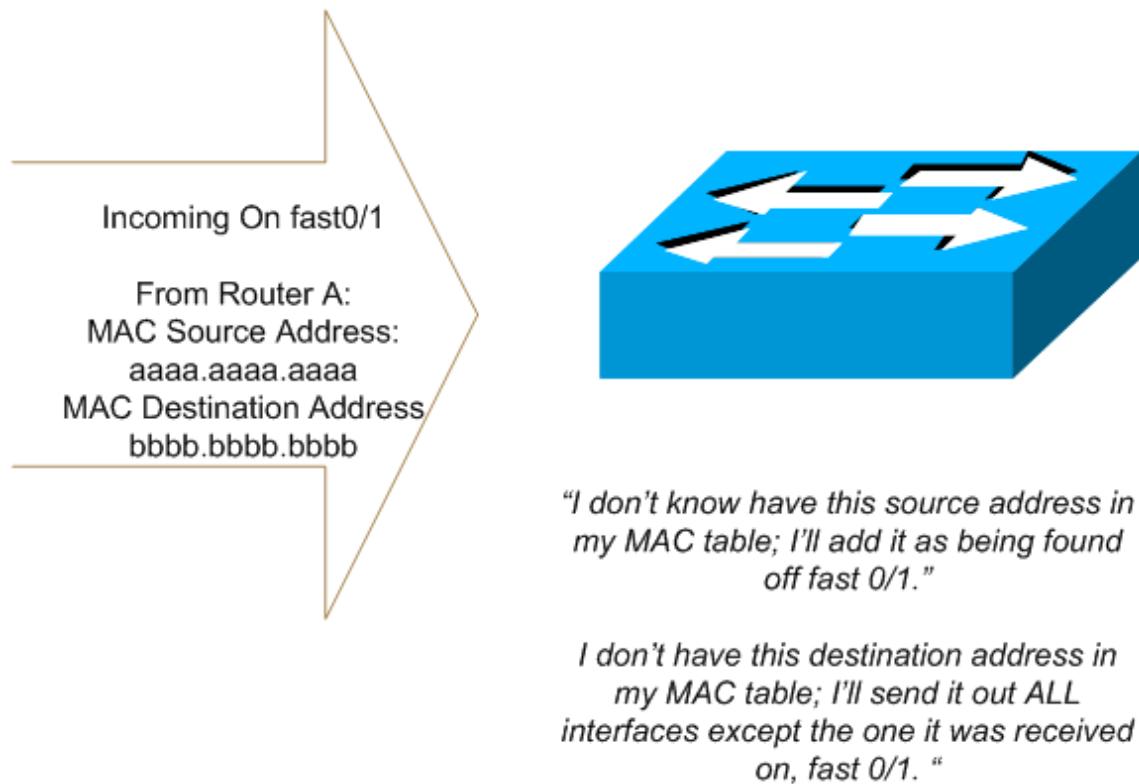
Consider this example: A frame is sent from HostA to HostB. HostA's MAC address is aaaa.aaaa.aaaa and is connected to the switch on port fast0/1, and HostB's MAC is bbbb.bbbb.bbbb and is connected to port fast0/2. The switch was just added to the network and has no entries in its MAC table.

When the switch receives this frame, it will first examine the source address and check its MAC table for an entry for that host. Since the table is empty, there's no entry for HostA. The switch will now make an entry in its MAC table indicating that HostA, with MAC address aaaa.aaaa.aaaa, can be reached on port fast0/1:

Device	MAC Address	Port
HostA	aaaa.aaaa.aaaa	0/1
HostB	?	?

The switch now knows what port to forward frames on that are destined for HostA. The frame is destined for HostB, though, and the switch does not have any idea what port that host can be found on. The switch will now forward that frame out every single port *except the port that the frame was received on*.

The switch does not know off which port the source or destination MAC address can be found.

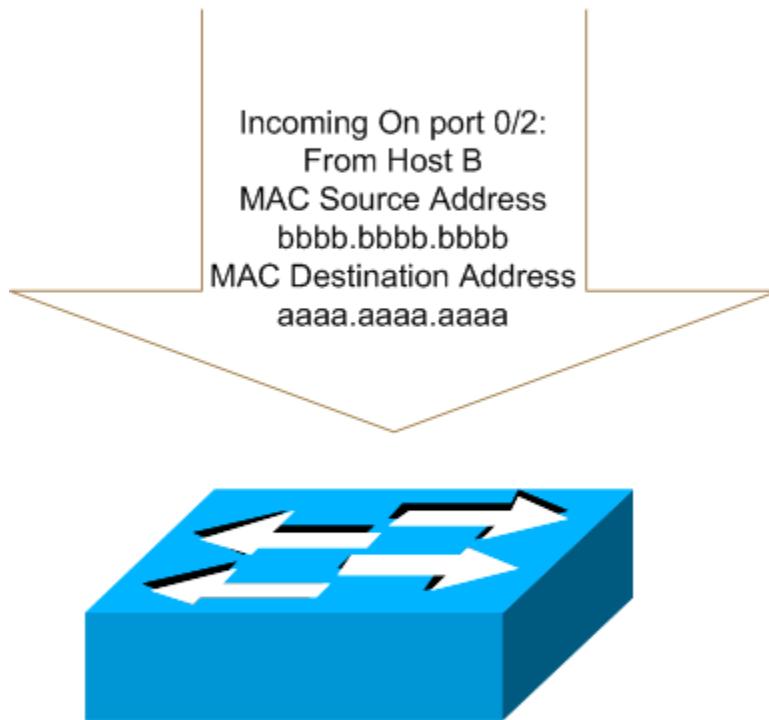


HostB will now respond. The frame comes in on port fast0/2. The switch checks the MAC table for Host B, sees that it does not have an entry for that host, and makes an entry in its MAC table, which now looks like this:

Device	MAC Address	Port
HostA	aaaa.aaaa.aaaa	0/1
HostB	bbbb.bbbb.bbbb	0/2

The switch now looks in its MAC table for an entry for HostA. There is one, so the frame will be forwarded out port 0/1 only. When HostA replies, instead of the flooding of frames that took place the first time frames were sent to HostB, the switch now has an entry for HostB as well, and can send the frames directly to HostB via port fast0/2.

The switch knows off what port the MAC source and MAC destination can be found.



"I know that the device with MAC address aaaa.aaaa.aaaa can be found off port 0/1; I'll only transmit these frames over this port."

The switch also uses the MAC table to determine when to filter frames. The switch will always filter frames when the source and destination can be found off the same port.

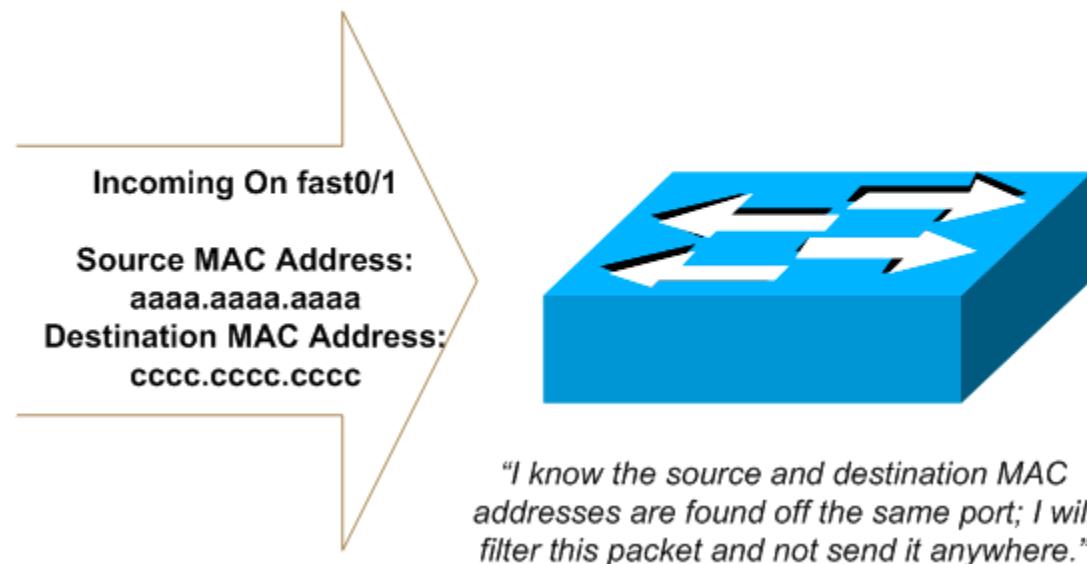
Once HostC begins to send packets to HostB, the switch's MAC table will look like this:

Device	MAC Address	Port
HostA	aaaa.aaaa.aaaa	0/1
HostB	bbbb.bbbb.bbbb	0/2
HostC	cccc.cccc.cccc	0/1

The switch has seen packets come in on port 0/1 with a source address of both aaaa.aaaa.aaaa and cccc.cccc.cccc, so the switch knows that HostA and HostC can both be found off port 0/1.

If HostA then transmits to HostC, or vice versa, the switch will receive these frames. There is no reason for the switch to forward these frames back out the same port, or any other port. The switch will *filter* frames when the source and destination are reachable via the same port.

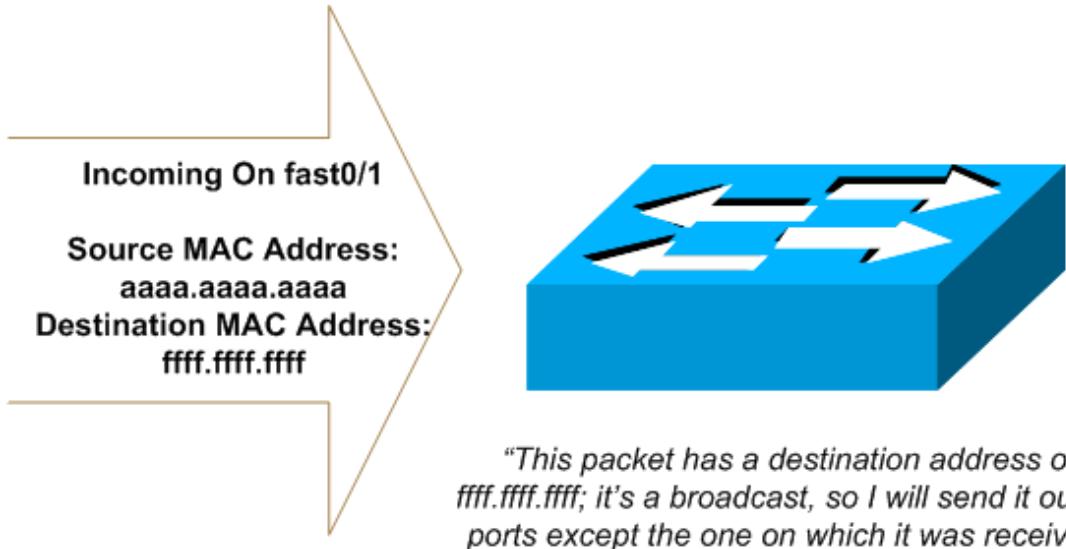
The switch's behavior when the source and destination MAC address are found off the same port.



Broadcasts And Multicasts

Broadcasts are forwarded out all interfaces except the port on which it was received. Broadcasts have a MAC address of FFFF.FFFF.FFFF. By default, multicasts are handled in the same fashion, although they do not share that MAC address with broadcasts.

Packets with a destination address of ffff.ffff.ffff are broadcasts.



In The Real World...

Multicasting is a concept that is only briefly mentioned in the CCNA exam. While the default behavior of a Cisco switch is to treat multicasting and broadcasting the same, real-world organizations that utilize multicasting rarely stick with that default.

IGMP Snooping is a Cisco feature that cuts down on the flooding of multicasts. The feature is not in the scope of the CCNA exam, but it's a good thing to know about when working with Cisco switches. Refer to www.cisco.com/univercd, select the Catalyst switch you're working with from the drop-down box, and refer to the Configuration Guide for more information about this (and many other) topics.

You should refer to this website often during both your Cisco studies and your networking career. Become comfortable navigating it.

Viewing The MAC Address Table

The basic command to view the MAC table on a 2950 switch is **show mac-address-table**. This command has several options, one of which is to show dynamically learned addresses only:

```
SW1#show mac-address-table dynamic  
Mac Address Table
```

Vlan	Mac Address	Type	Ports
1	0002.4b82.3941	DYNAMIC	Fa0/8
1	0004.4db9.0521	DYNAMIC	Fa0/6
1	0004.ddd1.4200	DYNAMIC	Fa0/9
1	0005.5d2f.f5db	DYNAMIC	Fa0/19
1	0005.5d43.f241	DYNAMIC	Fa0/19
1	000a.8a4b.fb13	DYNAMIC	Fa0/19
1	000a.8a4b.fb14	DYNAMIC	Fa0/20
1	000a.8a4b.fb16	DYNAMIC	Fa0/22
1	0010.7b1b.f181	DYNAMIC	Fa0/5
1	0010.7b1f.41c1	DYNAMIC	Fa0/2
1	0030.94b2.ef82	DYNAMIC	Fa0/19
1	0050.500f.3700	DYNAMIC	Fa0/1
1	00d0.ba1b.8fc0	DYNAMIC	Fa0/4
1	00d0.ba1b.8fc1	DYNAMIC	Fa0/19

Port-Based Authentication

The Cisco switch can use these MAC addresses to enforce **port security**. With port security, only devices with certain MAC addresses can connect to the port successfully. These MAC addresses are referred to as **secure MAC addresses**. The default number of secure MAC addresses is 1, but a maximum of 132 secure MACs can be configured.

There are three different types of secure MAC addresses:

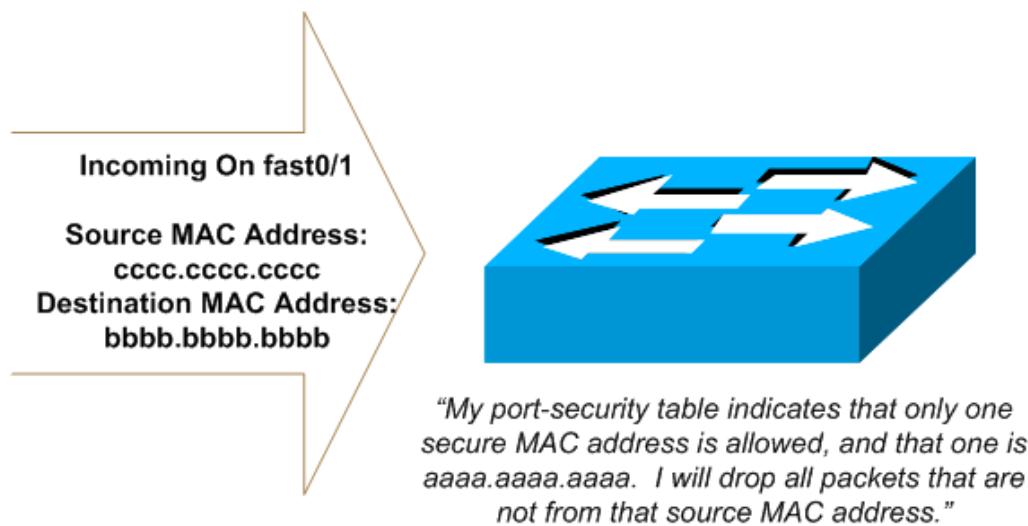
- A. **Static Secure** addresses are manually configured, kept in the address table, and added to the running configuration of the switch.
- B. **Dynamic Secure** addresses are dynamically learned, kept in the address table, but are removed when the switch restarts.
- C. **Sticky Secure** addresses are usually dynamically learned.
(They can be manually configured, but Cisco warns against it.)
The key is that this type of dynamically learned addresses are kept in the address table and are also added to the running configuration. If this configuration is saved, the addresses don't

have to be dynamically learned again when the switch is rebooted.

When the maximum number of secure MAC addresses is reached on the port, one of three actions will occur, depending on the port security mode. In **Protect** mode, once the number of secure MACs has reached its limit, packets with unknown source addresses are dropped. There is no notification that a violation has occurred. The port will continue to switch frames for the secure MAC address.

Consider the previous example where HostA is on port fast0/1. HostA is then removed from that port, and HostC is plugged into fast 0/1. If a single secure MAC address of aaaa.aaaa.aaaa was configured on port0/1 on the switch, HostC would then be unable to communicate with hosts off other ports.

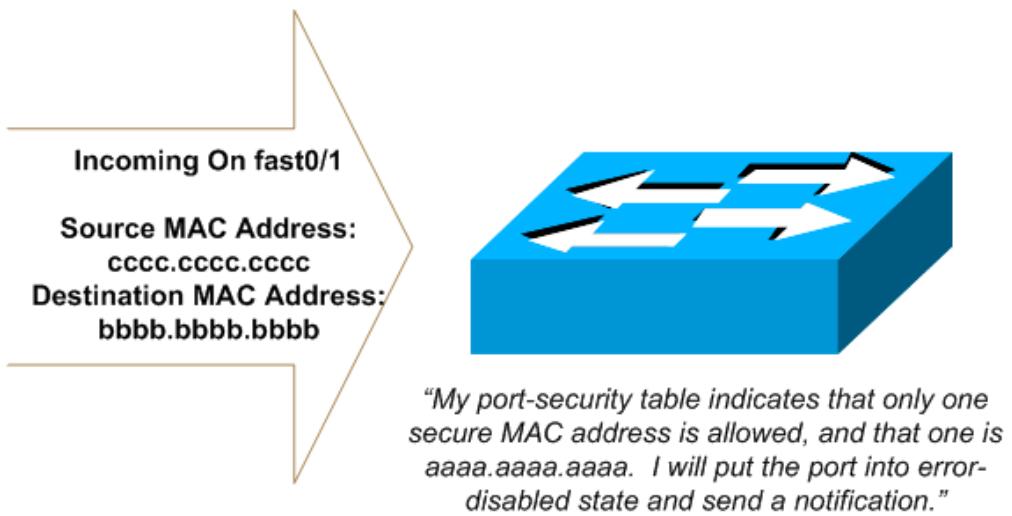
In Protect mode, frames with unknown MAC source addresses are dropped.



In **Restrict** mode, the same action is taken, but a syslog message is logged via SNMP, which is a messaging protocol used by Cisco routers. (Configuring SNMP on a 2950 is beyond the scope of the CCNA exam and this course.)

In **Shutdown** mode, the interface goes into error-disabled state, the port LED will go out, and a syslog message is logged. The port has to be manually opened. Shutdown mode is the default port-security mode.

In Shutdown mode, the port will go into error-disabled state if a violation is detected.



Spanning Tree Protocol

In almost every switching network, there will be path redundancy – that is, there will be more than one way to get to a given destination. If all the paths were available at all times, loops would form. The **Spanning Tree Protocol (STP)** prevents these loops from occurring by placing ports along the most desirable path into forwarding mode, while ports along less-desirable paths are placed into blocking mode. In this fashion, only one path is available and a loop cannot occur.

If a problem arises with the available path, STP will run the spanning-tree algorithm to recalculate the available paths and determine the best path. Ports along the new best path will be brought out of blocking mode and into forwarding mode, while ports along less-desirable paths are placed into blocking mode. In this fashion, only one path is available and a loop cannot occur.

If a problem arises with the available path, STP will run the spanning-tree algorithm to recalculate the available paths and determine the best path. Ports along the new best path will be brought out of blocking mode and into forwarding mode, with the end result again being a single path between LAN segments.

The Spanning Tree Process

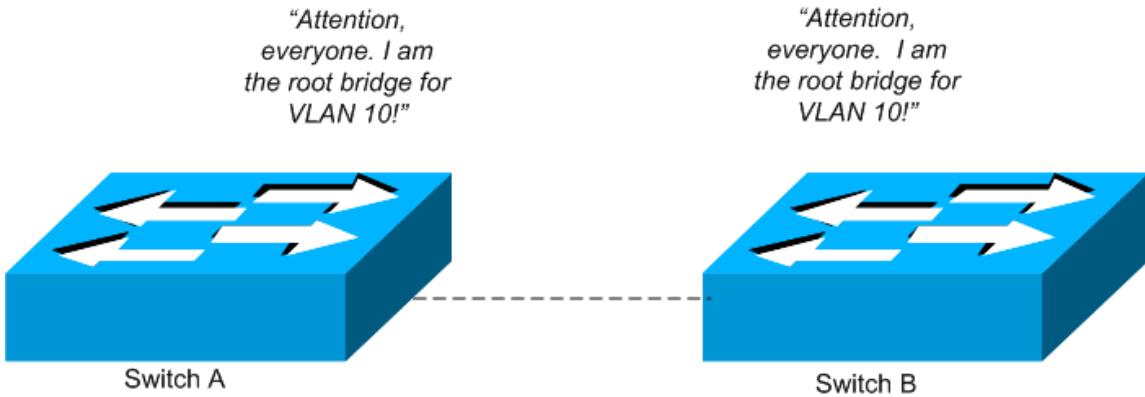
STP must first determine a **root bridge** for every LAN. When a switch is first powered on, it believes it is the root bridge for every single LAN. Since your network has multiple switches, and they all believe they are the root bridge for every LAN, there must be an election process to determine the true root bridge for each LAN.

The election process is carried out by the exchange of BPDUs (Bridge Protocol Data Units). The BPDU contains the following data:

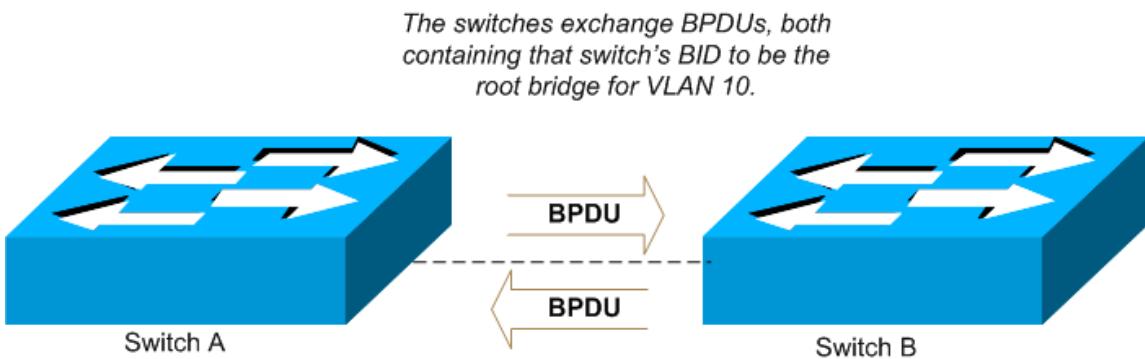
- A. **The Current Root Bridge's Bridge ID (BID).** The BID is a combination of the bridge's priority and MAC address. At the beginning of the election process, every switch thinks it is the root, so this will at first be the sending router's BID. As the election progresses, this BID will be the BID of the switch that this switch considers to be the root bridge. The bridge with the lowest BID will be the root bridge. The default priority value is 32768 for all switches; therefore, since the lowest BID wins, the switch with the lowest MAC address will become the root bridge unless the priority is changed.
- B. **Cost To Reach Root From This Bridge:** STP considers the path to have the lowest cost to be the best path.
- C. **BID Of The BPDU's Sender:** This simply identifies which switch sent the BPDU.

When a BPDU reaches a frame, it examines the BID of the switch that the sending switch thinks is the root bridge. If that BID is lower than that of the receiver, the receiver begins sending BPDUs announcing that switch as the root bridge. If the incoming BID is higher than that of the receiver, the receiver continues to announce itself as the root. This process continues until every switch has agreed on the root bridge.

In the following example, two switches held an election to determine which of them is the root bridge for Virtual LAN 10. (Virtual LANs, or VLANs, are discussed later in this chapter.) When VLAN 10 is configured, both switches think they are the root bridge for that VLAN.



The switches send each other BPDUs, containing the BID for each router. Remember, the lowest BID wins. Since the routers' priorities for VLAN 10 will be the same, the lowest MAC address will result in the lowest BID.



To view the results of the election, the command **show spanning-tree VLAN 10** is used:

```
SW1#show spanning-tree VLAN 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
  Address    0009.b738.9180
  This bridge is the root
  Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID Priority    32778 (priority 32768 sys-id-ext 10)
  Address    0009.b738.9180
  Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
  Aging Time 300

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/19        Desg FWD 19       128.19   P2p
  Fa0/20        Desg FWD 19       128.20   P2p
  Fa0/22        Desg FWD 19       128.22   P2p
```

```

SW2#show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
    Root ID  Priority  32778
              Address   0009.b738.9180
              Cost      19
              Port      19 (FastEthernet0/19)
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID Priority  32778 (priority 32768 sys-id-ext 10)
              Address   000a.8a4b.fb00
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

  Interface      Role Sts Cost      Prio.Nbr Type
  -----
  Fa0/19        Root FWD 19       128.19    P2p
  Fa0/20        Altn BLK 19       128.20    P2p
  Fa0/22        Altn BLK 19       128.22    P2p

```

Examining the output from **show spanning-tree vlan 10** on both switches, we note the following:

- A. The root bridge is indicated by the message “this bridge is the root” on SW1.
- B. There are three paths traffic destined for VLAN 10 could take; port 0/19, 0/20, and 0/22. On the root bridge, all three ports are in Forwarding mode. This is the expected and desired behavior.
- C. On the non-root bridge, SW2, only one of the ports is in forwarding mode, fast 0/19. This is the expected and desired behavior. This is the **root port**, the port with the lowest cost to the root. Since all three ports in this case have the same cost, the tiebreaker is the port priority, found under the “prio.nbr” field.

Note the two fields “Root ID” and “Bridge ID”. The Root ID information will be the same on every switch once the election is finished and the root bridge has been elected. The Bridge ID will be different on every switch; this is the BID, MAC address, and timer information of the local switch.

In The REAL World...

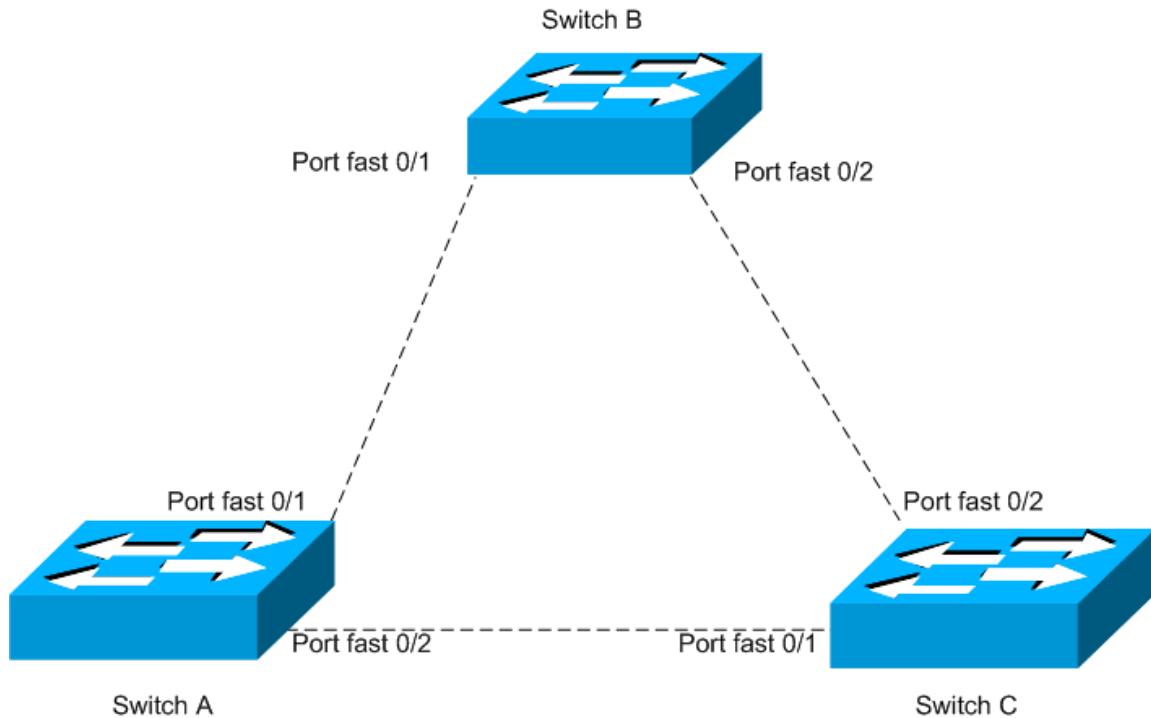
Leaving root bridge selection up to Cisco's defaults is not good network design. In a network with 100 switches, the same switch would end up being the root switch for every LAN in the network.

Once the root bridge and root ports on non-root bridges have been determined, there is one final set of ports that need to be put into forwarding mode. If an Ethernet segment has multiple bridges, only one bridge should be forwarding onto that segment. The bridge with the lowest cost to the root bridge will be elected the **designated bridge** for that segment, and its port on that Ethernet segment is the **designated port**.

	Forwarding Ports	How Port(s) Determined
Root Bridge	All	All ports forward.
Non-Root Bridge	Root Port	Received lowest-cost BPDU from Root Bridge.
Designated Bridge	Designated Port	Forwarding lowest-cost BPDU onto segment for which it is the Designated Bridge.

An Illustration Of The Spanning-Tree Process

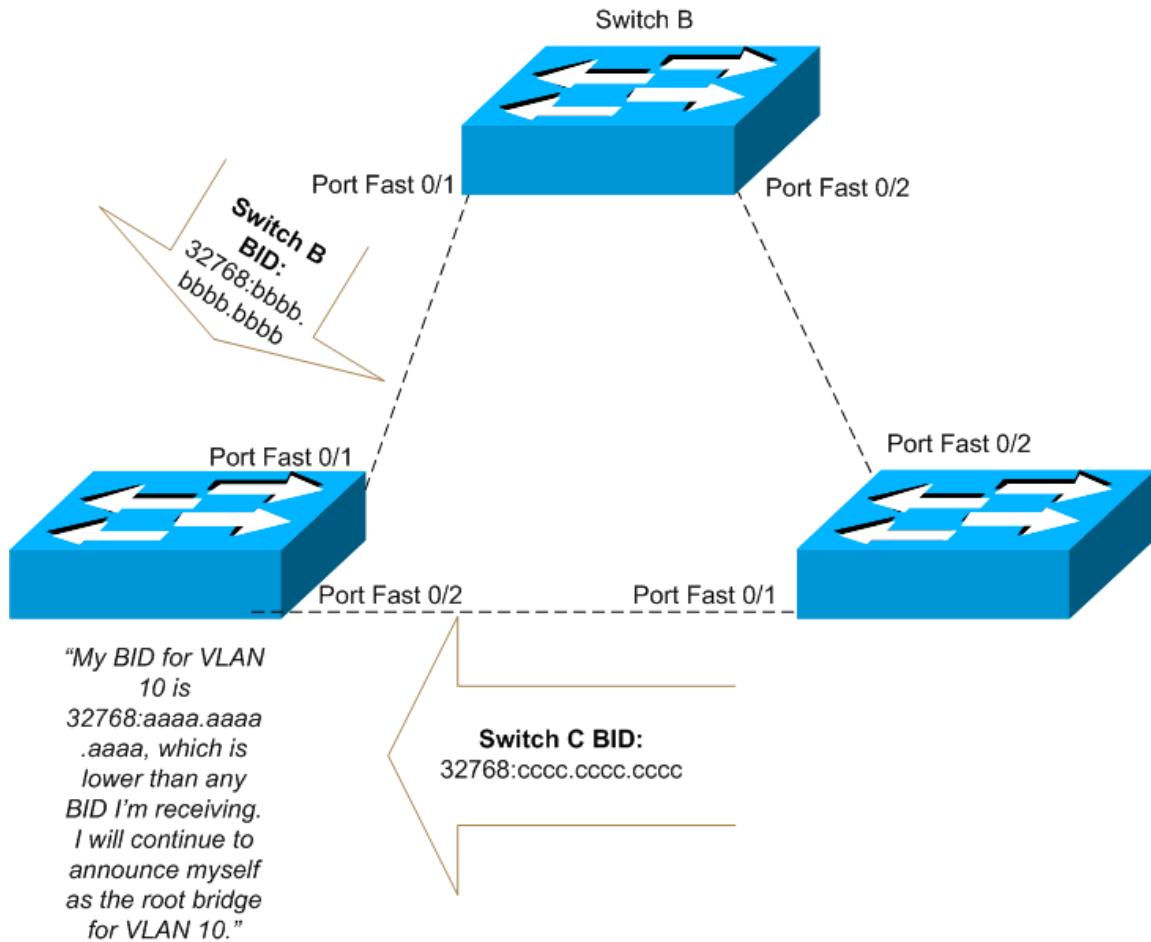
Consider the following network. Three switches, fully meshed, see VLAN 10 come online. All three switches will believe they are the root bridge for VLAN 10.



	MAC Address	Default Priority	Bridge ID (BID)
Switch A	aaaa.aaaa.aaaa	32768	32768:aaaa.aaaa.aaaa
Switch B	bbbb.bbbb.bbbb	32768	32768:bbbb.bbbb.bbbb
Switch C	cccc.cccc.cccc	32768	32768:cccc.cccc.cccc

Each switch will send BPDUs out each of their port announcing themselves as the root switch. To clarify the root bridge election process, let's examine the actions of each switch as it receives BPDUs from the other two switches.

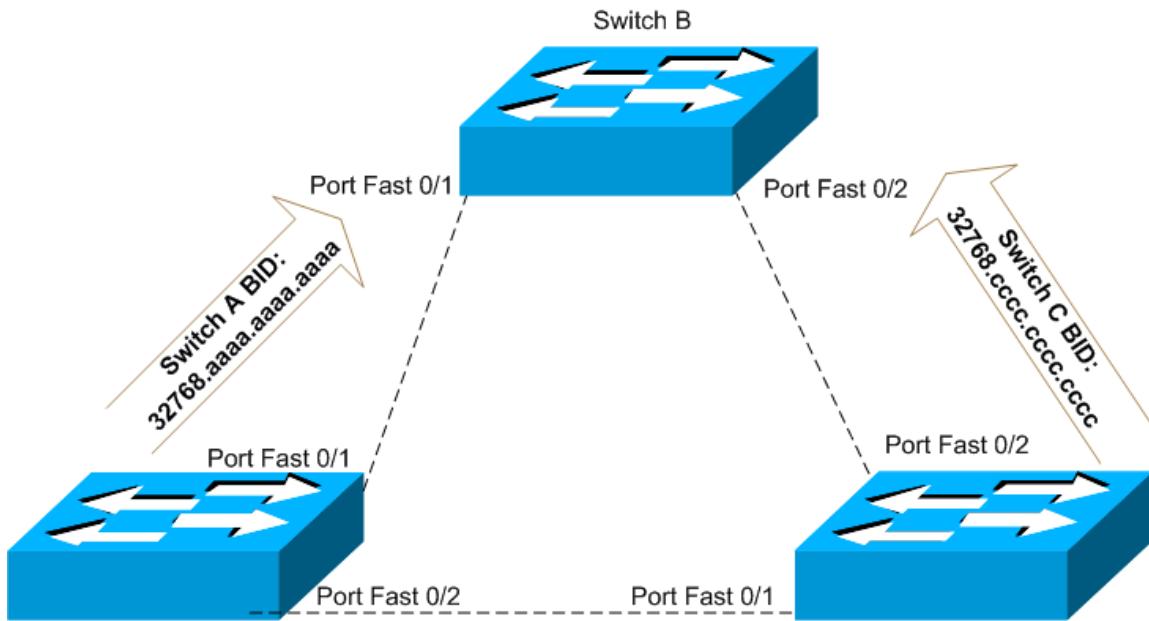
Switch A's Behavior During The Root Bridge Election:



Switch A receives BPDUs from Switch B and Switch C, each claiming they are the root bridge. Switch A examines their BIDs, and sees that its own BID is lower than either of the other bridges' BIDs. Switch A will continue to advertise itself as the root bridge.

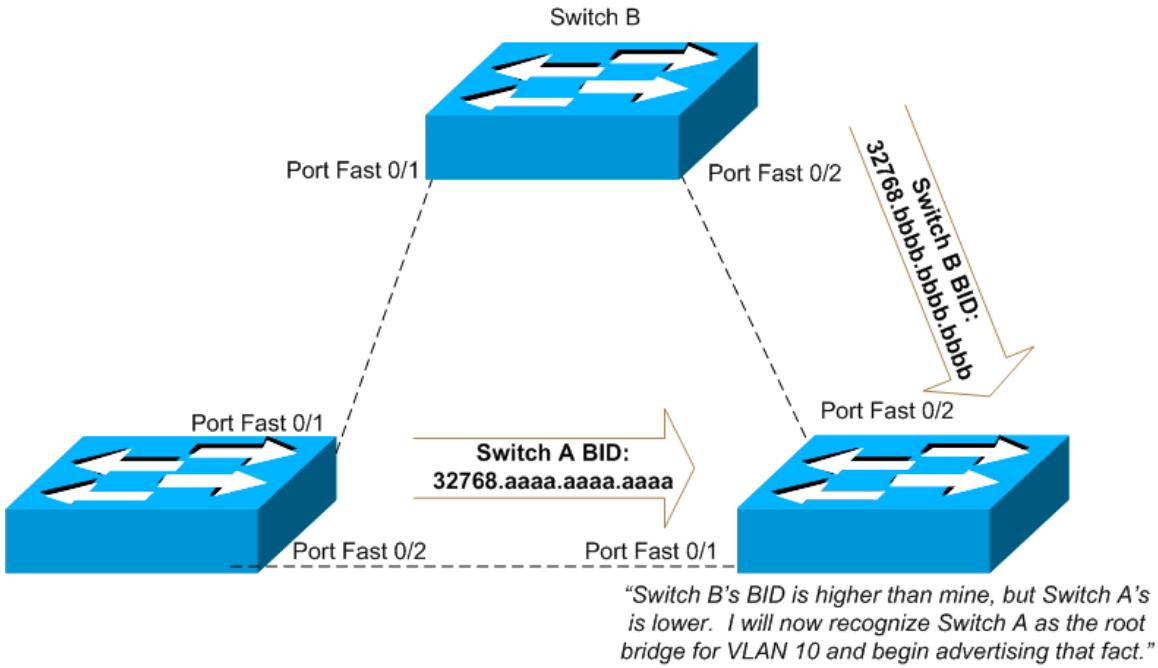
Switch B's Behavior During The Root Bridge Election:

"Switch C's BID is higher than mine, but Switch A's is lower. I will now announce Switch A as the root bridge for VLAN 10."



Switch B receives BPDUs from Switch A and Switch C, each claiming they are the root bridge. Switch B examines the BIDs from these switches. Switch B sees it has a lower bid than Switch C, but a higher bid than Switch A. Switch B recognizes Switch A should be the root bridge due to its lower BID. Switch B will now send BPDUs naming Switch A as the root bridge.

Switch C's Behavior During The Root Bridge Election:



Switch C receives BPDUs from Switch A and Switch B. Switch C will see that it has a higher BID than Switch A, and also that Switch B is naming Switch A as the root bridge. Switch C will now recognize Switch A as the root bridge and will send BPDUs naming Switch A as the root bridge.

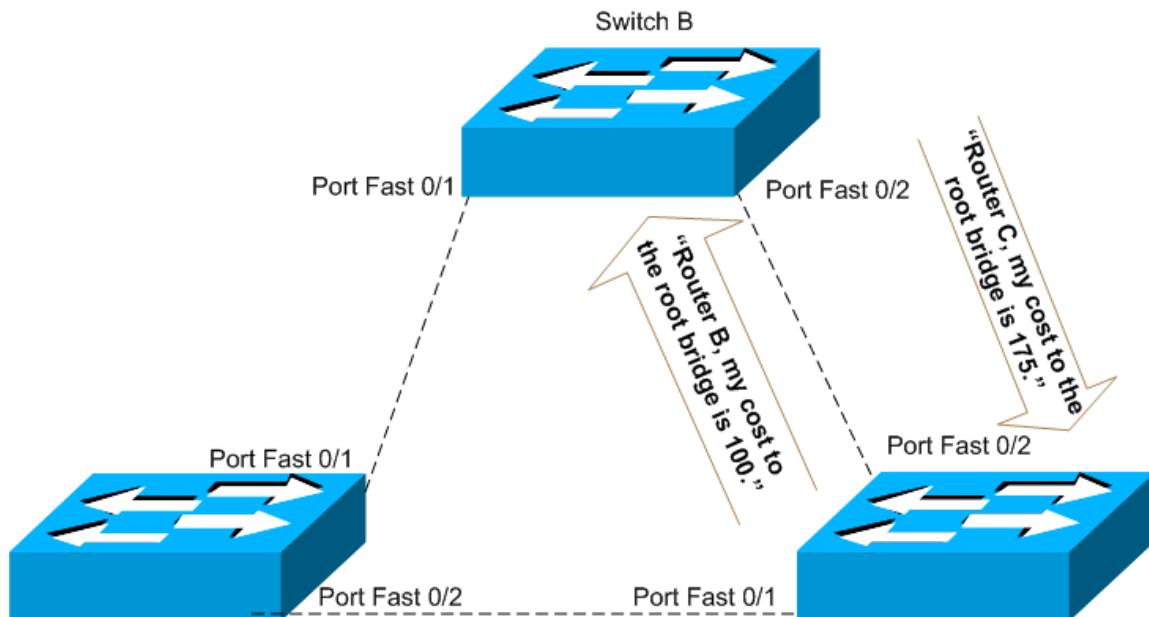
All three bridges now recognize Switch A as the root bridge. Accordingly, ports 0/1 and 0/2 on Switch A are now placed into Forwarding mode.

Next, the root ports on each non-root bridge must be selected. Each non-root bridge has two different ports that it can reach the root bridge through, but the **cost** is lower on the directly connected ports. Those ports will now be selected as the root port on their respective switches.

The Current Port Status On All Three Switches:

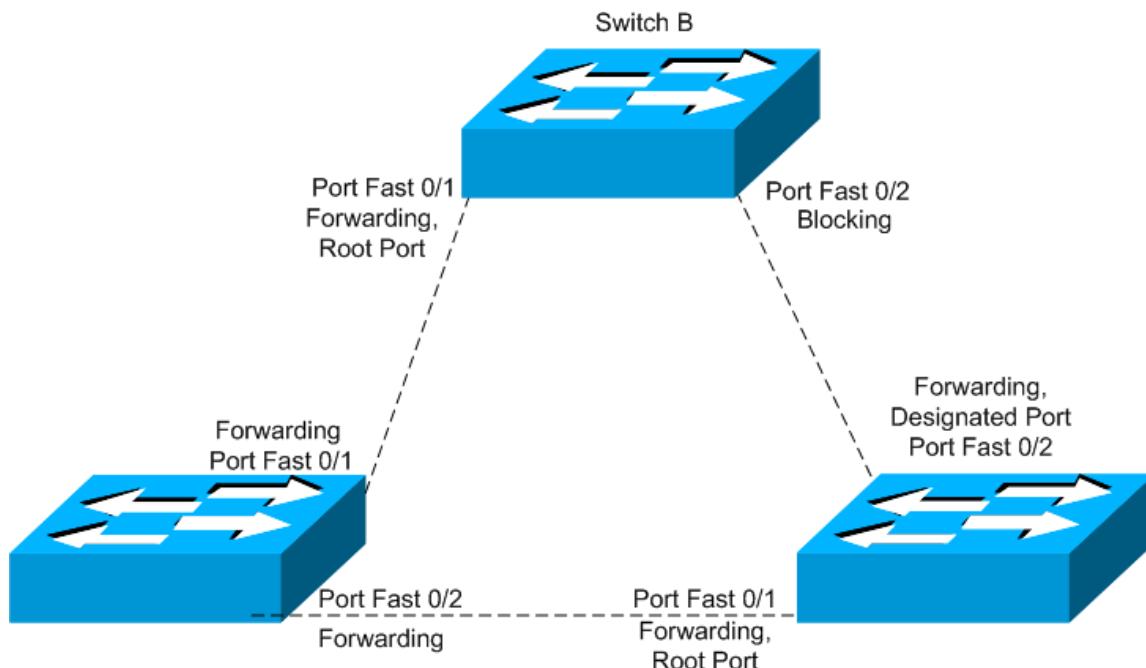
	Bridge Type	Forwarding Ports
SwitchA	Root	All
SwitchB	Non-Root	Port 0/1
SwitchC	Non-Root	Port 0/1

Finally, either Switch B or Switch C must be elected the designated bridge of the LAN segment connecting them. Switch B is advertising to Switch C that it can reach the root with a cost of 175; Switch C is advertising to Switch B that it can reach the root with a cost of 100. The lowest cost will win the election, resulting in Switch C becoming the designated bridge, and port 0/2 on Switch C is the designated port for that LAN segment.



The Final Bridge And Port Assignments And States:

	Bridge Type	Forwarding Ports	Blocking Ports
SwitchA	Root Bridge	All, by default	None, by default
SwitchB	Non-Root	0/1 (Root Port)	0/2
SwitchC	Designated Bridge	0/1 (Root Port), 0/2 (Designated Port)	None



The STP Timers

Once these elections have taken place, the root bridge will be sending a Hello BPDU out all its ports every two seconds. This Hello BPDU serves as the heartbeat of STP, since as long as the non-root bridges receive it, they know the path to the root is unchanged and stable.

Once that heartbeat disappears, it's an indication of a failure somewhere along the path. STP will run the spanning-tree algorithm to determine the best available path, and ports will be brought out of blocking mode as needed to build this path.

The Hello BPDUs carry values for three timers that are used by all bridges in identifying situations when the STP algorithm needs to be run again:

- A. **Hello Time:** Time between Hello BPDUs being sent by the root.
Default: 2 seconds.
- B. **MaxAge:** The bridge should wait this amount of time after not hearing a Hello BPDU before attempting to change the STP topology. Default: 20 seconds.
- C. **ForwardDelay:** The amount of time a port should stay in the listening and learning stages as it changes from blocking to forwarding mode. Default: 15 seconds.

The STP Interface States:

When a port goes from blocking state to forwarding state, it does not do so instantly. If it did, loops could result. STP has interfaces go through two intermediate states between blocking and forwarding: **listening** and **learning**.

A port coming out of blocking state first goes into listening state. The port is listening for Hello BPDUs from other possible roots. The port will listen for the defined value of the ForwardDelay timer, 15 seconds by default. The port will then go into learning state. This state has the port learn the new location of MAC addresses, but will not allow forwarding of them, since there is a good possibility other switches are currently converging and loops could develop if MAC addresses were learned from other switches during convergence. Learning state also lasts the duration of the ForwardDelay timer.

PortFast

Consider the amount of time a port ordinarily takes to go from blocking to forwarding when it stops receiving Hello BPDUs:

State	Duration Of State	Total Time Elapsed
Blocking	Waits 20 Seconds After Not Hearing Hellos (MaxAge)	20 Seconds
Listening	15 Seconds (ForwardDelay)	35 Seconds
Learning	15 Seconds (ForwardDelay)	50 Seconds
Forwarding		

Almost an entire minute will pass before the port can begin forwarding. Although the listening and learning stages are there for a reason, the primary one being loop prevention during convergence, these stages can be bypassed with the Cisco feature **portfast**.

Portfast allows a port to bypass the listening and learning stages of this process., but is only appropriate to use on switch ports that connect directly to an end-user device, such as a PC. Using portfast on a port leading to another networking device can lead to loops. That threat is so serious that Cisco even warns you about it on the router when you configure portfast:

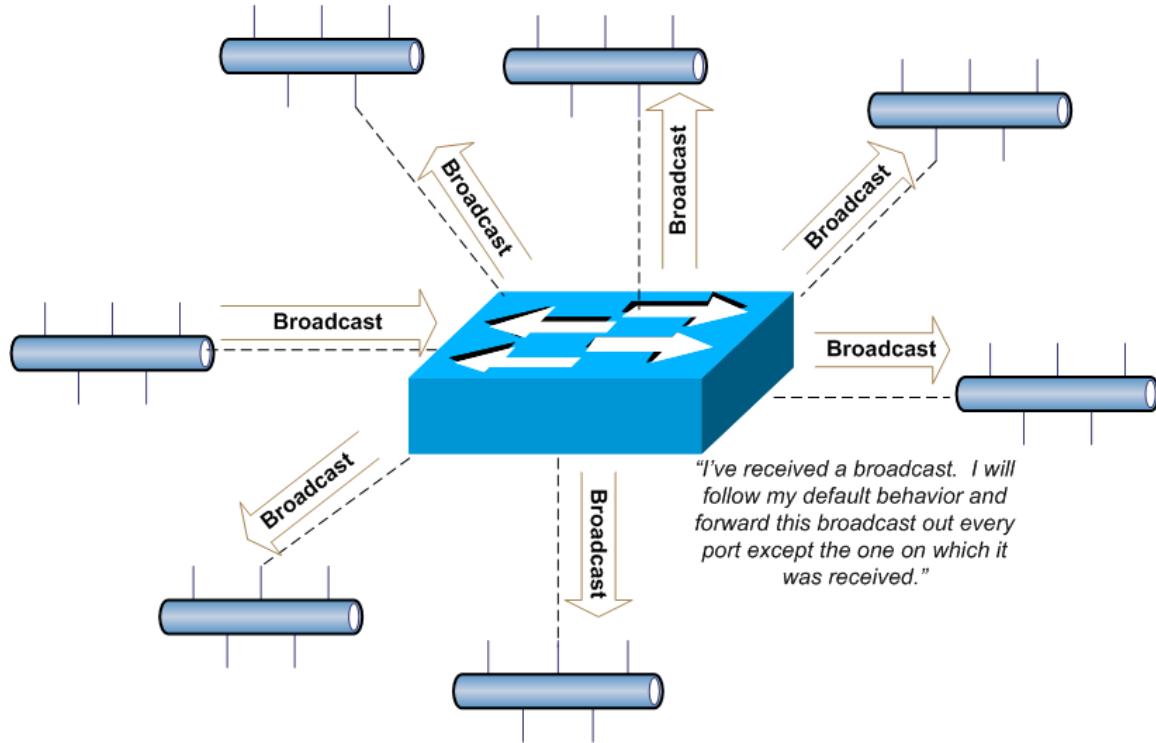
```
SW2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#int fast 0/2
SW2(config-if)#spanning portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/2 but will only
have effect when the interface is in a non-trunking mode.
SW2(config-if)#
```

VLANs and Trunking

By default, switches forward broadcasts. Consider a situation where you have 8 different devices connected to a switch. Recall that switches flood broadcasts by default; the broadcast will be sent out all interfaces except the one it was received on. As the number of devices connected to the switch increases, the amount of bandwidth taken up with unnecessary broadcasts increases.

The Default Behavior Of A Switch Receiving A Broadcast:

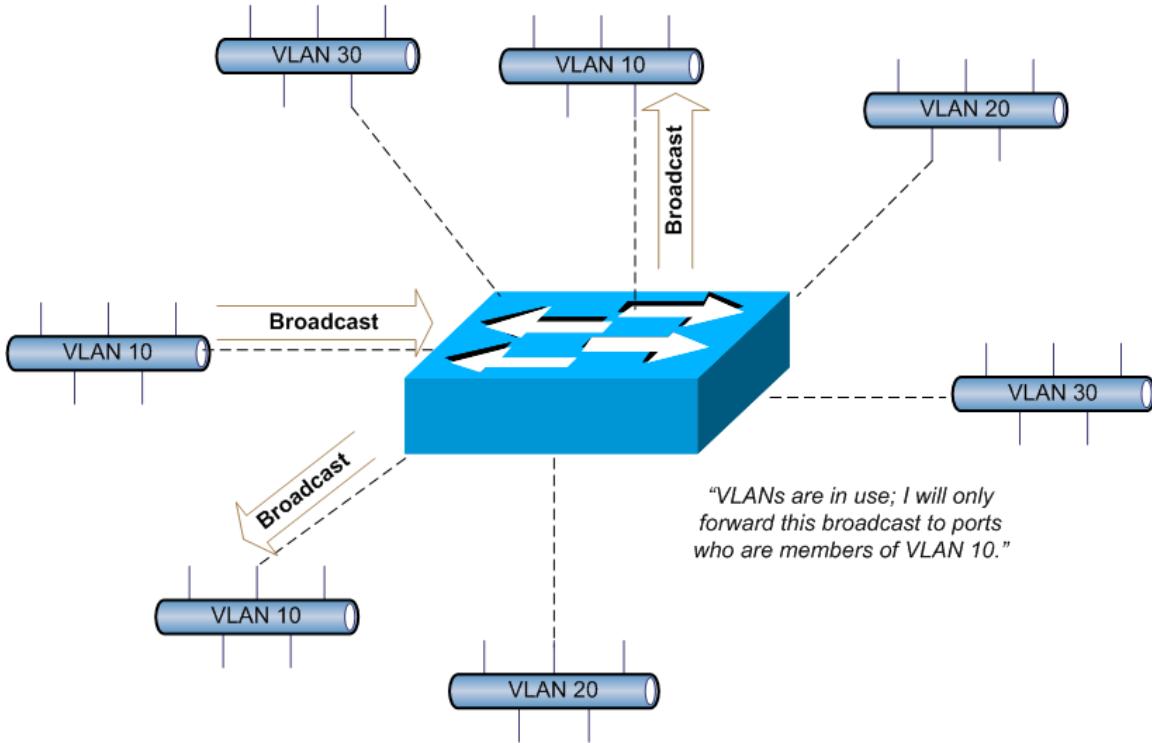


If your users are grouped into multiple departments, there is no need to users not in their department to receive these broadcasts. Preventing those broadcasts from reaching unnecessary users can save a great deal of bandwidth. Or perhaps for security purposes, the departments need to be kept separate on the switch, and unable to be reached from ports in other departments.

VLANs allow this kind of logical grouping. A VLAN is simply a group of ports placed into its own, smaller broadcast domain. Layer 2 switches will forward frames between ports in the same VLAN, but will not do so between ports not in the same VLAN.

Consider the previous example. Three VLANs have been created; VLAN 10, 20, and 30. Broadcasts sent by a host in VLAN 10 will only be forwarded to other hosts in VLAN 10. Hosts in other VLANs will never see the broadcasts.

Switches Do Not Forward Broadcasts Between VLANs.

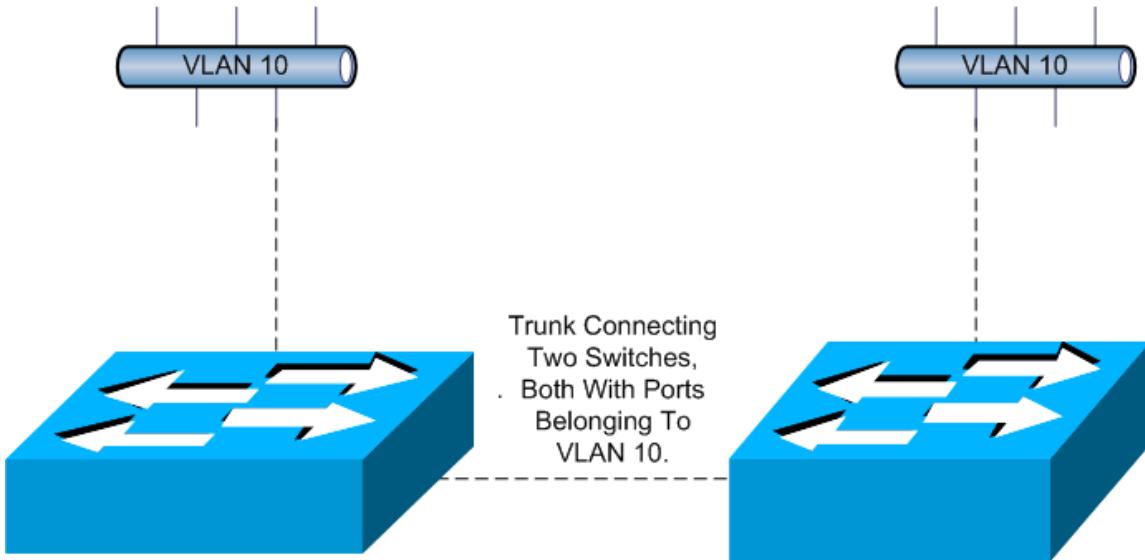


The broadcast is now sent only to other ports that belong to VLAN 10.

The problem that arises is that *no* traffic is going to be sent between VLANs. A PC in VLAN 10 will be unable to communicate with a PC in VLAN 30, and vice versa. Communication between devices in different VLANs requires the use of a "Layer 2 / Layer 3" switch, or a router. Resolution of this issue will be addressed in the Advanced TCP/IP section. For now, though, it's important to understand that VLANs help to limit broadcasts, and devices in different VLANs, by default, cannot communicate.

The switch currently focused on in the CCNA exam, the 2950, cannot perform this task; use of a router will be required.

Trunking refers to allowing VLAN traffic to flow over interconnected switches in the same network. In order for a switch to know what VLAN a frame received from another switch is destined for, a tag is placed on the frame indicating the destination VLAN. In this fashion, a VLAN with members on multiple switches can communicate.



Cisco switches can generally use one of two trunking protocols, ISL and IEEE 802.1q, commonly referred to as "dot1q". Note that the 2950 switch does **not** support ISL trunking, but many other Cisco switches do.

ISL is the Cisco-proprietary trunking protocol. For this reason, it can only be used between two Cisco switches. The entire frame is encapsulated with an ISL header and trailer.

Dot1q is the industry standard for trunking; if a non-Cisco switch is involved in the trunk, this is the trunking protocol to use. Dot1q does not encapsulate the entire frame. Instead, a 4-byte header is added to the Ethernet header, indicating the VLAN to which the frame is intended.

The key difference between the two is the way they handle the **native vlan**. By default, the native vlan is VLAN 1. The native vlan is a kind of "default vlan" in that when dot1q is ready to transmit a frame over the trunk, the protocol will not put that 4-byte header onto the frame. Instead, the frame is transmitted "as-is". When the receiving frame sees there is no header on the frame, it assumes the frame is intended for the native vlan, and it is forwarded accordingly. Dot1q allows for the changing of the native vlan.

ISL does not recognize the concept of the native vlan. Every single frame transmitted over an ISL trunk will be encapsulated.

	ISL	Dot1q
Definition	Cisco-proprietary	Industry standard
Encapsulation	Entire frame is encapsulated	No encapsulation; Places 4-byte VLAN ID in header
Native VLAN	Does not recognize	VLAN 1 by default; Can be changed

VLAN Trunking Protocol (VTP)

VTP allows switches to advertise VLAN information between other members of the same VTP domain. VTP allows a consistent view of the switched network across all switches. When a VLAN is created on one switch in a VTP server, all other VTP devices in the domain are notified of that VLAN's existence. VTP servers will know about every VLAN, even VLANs that have no members on that switch.

Switches run VTP in one of three modes. In **server mode**, VLANs can be created, modified, and deleted on a VTP server. When these actions are taken, the changes are advertised to all switches in the VTP domain. VTP Servers keep VLAN configuration information upon reboot.

In **client mode**, the switch cannot modify, create, or delete VLANs. VTP clients cannot retain VLAN configuration information upon reboot; they have to obtain this information from a VTP server.



In real-world networks, this is generally done to centralize the creation and deletion of VLANs. An interesting side effect of the server/client methodology is that if a VLAN is only to have ports on the VTP client switch, *the VLAN must still first be created on the VTP server*. The VTP client will learn about the VLAN from the VTP server, and ports can then be placed into that VLAN.

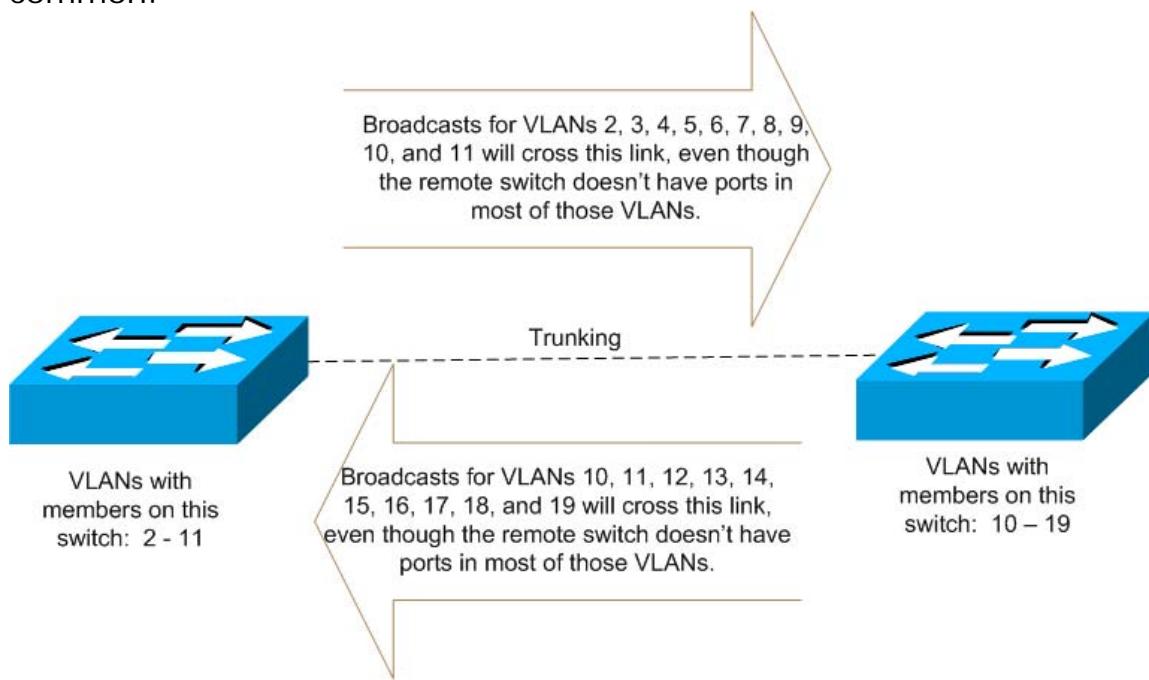
A. The third VTP mode is **transparent mode**. VTP switches in this mode ignore VTP messages. They do forward the VTP advertisements received from other switches. VLANs can be created, deleted, and modified on a transparent server, but those changes are not advertised to the other switches in the VTP domain.

In The REAL World...

If you think the concept of VTP transparent mode sounds a little confusing, you're right. It's rarely seen in production networks; I've personally never seen it used. It is important to know it exists, though.

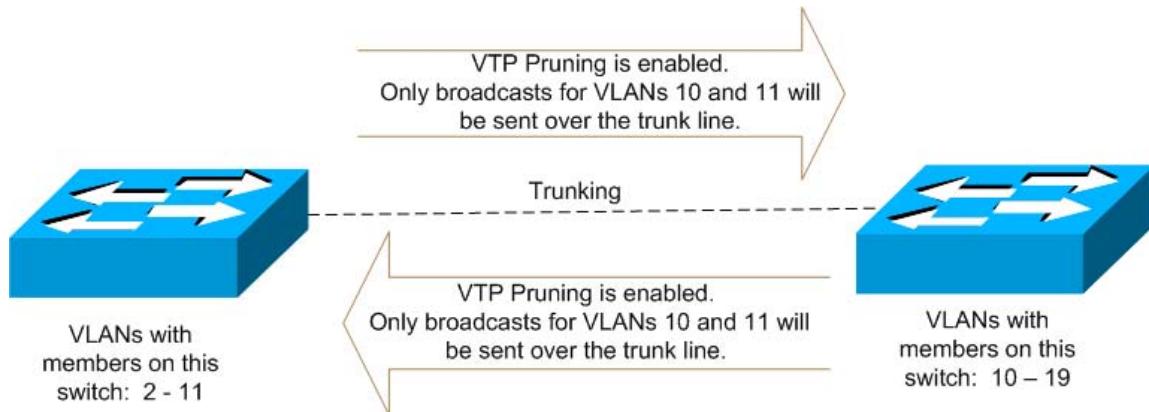
VTP Pruning

Consider this situation: two switches are trunking, and each has ports in ten VLANs. Of all those VLANs, the switches only have two in common.



The switches both have ports in VLANs 10 and 11, but have no other common VLANs. By default, broadcast and multicast traffic destined for *any* VLAN will cross the trunk, resulting in a lot of unnecessary traffic crossing the link.

This default behavior can be stopped by enabling vtp pruning. With vtp pruning enabled on these switches, a VLAN's broadcasts will be sent across the trunk only when there are ports belonging to that particular VLAN on the opposite switch. Broadcasts for VLANs 10 and 11 will go across the trunk, but not for the other VLANs.



In The REAL World...

No doubt you've noticed the emphasis on controlling broadcasts on and between switches. Unnecessary broadcasts quickly take up bandwidth, and worse, can result in a *broadcast storm*.

Consider the earlier example where a broadcast was forwarded out seven ports because VLANs were not yet configured. If each of those segments responds with its own broadcast, a snowball effect can quickly occur. Not only might the switches be overwhelmed by handling the multiplying broadcast traffic, but unicast and multicast traffic may not be able to get through.

The word “proactive” is overused today, but you cannot be too proactive in preventing broadcast storms. Don’t allow switches to forward broadcasts where they are not needed.

Port Types And Their Effect On Trunking

A Cisco switch port can run in **access** or **trunk** modes:

An **access port** belongs to one VLAN, and will only carry traffic for that particular LAN. When traffic arrives on an access port, it is assumed that it belongs to the LAN assigned to that port.

Trunk ports carry traffic for multiple VLANs. The default behavior of a trunk port is that it is a member of all VLANs in the VLAN database. The Cisco 2950 supports only 802.1q ("dot1q") trunk ports.

A port can be placed into a dynamic trunk mode, or into "static" trunk mode. While ports on opposite ends of a potential trunk can be placed into different trunk mode types, trunking may not occur if an invalid combination of these types is used.

Trunk	"Static" trunking. Interface enters permanent trunk mode, and will negotiate trunking even if the remote port is not a trunking interface.
Dynamic Desirable	Actively attempts to trunk with remote port.
Dynamic Auto	Does not actively attempt to trunk with remote port, but will form trunk if remote port attempts to negotiate trunking.

Trunk Formation With Different Mode Combinations:

Port 1	Port 2	Trunk Formed ?
Trunk	Trunk	Yes
Trunk	Dynamic Desirable	Yes
Trunk	Dynamic Auto	Yes
Dynamic Desirable	Trunk	Yes
Dynamic Desirable	Dynamic Desirable	Yes
Dynamic Desirable	Dynamic Auto	Yes
Dynamic Auto	Trunk	Yes
Dynamic Auto	Dynamic Desirable	Yes
Dynamic Auto	Dynamic Auto	NO

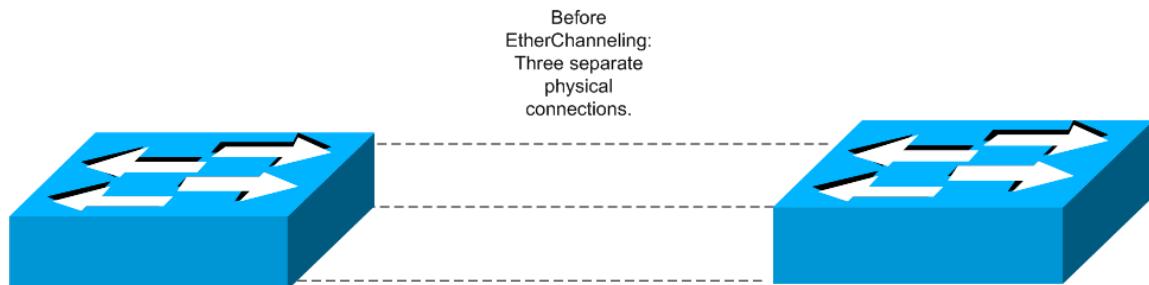
The only combination that results in no trunk forming is if both ports are running in Dynamic Auto. Even though a trunk can be formed with one port in that mode, if both ports are in Dynamic Auto, they are both waiting for the other port to begin negotiating the trunk, and no trunk will ever form.

EtherChannels

An EtherChannel is the logical bundling of two to eight parallel Ethernet trunks. This provides greater throughput, and is another effective way to avoid the 50-second wait between blocking and forwarding states in case of a link failure.

The Spanning-Tree Protocol considers an EtherChannel to be one link. If one of the physical links making up the logical EtherChannel should fail, there is no STP reconfiguration, since STP doesn't know the physical link went down. STP sees only the EtherChannel, and a single link failure will not bring an EtherChannel down.

Consider this example. There are three trunk ports between two switches:



show spanning vlan 10 illustrates that STP sees three separate links:

```

SW2#show spanning vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
    Root ID  Priority  32778
              Address   0009.b738.9180
              Cost      19
              Port      19 (FastEthernet0/19)
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID Priority  32778 (priority 32768 sys-id-ext 10)
              Address   000a.8a4b.fb00
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 15

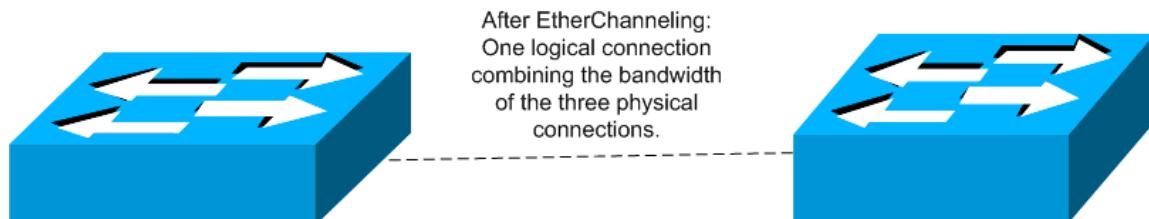
  Interface      Role Sts Cost      Prio.Nbr Type
  -----
  Fa0/19        Root FWD 19       128.19   P2p
  Fa0/20        Altn BLK 19       128.20   P2p
  Fa0/22        Altn BLK 19       128.22   P2p

```

If port 0/19 goes down, port 0/20 will begin the process of going from blocking to learning. In the meantime, communication between the two switches is lost.

This temporary lack of a forwarding port can be avoided with an EtherChannel. By combining the three physical ports into a single logical link, not only is the bandwidth of the three links combined, but the failure of a single link will not force STP to recalculate the spanning tree.

After configuring an EtherChannel on each router with the interface-level command **channel-group**, the output of commands **show interface trunk** and **show spanning vlan 10** show STP now sees the three physical links as one logical link.



```

SW2#show interface trunk

Port      Mode       Encapsulation  Status        Native vlan
Po1       on         802.1q          trunking     1

Port      Vlans allowed on trunk
Po1       1-4094

Port      Vlans allowed and active in management domain
Po1       1,10

Port      Vlans in spanning tree forwarding state and not pruned
Po1       none

```

```

SW2#show spanning vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
    Root ID    Priority    32778
                Address     0009.b738.9180
                Cost         9
                Port        65 (Port-channel1)
                Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
                Address     000a.8a4b.fb00
                Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
                Aging Time  300

    Interface  Role Sts Cost      Prio.Nbr Type
    -----      -----
    Po1        Root FWD 9        128.65   P2p

```

LAN Switching Questions

1. Which of the following is a Layer 2 broadcast?

- A. **AAAA.AAAA.AAAA**
- B. **0000.0000.0000**
- C. **FFFF.FFFF.FFFF**
- D. **1111.1111.1111**
- E. **200.200.200.255**

Answer: C. FFFF.FFFF.FFFF is a Layer 2 broadcast.

2. Which statement best describes how a switch handles broadcasts by default?

- A. **All broadcasts are filtered.**
- B. **All broadcasts are forwarded.**
- C. **They are forwarded only to the VLANs for which they are intended.**
- D. **They are forwarded to the VTP server, which performs broadcast forwarding for the VTP domain.**

Answer: B. The default behavior of a switch is to forward broadcasts. VLANs can be configured to limit broadcast propagation, but this is not a default behavior.

3. A switch receives a frame with a destination MAC address of aaaa.aaaa.aaaa. The switch does not know what port should be used to reach the device with that MAC address. What will the switch do with this frame?

- A. **The switch will filter the frame.**
- B. **The switch will forward the frame out all ports.**
- C. **The switch will forward the frame out all ports except the port on which it was received.**
- D. **The switch will perform a reverse ARP operation to get the MAC address, then the frame will be unicast.**

Answer: C. If the switch does not know which port to forward a frame out of, the switch will forward the frame out all ports except that which it was received.

4. A switch receives a frame with a destination MAC address of bbbb.bbbb.bbbb on port 0/3. The switch knows the destination address can be reached out the same port. What will the switch do ?
- A. **The switch will filter the frame.**
 - B. **The switch will forward the frame to the destination.**
 - C. **The switch will forward the frame out all ports except the one it was received on.**
 - D. **The switch will send a message to the source, indicating that no more frames for this destination should be sent to the switch.**

Answer: A. The switch will filter the frame; that is, the frame will not be forwarded out any ports.

5. What two sources of information do switches use to build their MAC address tables? (Choose two.)
- A. **The source address of incoming frames.**
 - B. **The destination address of incoming frames.**
 - C. **Statically configured MAC addresses.**
 - D. **VTP servers.**
 - E. **MHOST files.**

Answer: A, C. The main source of MAC addresses that are entered into the MAC table is the source address of incoming frames. The destination address doesn't help build the table; that address is either already known, or the frame is flooded. Neither action adds an entry to the MAC table. MAC addresses can also be statically entered. VTP servers have nothing to do with building the MAC table, and there's no such thing as an MHOST file in Cisco.

6. By default, what frames are flooded out all ports except the port upon which it was received? Choose two.

- A. **Frames destined for a known unicast address.**
- B. **Frames destined for an unknown unicast address.**
- C. **Broadcast frames.**
- D. **VTP frames.**
- E. **STP frames.**

Answer: **B, C.** Both broadcast frames and frames destined for an unknown unicast address are flooded out all ports except the port it was received on.

7. What command allows viewing of the MAC address table ?

- A. **show Layer2 address**
- B. **show mac-address-table**
- C. **show macs**
- D. **show table**

Answer: **B.**

8. Which of the following are port security violation modes on a Cisco 2950 switch? (Choose three.)

- A. **Shutdown**
- B. **Report**
- C. **Reload Port**
- D. **Request**
- E. **Restrict**
- F. **Protect**
- G. **Purge MAC**
- H. **Preserve**

Answer: **A, E, F.**

9. Which of the following is true of port security on a Cisco 2950 switch? Choose two.
- A. The default violation mode is Shutdown.
 - B. The default violation mode is Restrict.
 - C. The default violation mode is Protect.
 - D. Any switchport can be configured with port security.
 - E. Ports connecting to end hosts cannot be configured with port security.
 - F. Ports connecting to other switches cannot be configured with port security.

Answer: A, F. The default mode is shutdown, and ports connecting to other switches cannot use port security. To configure a port as not having a switch attached to that port, configure it as an access port (**switch mode access**).

10. In Cisco port security, what is true of the term "sticky secure MAC address"? (Choose two.)

- A. Once configured, the address cannot be removed from the switch without erasing the router configuration.
- B. No other secure MAC addresses can be configured on a port once a sticky secure MAC has been configured.
- C. The sticky MAC address will be learned from the first MAC address the switch learns on that port.
- D. The sticky MAC address will be added to the running configuration of the switch.
- E. The sticky MAC address cannot connect to any other port on the switch except the one it is learned on.

Answer: C, D. By configuring **switchport port-security mac-address sticky** on an interface already configured for port-security, the MAC source of the next frame received on that port is the secure MAC address. Additionally, the sticky MAC will be added to the running configuration of the switch. Saving the running configuration will result in that secure MAC address remaining on that port after a switch reboot.

11. What protocol runs at Layer 2 and prevents loops from forming in networks with redundant paths?

- A. TCP
- B. IP
- C. STP
- D. UDP
- E. ISL

Answer: C. Spanning-Tree Protocol (STP) runs on switches to prevent loops from forming when more than one path between destinations exists.

12. Which statement best describes how STP handles broadcasts?

- A. Broadcasts travel the single path between destinations, like all other data.
- B. In order to ensure the entire network receives the broadcast, broadcasts are “leaked” through blocked ports.
- C. STP implicitly denies broadcasts.
- D. STP sends broadcasts out root ports only.

Answer: A. STP takes no particular action toward broadcasts; broadcasts will travel the single path between destinations like all other data.

13. Your network has four switches. The priorities and MAC addresses are:

	Priority	MAC Address
Switch 1	32768	1111.1111.1111
Switch 2	16384	2222.2222.2222
Switch 3	8194	3333.3333.3333
Switch 4	8194	4444.4444.4444

Which statements are true of the root bridge?

- A. Switch 1 will become the root bridge.
- B. Switch 2 will become the root bridge.
- C. Switch 3 will become the root bridge.

D. Switch 4 will become the root bridge.

Answer: C. The lowest BID wins the election for root port, and the BID consists of the priority and the MAC address. For example, Switch 3's BID is 8194:3333.3333.333, and Switch 4's BID is 8194.4444.4444.4444. Since their priorities are the same, the difference is that Switch 3's MAC address is lower than Switch 4's, making Switch 3's BID lower.

14. Your network has four switches. The priorities and MAC addresses are:

	Priority	MAC Address
Switch 1	4096	1111.1111.1111
Switch 2	16384	2222.2222.2222
Switch 3	16384	3333.3333.3333
Switch 4	32768	4444.4444.4444

Which statements are true of the root bridge ?

- A. Switch 1 will become the root bridge.**
- B. Switch 2 will become the root bridge.**
- C. Switch 3 will become the root bridge.**
- D. Switch 4 will become the root bridge.**
- E. The MAC address comes into play in this election.**
- F. The MAC address does not matter in this election.**

Answer: A, F. The lowest BID becomes the root election. Switch 1 has the lowest priority of the four, guaranteeing it the lowest BID. The MAC address is part of the BID, but no matter what the MACs are, Switch 1 will still have the lowest BID.

15. Your root switch is connected to another switch by three trunks.
You run **show spanning vlan 10** on your root switch and receive the following output:

```
SW1#show spanning vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
              Address    0009.b738.9180
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
              Address    0009.b738.9180
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Po1           Desg FWD 9        128.65   P2p
```

Which of the following statements is true? (Choose three.)

- A. **Hello BPDUs are originating from this switch.**
- B. **Hello BPDUs are being received from another switch.**
- C. **This switch had the highest BID during the root bridge election.**
- D. **This switch had the lowest BID during the root bridge election.**
- E. **This switch has an EtherChannel configured on it.**
- F. **This switch's forwarding interface is port 0/1 (po1).**

Answer: A, D, E. This is the root bridge, so Hello BPDUs are originating from this switch and are forwarded by other switches. To become the root bridge, the switch had to have the lowest BID during the election. Finally, "po1" indicates Port-channel1, which is an EtherChannel.

16. Your root switch is connected to another switch by three trunks.

You run **show spanning vlan 10** on your root switch and receive the following output:

```
SW1#show spanning vlan 10
```

```
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
              Address     0009.b738.9180
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-ext 10)
              Address     0009.b738.9180
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/19        Desg FWD 19       128.19   P2p
  Fa0/20        Desg FWD 19       128.20   P2p
  Fa0/22        Desg FWD 19       128.22   P2p
```

You notice that all three ports connecting to the other switch are in Forwarding mode. Which of the following is true?

- A. **There is an error with STP, because only one port can be in Forwarding mode.**
- B. **There is an error with the VLAN configuration.**
- C. **The other switch is rebooting, resulting in all three ports going into Forwarding mode. Once the other switch finishes coming back up, two of these ports will go into Blocking mode.**
- D. **There is no problem; this is normal.**

Answer: D. All ports on the root switch will be in Forwarding mode.

17. How many root ports can a non-root switch have for a given VLAN?

- A. **One.**
- B. **Two – one primary root port and one backup root port.**
- C. **As many as STP assigns the switch.**
- D. **None – nonroot switches don't have root ports.**

Answer: A. Nonroot switches have one root port per VLAN – period.

18. By default, how often does a root bridge send out Hello BPDUs?

- A. **Every second.**
- B. **Every two seconds.**
- C. **Every minute.**
- D. **Every two minutes.**
- E. **Never – root bridges don't send BPDUs.**

Answer: B. Root bridges send Hello BPDUs every two seconds.

19. The Hello BPDU value **ForwardDelay** determines the length of which two STP port states ? (Choose two.)

- A. **Blocking**
- B. **Listening**
- C. **Learning**
- D. **Forwarding**

Answer: B, C. **ForwardDelay** determines how long ports stay in the Listening and Learning stages.

20. Which of the following is true of the MaxAge timer carried in Hello BPDUs? (Choose two.)

- A. **This timer defines how long the root bridge waits to send the Hello BPDUs.**
- B. **This timer defines how long any bridge should wait to attempt running the STP algorithm once Hello BPDUs stop being received.**
- C. **This timer defines how long a port remains in the listening and learning stages.**
- D. **The default setting for this timer is two seconds.**
- E. **The default setting for this timer is 15 seconds.**
- F. **The default setting for this timer is 20 seconds.**

Answer: B, F. Choices A and D are true of the Hello Time timer; C and E are true of the ForwardDelay timer.

21. During which of the following states does a port forward data frames?

- A. Listening, learning, and forwarding.
- B. Learning and forwarding.
- C. Listening and forwarding.
- D. Forwarding only.

Answer: D. The only time a port can forward data frames is when it's in Forwarding state.

22. Which of the following is true of EtherChannels? (Choose three.)

- A. EtherChannels are logical bundling of physical links.
- B. EtherChannels are a physical bundling of physical links.
- C. Links do not have to be trunked to part of an EtherChannel.
- D. Links must be trunked to be part of an EtherChannel.
- E. Up to ten physical links can belong to a single EtherChannel.
- F. STP considers an EtherChannel to be a single link, but a failure of one of the links making up the EtherChannel forces STP to recalculate the spanning tree.
- G. STP considers an EtherChannel to be a single link, and a failure of one of the links making up the EtherChannel will not result in STP recalculation.

Answers: A, D, G. An EtherChannel is a logical bundling of up to eight trunked, parallel Ethernet links. STP sees an EtherChannel as a single logical link, and a failure of one of the physical links will not result in STP recalculation.

23. Which of the following devices can route packets between VLANs? Choose two.

- A. Router
- B. Layer 2 Switch
- C. Multilayer Switch
- D. Hub
- E. Repeater
- F. Concentrator

Answers: A, C. Routers, multilayer switches, and Layer 3 switches can all route packets between VLANs. A simple Layer 2 switch cannot, nor can a hub, repeater, or concentrator.

24. HostA and HostB are in VLAN 10. HostC and HostD are in VLAN 20. Assuming the VLANs have been configured correctly but routing is not taking place, which of the following is true? Choose four.

- A. HostA can ping HostB.
- B. HostA cannot ping HostB.
- C. HostA cannot ping HostC.
- D. HostB can ping HostC.
- E. HostB can ping HostD.
- F. HostC can ping HostD.
- G. HostC cannot ping HostD.
- H. HostD cannot ping HostA.

Answer: A, C, F, H. Typically, hosts in the same VLAN are on the same subnet. Communication between VLANs requires routing take place. Hosts in the same VLAN will be able to ping each other, but hosts in different VLANs will not be able to do so.

25. Which of the following is true of trunking?

- A. ISL is a Cisco-proprietary protocol, and can therefore be used to configure trunking between non-Cisco devices.
- B. 802.1q is the industry standard, and can therefore be used to configure trunking between a non-Cisco device and a Cisco switch.
- C. ISL encapsulates the entire frame.
- D. 802.1q encapsulates the entire frame.
- E. ISL understands the concept of the native vlan.
- F. 802.1q understands the concept of the native vlan.

Answer: B, C, F. Answer "A" is half-true; ISL is a Cisco-proprietary protocol, which means it cannot be used for trunking between non-Cisco devices. ISL does encapsulate the entire frame, where 802.1q adds a header identifying the VLAN ID. ISL does not allow the use of native vlans.

26. Which is true of native vlans?
- A. ISL allows the use of native vlans. The default native VLAN is VLAN 1, and this can be changed.
 - B. ISL allows the use of native vlans. The default native VLAN is VLAN 1, and this cannot be changed.
 - C. 802.1q allows the use of native vlans. The default native VLAN is VLAN 1, and this can be changed.
 - D. 802.1q allows the use of native vlans. The default native VLAN is VLAN 1, and this cannot be changed.

Answer: C. 802.1q allows native vlans, the default is VLAN 1, and this can be changed with the interface-level command ***switchport trunk native vlan <x>***, where "x" is the desired native vlan.

27. You are running the IEEE 802.1q trunking protocol between two Cisco 2950 switches. One of your switches is receiving frame with no VLAN ID. What action will the switch take with these frames?

- A. The frames are corrupt. They will be discarded and an error message sent to the sending switch.
- B. These are 802.1q management frames; they will be flooded to other switches on VLAN 0, the Cisco management VLAN for 2950 switches.
- C. These frames are destined for the native vlan and will be forwarded accordingly.
- D. These frames will be queued while the receiving switch sends a message to the VTP server to see where the frame should be forwarded.

Answer: C. When running 802.1q, the switch will not put a VLAN ID in any frame that is destined for the current native VLAN. A switch receiving frames with no VLAN ID will assume the frames are meant for the native VLAN.

28. What are the three modes a switch running VTP can run in?

- A. Server**
- B. Master**
- C. Client**
- D. Transparent**
- E. Receiver**
- F. Read-Only**
- G. Hibernation**

Answer: A, C, D.

29. A VTP client receives an advertisement with a configuration revision number of 56. The client accepts the advertisement. Three minutes later, the client receives a VTP advertisement with a configuration revision number of 45. What will the client do with this advertisement?

- A. Accept it. VTP advertisements with lower numbers are always accepted.**
- B. Discard it. VTP advertisements with lower numbers are always discarded.**
- C. Send the advertisement back to the server.**
- D. Accept it, but send a VTP Query message to the server.**
- E. Discard it, but send a VTP Query message back to the server.**

Answer: B. A VTP client will only accept a VTP advertisement if the revision number is higher than the revision number of the last VTP advertisement it accepted. The revision numbers are incremented by the VTP server anytime VLAN configuration information is sent.

30. You need to add three VLANs to your network. You have two VTP Servers and ten VTP Clients, all in the same domain. What is the absolute minimum number of devices you need to manually add these new VLANs to?

- A. One – one VTP server.**
- B. Two – both VTP servers.**
- C. Two – one VTP server and one VTP client.**
- D. Three – both VTP servers and one VTP client.**

Answer: A. Configure the VLANs on one VTP server, and that server will send an advertisement to every VTP device in its domain, including the other VTP server.

31. Which of the following processes reduces broadcast propagation across a trunk?

- A. VTP Portfast**
- B. VTP Flood Guard**
- C. VTP Port Guard**
- D. VTP Pruning**
- E. VTP Encryption**

Answer: D. VTP pruning prevents broadcasts from being sent over a trunking line between two switches when the hosts for the broadcast's destination VLAN all reside on the sending switch. If the remote switch has no hosts needing the broadcast, there is no reason to send the broadcast over the link.

32. You have just put a brand-new Cisco 2950 switch on your network. You run **show vtp status** to check the default settings. Which of the following is true? (Choose two.)

- A. VTP Pruning is enabled.**
- B. The VTP password is "CISCO".**
- C. The switch is a VTP Server.**
- D. The switch is a VTP Client.**
- E. The VTP domain name is "NULL".**
- F. The VTP domain name is "VTP1".**
- G. The VTP domain name does not appear.**

Answer: C, E. The switch will be set to VTP Server mode, the default. The VTP domain name will be NULL. By default, VTP pruning is disabled, and VTP passwords will not be set.

33. Your network consists of three VTP servers, ten VTP clients, and one switch running in VTP transparent mode. You wish to add a VLAN that will be seen by all the VTP clients. Which of the following best describes where you can add this VLAN and achieve the desired result?

- A. A VTP Server only.**
- B. A VTP Server or a VTP client.**
- C. The switch running in VTP Transparent mode.**

D. You can add a VLAN to any of these clients.

Answer: A. You cannot add a VLAN to a VTP client. You can add a VLAN to a switch running transparent mode, but the VLAN would not be advertised to the other switches. Changes made to a switch in VTP Transparent mode affect only that particular switch.

34. You add a VLAN to a VTP server in VLAN Database mode. After adding the VLAN and using Ctrl-Z to save your change, you run **show vlan brief** to make sure the VLAN was added. You do not see the VLAN. What is the issue?

- A. To exit VLAN Database mode and save changes, you need to use CTRL-H.**
- B. To exit VLAN Database mode and save changes, you need to type "exit".**
- C. To exit VLAN Database mode and save changes, you need to type "reset".**
- D. To exit VLAN Database mode and save changes, you need to type "VTP leave".**

Answer: B. VLAN Database mode is different from other modes in that you must type "apply" or "exit" when leaving the mode in order for your changes to be saved. Entering "apply" will save the changes and bump the configuration revision number by 1 and will leave you in VLAN Database mode. "Exit" will do the same and exit VLAN Database mode. Leaving this mode with CTRL-Z will not save changes.

35. You wish to enable trunking between two Cisco 2950 switches. The switches are physically connected via port fast0/7 on both switches. On one switch, the port is in dynamic desirable mode; the other side is in trunking mode. What is the result?

- A. Because of the mismatched mode types, trunking cannot occur.**
- B. On Cisco 2950 switches, only ports fast0/1 through fast0/6 can trunk.**
- C. Trunking will occur.**
- D. Trunking will occur, but errors will occur when transmitting frames over the native vlan.**

Answer: C. If one port is in trunk mode, and the other is in dynamic desirable, trunking will occur. There is no limitation on a Cisco 2950 switch as to which FastEthernet ports can trunk.

36. You wish to enable trunking between two Cisco 2950 switches. The switches are physically connected via port fast 0/7 on both switches. On one switch, the port is in dynamic auto mode; the other side is in dynamic desirable mode. What is the result ?

- A. Trunking cannot occur until both ports are running dynamic desirable.
- B. Trunking will occur.
- C. Trunking cannot occur with both ports in any dynamic mode. They must be statically configured.
- D. Trunking cannot occur until both ports are in dynamic auto.

Answer: B. This is an acceptable combination of dynamic modes. Trunking will take place. Trunking cannot occur if both ports are in dynamic auto.

37. You wish to enable trunking between two Cisco 2950 switches. The switches are physically connected via port fast0/9 on both switches. Ports 0/9 on both switches are running in dynamic auto mode. Which statement is true?

- A. Trunking cannot occur, but will occur if one port is changed to dynamic desirable mode.
- B. Trunking cannot occur, and can occur only if both ports are changed to dynamic desirable.
- C. Trunking cannot occur, and can occur only if both ports are changed to trunking mode.
- D. Trunking will occur.

Answer: A. Trunking cannot occur between two ports that are both set to dynamic auto; this would result in both ports waiting for the other port to begin negotiating the trunk. Only one side needs to be set to dynamic desirable for trunking to take place.

38. You wish to enable trunking between two Cisco 2950 switches. The switches are physically connected via port /9 on both switches. SwitchA's port is in access mode; SwitchB's port is in dynamic auto mode. Which statement is true?

- A. Trunking will occur.
- B. Trunking can occur if SwitchB's port is changed to dynamic desirable.
- C. Trunking can occur if SwitchB's port is changed to trunking mode.
- D. Trunking will not occur, and cannot occur regardless of what changes are made to SwitchB.

Answer: D. A port placed in access mode cannot form a trunk under any circumstances, regardless of what mode the remote switch's port is in.

39. You are building a trunk between two Cisco 2950 switches. They are connected through port 0/4 on each switch. The port on SwitchA is set to dynamic auto mode. What mode(s) must the port on SwitchB be set to in order for trunking to take place?

- A. Dynamic auto, dynamic desirable, access, or trunk.
- B. Dynamic desirable, access, or trunk.
- C. Dynamic desirable or trunk.
- D. Dynamic desirable or dynamic auto.
- E. Dynamic desirable only.
- F. Dynamic auto only.
- G. Access only.

Answer: C. A port placed in dynamic auto mode will form a trunk if the port on the other end of the connection is placed in dynamic desirable or trunk. If the remote port was in dynamic auto, both ports would wait for the other to begin forming a trunk. No access port can ever become part of a trunk.

40. You are building a trunk between two Cisco 2950 switches. They are connected through port 0/4 on each switch. The port on SwitchA is set to dynamic desirable mode. What mode(s) must the port on SwitchB be set to in order for trunking to take place?

- A. Dynamic auto, dynamic desirable, access, or trunk.
- B. Dynamic auto, dynamic desirable, or trunk.
- C. Dynamic desirable, access, or trunk.
- D. Dynamic desirable or trunk.
- E. Dynamic desirable or dynamic auto.
- F. Dynamic desirable only.
- G. Dynamic auto only.
- H. Access only.

Answer: B. A port in dynamic desirable is actively seeking to trunk. The remote port can be in dynamic auto, dynamic desirable, or trunk. No access port can ever become part of a trunk.

LAN Switching Lab

With the command **vtp domain**, place both switches in the vtp domain CCNA. Set a password of CISCO for the domain. Enable pruning with the **vtp pruning** command.

Configuring the VTP domain CCNA, setting a password of CISCO for the domain, and enabling VTP pruning.

```
SW1#conf t
SW1(config)#vtp domain CCNA
Changing VTP domain name from NULL to CCNA
SW1(config)#vtp password CISCO
Setting device VLAN database password to CISCO
SW1(config)#vtp pruning
Pruning switched on

SW2#conf t
SW2(config)#vtp domain CCNA
Changing VTP domain name from NULL to CCNA
SW2(config)#vtp password CISCO
Setting device VLAN database password to CISCO
SW2(config)#vtp pruning
Pruning switched on
```

Run **show vtp status** on both routers to ensure they belong to the correct VTP domain.

```
SW1#show vtp status
VTP Version      : 2
Configuration Revision : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 5
VTP Operating Mode      : Server
VTP Domain Name        : CCNA
VTP Pruning Mode       : Enabled
```

```
SW2#show vtp status
VTP Version      : 2
Configuration Revision : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 5
VTP Operating Mode      : Server
VTP Domain Name        : CCNA
VTP Pruning Mode       : Enabled
```

By default, both switches are in VTP Server mode. With the **vtp mode client** command, put SW2 in vtp client mode. All VLANs created in this lab will now have to be created on SW1, the VTP Server. Verify the change with **show vtp status**.

```
SW2#show vtp status
VTP Version      : 2
Configuration Revision : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 5
VTP Operating Mode    : Client
VTP Domain Name      : CCNA
VTP Pruning Mode     : Enabled
```

SW2 is now a VTP client. VLANs cannot be created, modified, or deleted on this device.

R2 and R3 are connected to the switches. On R2 and R3, ensure the Ethernet interfaces are open, and run **show cdp neighbor** to see what switch ports the routers are connected to.

*Run **show interface Ethernet** to ensure the Ethernet interface is up, the line protocol is up, and to view the IP address assigned to the interface.*

```
R2#show interface ethernet0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0050.500f.3700 (bia 0050.500f.3700)
  Internet address is 172.23.23.2/27
```

```
R3#show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0010.7b1f.41c1 (bia 0010.7b1f.41c1)
  Internet address is 172.23.23.3/27
```

Both Ethernet interfaces are up. Run “show cdp neighbor” on both routers to determine the switch ports the routers are connected to.

```
R2#show cdp neighbor
Device ID      Local Intrfce   Holdtme  Capability Platform Port ID
SW1           Eth 0/0        154       S I        WS-C3550-2   Fas 0/1
```

```
R3#show cdp neighbor
Device ID      Local Intrfce   Holdtme  Capability Platform Port ID
SW1           Eth 0/0        140       S I        WS-C3550-2   Fas 0/2
```

These two ports will now be placed into VLAN 23. This vlan must be created on sw1, since that is the VTP server. Do so with the **vlan 23** command, and verify the vlan's creation on both switches with **show vlan brief**.

```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#vlan 23
SW1(config-vlan)#^Z
SW1#show vlan bri
00:40:13: %SYS-5-CONFIG_I: Configured from console by console
SW1#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11
23	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

```
SW2#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11
23	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

The VLAN was created on the VTP Server, SW1. "show vlan" verifies not only that the VLAN was successfully created on SW1, but that SW2 sees the newly created VLAN as well.

On switch1, place port 0/2 into access mode and then into vlan 23, as shown below. Verify with **show vlan brief**.

```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#int fast 0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 23
SW1(config-if)#^Z

SW1#show vlan brief

VLAN Name          Status    Ports
-----            -----
1    default        active   Fa0/1, Fa0/3, Fa0/4, Fa0/5
                           Fa0/6, Fa0/7, Fa0/8, Fa0/9
                           Fa0/10, Fa0/11
23                  active   Fa0/2
```

On switch2, do the same for port 0/3.

```
SW2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#int fast 0/3
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 23

SW2#show vlan brief

VLAN Name          Status    Ports
-----            -----
1    default        active   Fa0/1, Fa0/2, Fa0/4, Fa0/5
                           Fa0/6, Fa0/7, Fa0/8, Fa0/9
                           Fa0/10, Fa0/11
23                  active   Fa0/3
```

From R2, send a **ping** to R3's Ethernet interface. From R3, send a **ping** to R2's Ethernet interface. The pings succeed. But how do the

```
R2#ping 172.23.23.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.23.23.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

```
R3#ping 172.23.23.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.23.23.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

Exclamation points indicate connectivity between the local device and the IP address that was “pinged” exists.

To see what trunks are running on a 2950, run the command **show interface trunk**. By doing so on either switch, you will see there is a trunk running across port fast0/12. The two ports are connected by a crossover cable, and since the ports are in dynamic desirable by default, a trunk formed without manual configuration.

```
SW2#show int trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/12	desirable	802.1q	trunking	1

Port	Vlans allowed on trunk
Fa0/12	1-4094

Port	Vlans allowed and active in management domain
Fa0/12	1,23

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/12	1,23

On switch1, enter vlan database mode with the **vlan database** command. Change the name of VLAN 23 to VLAN_A with the **vlan name** command. Run **show vlan brief** to verify the name change.

Renaming a VLAN in “vlan database” mode.

```
SW1#vlan database  
SW1(vlan)#vln 23 name VLAN_A
```

VLAN 23 modified:

Name: VLAN_A

```
SW1(vlan)#exit
```

APPLY completed.

Exiting....

```
SW1#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/3, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/21, Fa0/23, Fa0/24, Gi0/1 Gi0/2
23 VLAN_A	active	Fa0/1, Fa0/2

To save changes, vlan database mode MUST be exited with “exit” or “apply”. CTRL-Z, used to exit a mode and save changes on a Cisco router, will NOT save changes made in vlan database mode. The router will not send a message that changes were not saved.

Here, CTRL-Z is used to exit vlan database mode after changing the VLAN name again. “show vlan brief” verifies that the name change was not saved.

```
SW1#vlan database
```

```
SW1(vlan)#vln 23 name cisco
```

VLAN 23 modified:

Name: cisco

```
SW1(vlan)#^Z
```

```
SW1#
```

```
SW1#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/3, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/21, Fa0/23, Fa0/24, Gi0/1 Gi0/2

On switch1, view the spanning tree information for VLAN 23 with the **show spanning tree vlan 23** command.

```
SW1#show spanning-tree vlan 23

VLAN0023
  Spanning tree enabled protocol ieee
  Root ID  Priority  32791
            Address  0009.b738.9180
  This bridge is the root
    Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID Priority  32791 (priority 32768 sys-id-ext 23)
    Address  0009.b738.9180
    Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
    Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Desg	FWD	100	128.2	Shr
Fa0/11	Desg	FWD	19	128.11	P2p
Fa0/12	Desg	FWD	19	128.11	P2p

```
SW2#show spanning-tree vlan 23

VLAN0023
  Spanning tree enabled protocol ieee
  Root ID  Priority  32791
            Address  0009.b738.9180
            Cost      19
            Port      19 (FastEthernet0/19)
    Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Bridge ID Priority  32791 (priority 32768 sys-id-ext 23)
  Address  000a.8a4b.fb00
  Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
  Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/11	Root	FWD	19	128.11	P2p
Fa0/12	Altn	BLK	19	128.12	P2p

SW1 is the root bridge. Recall that the lowest BID will win the root bridge election. Both bridges have the same priority; since the BID is a concatenation of the priority and MAC address, the device with the lowest MAC address will be the root bridge.

Look under the BridgeID on both switches. The highlighted address is that switch's MAC address. The first four bits of the MAC address on SW1 are 0009, where the first four bits of SW2's MAC are 000a. MAC addresses are expressed in hex, and since "a" in hex represents 10, SW1 will have the lower MAC address and is therefore elected the root bridge.

The default behavior of the root bridge is that all ports will be in forwarding mode, which is exactly what is happening on SW1. On SW2, one port is the root port and is in forwarding mode. The other port is placed into blocking mode.

```
SW2#conf t
SW2(config)#spanning-tree vlan 23 root primary
SW2#show spanning vlan 23
VLAN0023
  Spanning tree enabled protocol ieee
  Root ID  Priority  24599
    Address  000a.8a4b.fb00
    This bridge is the root
    Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Bridge ID Priority  24599 (priority 24576 sys-id-ext 23)
  Address  000a.8a4b.fb00
  Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
  Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/11	Desg	FWD	19	128.11	P2p
Fa0/12	Desg	FWD	19	128.12	P2p

*The **root primary** command adjusted the priority of SW2 to 24599, more than enough to become the new root bridge. In the concatenation of the priority and MAC address that creates the BID, the priority comes first, so a BID with a lower priority will always become the root bridge, regardless of what the MAC addresses are.*

Both ports are now in Forwarding mode, as is expected of the root bridge.

```

SW1#show spanning-tree vlan 23

VLAN0023
Spanning tree enabled protocol ieee
Root ID Priority 24599
    Address 000a.8a4b.fb00
    Cost 19
    Port 19 (FastEthernet0/19)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32791 (priority 32768 sys-id-ext 23)
    Address 0009.b738.9180
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300

Interface      Role Sts Cost      Prio.Nbr Type
-----
Fa0/2          Desg FWD 100      128.2  Shr
Fa0/11         Root FWD 19      128.19 P2p
Fa0/12         Altn BLK 19      128.20 P2p

Switch1 is no longer listed as the root bridge, and one of the two ports connected to Switch2 has gone to blocking mode.

```

On SW1, configure PortFast on the port leading to R2 with **spanning portfast**, and note the warning the router displays. Remove PortFast with **no spanning portfast**.

```

SW1#conf t
SW1(config)#interface fast 0/11
SW1(config-if)#spanning portfast

%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/11 but will only
have effect when the interface is in a non-trunking mode.
SW1(config-if)#no spanning portfast

```

View the MAC addresses of R2 and R3 with **show mac-address-table dynamic vlan 23**. Configure port-based authentication on the ports connected to R2 and R3. On those ports, configure the command **switchport port-security**, and configure the MAC address of the connected device as the only MAC address that will be allowed to connect to this port.

```
SW1#conf t
SW1(config)#interface fast 0/2
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security maximum 1 <the default>
SW1(config-if)#switchport port-security mac 0050.500f.3700
```

Run “show port-security” to display the ports that port-security is active on. Note that the SecurityViolation default setting is “shutdown”, meaning the port will be shut down if a device with a different MAC address attempts to connect to this port.

SW1#show port-security

Secure Port	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Action
Fa0/2	1	1	0	Shutdown

SW2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

```
SW2(config)#int fast 0/3
SW2(config-if)#switchport port-security
SW2(config-if)#switchport port-security max 1
SW2(config-if)#switchport port-security mac 0010.7b81.bc87
SW2(config-if)#{^Z}
```

SW2#show port-sec

Secure Port	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Action
Fa0/3	1	1	0	Shutdown

Combine the two physical connections between the two switches into one logical connection by creating an EtherChannel. On each of the ports physically connected to the other switch, run **channel-group 1 mode on**.

```
SW1#conf t
SW1(config)#interface fast 0/11
SW1(config-if)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1
03:37:59: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
SW1(config)#interface fast 0/12
SW1(config-if)#channel-group 1 mode on

SW2#conf t
SW2(config)#interface fast 0/11
SW2(config-if)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1
03:38:11: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
SW2(config-if)#interface fast 0/12
SW2(config-if)#channel-group 1 mode on
```

Run “show spanning vlan 23” on R1 and note the difference from earlier in the lab.

```
SW1#show spanning vlan 23
```

```
VLAN0023
Spanning tree enabled protocol ieee
Root ID Priority 24599
    Address 000a.8a4b.fb00
    Cost 12
    Port 65 (Port-channel1)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
Bridge ID Priority 32791 (priority 32768 sys-id-ext 23)
    Address 0009.b738.9180
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Po1	Root	FWD	12	128.65	P2p

The two physical connections are now seen as one logical connection, port-channel 1 (Po1).

Section Three: Frame Relay

Frame Relay is one of the most commonly used Layer 2 technologies. Any WAN you ever work with is most likely going to have Frame Relay in there somewhere, and you've got to know how to work with it and how to troubleshoot it.

I advise you to practice Frame Relay in a lab environment, especially the Frame Relay debugs.

*Chris Bryant
CCIE #12933*

Commands Used In This Chapter And Labs Include:

Debug frame lmi -- Used to detect LMI type a router is using. Mismatched LMIs lead to Frame Relay failures.

Encapsulation frame relay – Enables Frame Relay encapsulation on an interface.

<no> frame-relay inverse-arp – Frame Relay ARP performs a dynamic DLCI – IP Address mapping function that may not always be desired. To prevent dynamic mapping, run the **no frame-relay inverse-arp** command.

Frame-relay map – Used to configure manual DLCI – IP Address statements.

Frame-relay lmi-type – Used to manually configure an interface's LMI type.

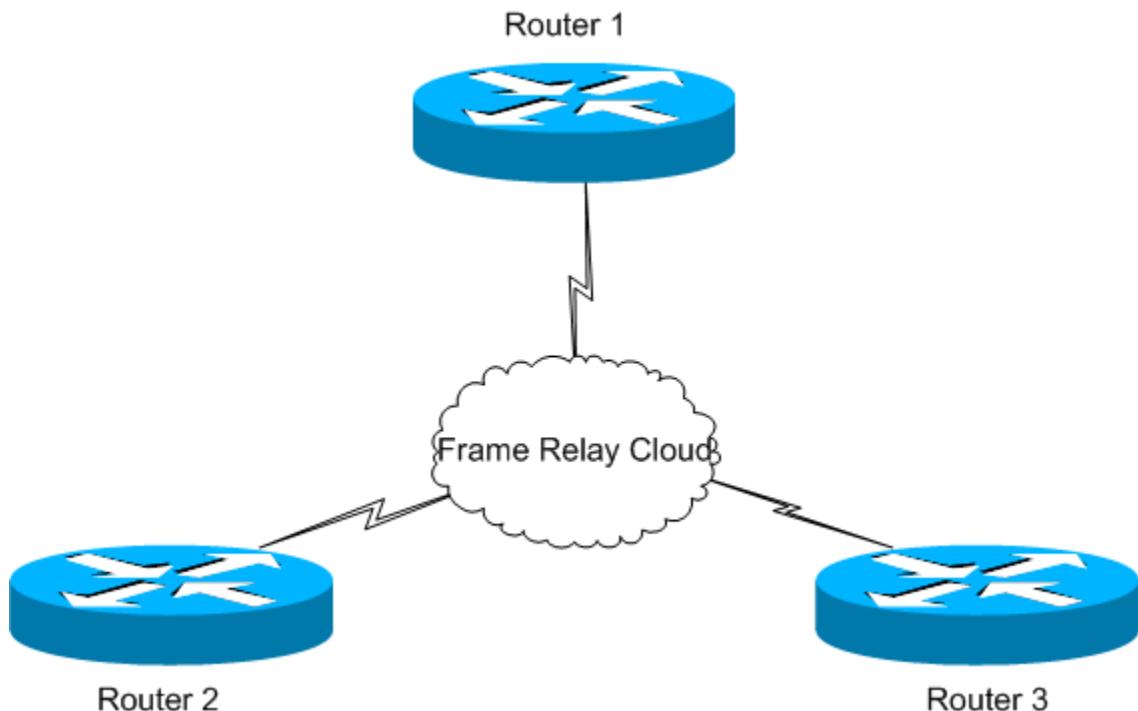
Show frame-relay lmi – Displays number of LMIs that have been received and sent, and status timeouts that may lead to a Frame Relay connectivity issue.

Show frame-relay map – Displays dynamic and static DLCI – IP address mappings.

Show frame-relay pvc – Displays locally configured DLCIs, how long they have been up, and the last time the PVC status changed.

Frame Relay is a Layer 2 connectivity method that delivers one major benefit over point-to-point links: **cost**.

Frame Relay uses **virtual circuits**, either permanent or on-demand, which is a logical path between two DTEs. There will not be a direct physical connection between the two DTEs. Instead, a logical connection is formed through a path of DCEs. This path is referred to as a **frame relay cloud**.



Many users will share the same frame relay cloud. The frame relay service provider guarantees a certain amount of bandwidth will be available to a given user at any time. The more guaranteed bandwidth desired, the more it costs, but it's still cheaper than a dedicated point-to-point link. This guaranteed bandwidth is referred to as the **committed information rate (CIR)**.

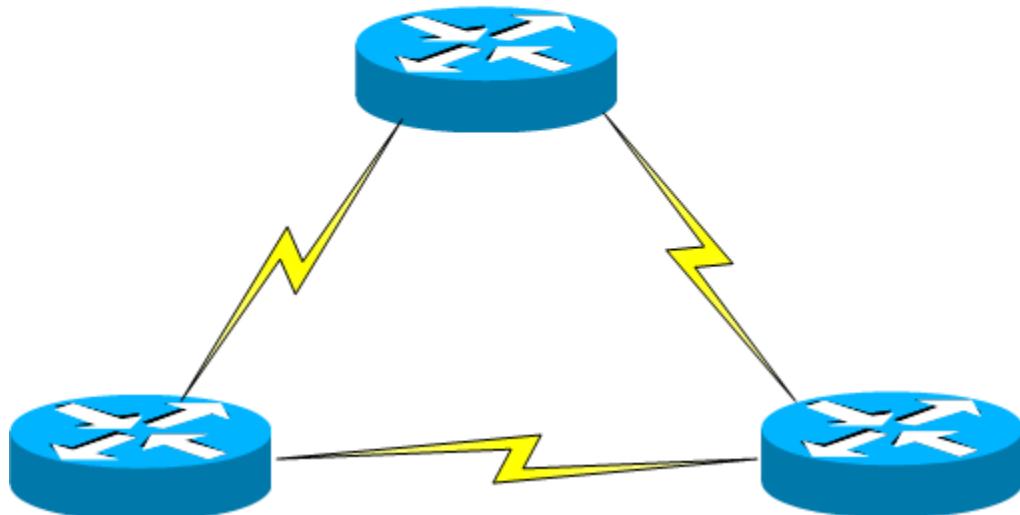
There are two types of virtual circuits. A permanent virtual circuit (PVC) is available at all times, where a switched virtual circuit (SVC) is up only when certain criteria are met. The configuration of a Frame Relay SVC is beyond the scope of the CCNA exam and will not be discussed here.

In The REAL World...

Frame Relay SVCs are used in production networks, but they're rare. Why? Because PVCs are so cheap. It's important to know SVCs exist, though.

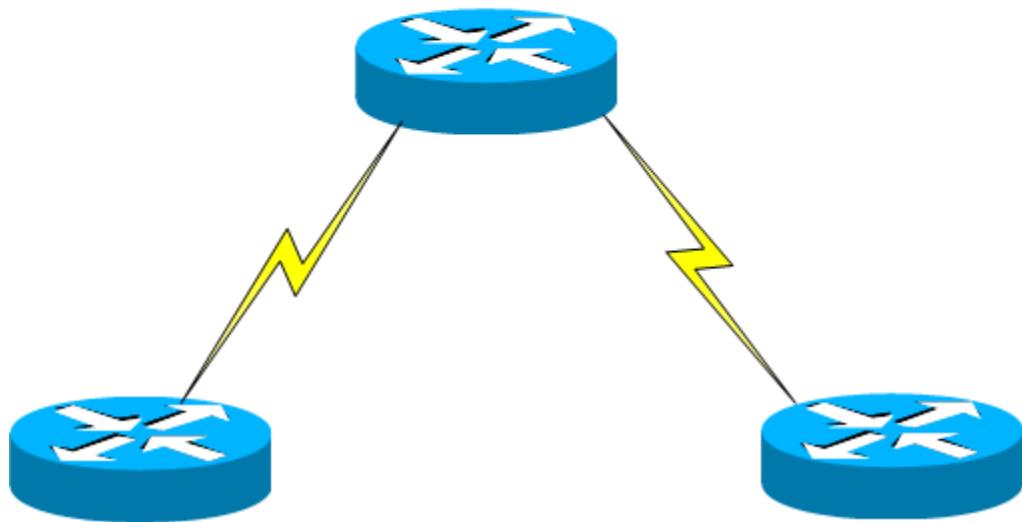
A PVC can be **full-mesh** or **partial-mesh**. A full mesh describes a topology where every router has a logical connection to every other router in the frame relay network.

A Frame Relay Full-Mesh Network:
Each router has a logical connection
to every other router.



More common is the partial-mesh configuration, where a single router (the **hub**) has a logical connection to every other router (the **spokes**), but the spokes do not have a logical connection to each other. Communication between spokes will go through the hub.

A Frame-Relay Partial-Mesh Network:
Each spoke router connects to the
hub router. Spoke routers do not
communicate directly, but through the
hub router.



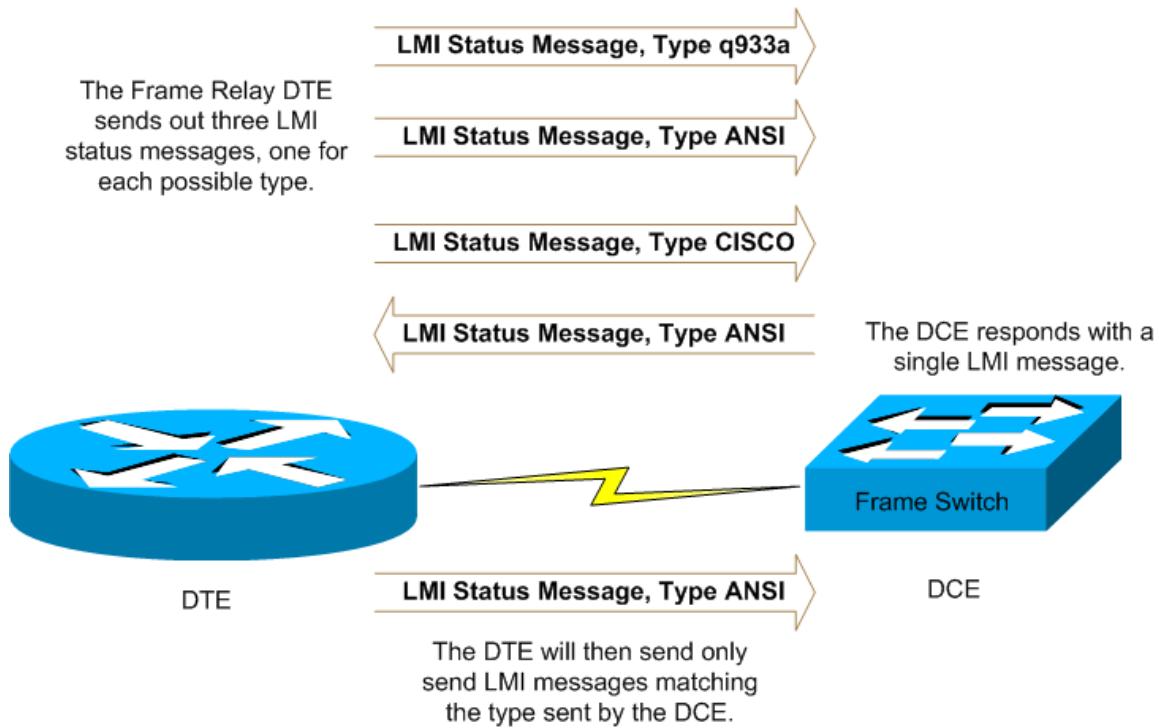
The LMI: The Heartbeat Of Frame Relay

The Local Management Interface (LMI) messages are sent between the DCE, typically the service provider, and the DTE, the Cisco router. LMI Status messages serve as keepalives for the frame connection. If keepalives are not continually received by both the DCE and DTE, the frame connection will drop. The LMI also indicates the PVC status to the router, reflected as either active or inactive.

The LMI types must match on the DTE and DCE for the PVC to be established. There are three types of LMI: Cisco, ansi, and q933a.

Cisco routers feature LMI Autosense, where the router will send out an LMI Status message for all three LMI types. The router then waits for a response for one of those LMI types from the DCE, and then sends out its own LMI message to match that LMI type.

The Frame Relay LMI Autosense Process.



In The REAL World...

LMI Autosense sounds great, doesn't it? Just plug and play. Except for one small detail: *it doesn't always work*. Get in the habit now of hard-coding your LMI types. This skill will come in handy as you progress up the Cisco certification ladder, and on the job as well.

Encapsulation Types

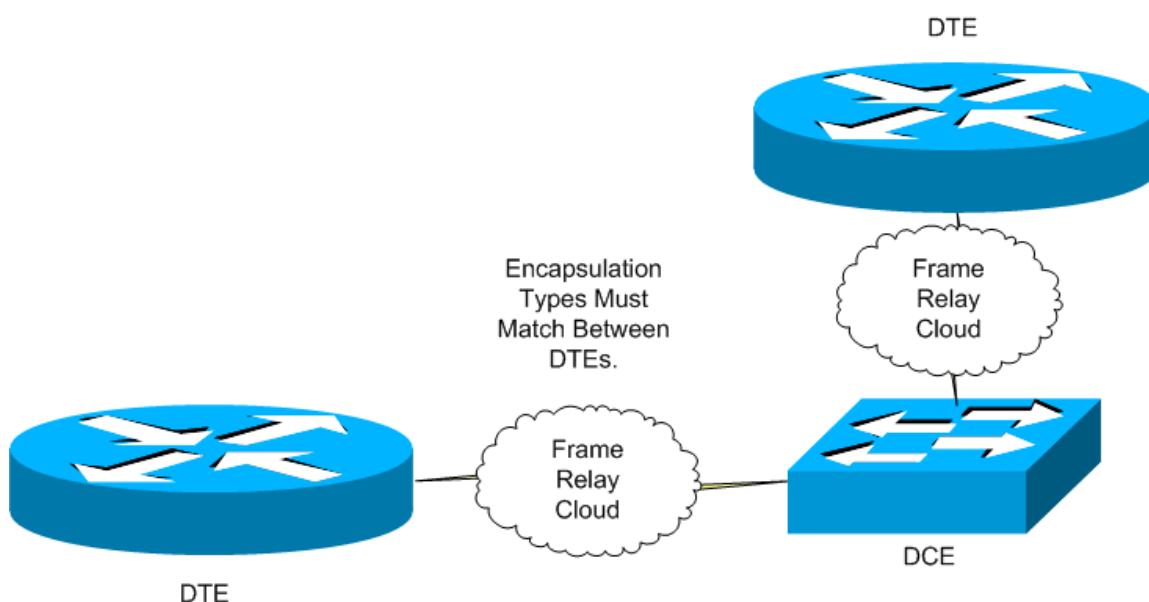
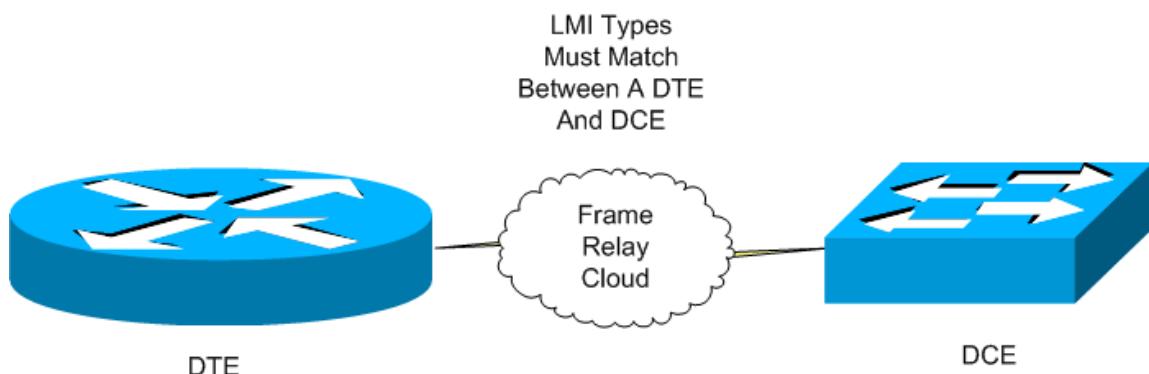
Where LMI types must match between the DCE and DTE, the Frame encapsulation type must match only between the DTEs. The DCEs do not care about the encapsulation type. VCs on the same router can use different encapsulation types. The Cisco-proprietary encapsulation type is **cisco**, and the other choice is **ietf**. The default type is **cisco**, but this is not listed in IOS help:

```
R1#conf t
R1(config)#interface serial 1
R1(config-if)#encapsulation frame-relay ?
  ietf Use RFC1490/RFC2427 encapsulation
```

*The default frame relay encapsulation type, **cisco**, is not listed in IOS Help.*

Keeping track of which devices must agree on what can be a little confusing at first. This chart sums it up:

Topic	Choices	Who Must Agree
Encapsulation Type	CISCO, IETF	Both Routers
LMI Type	CISCO, ANSI, Q933A	Both Routers and their Neighboring Frame Relay Switch.



DLCI Addressing

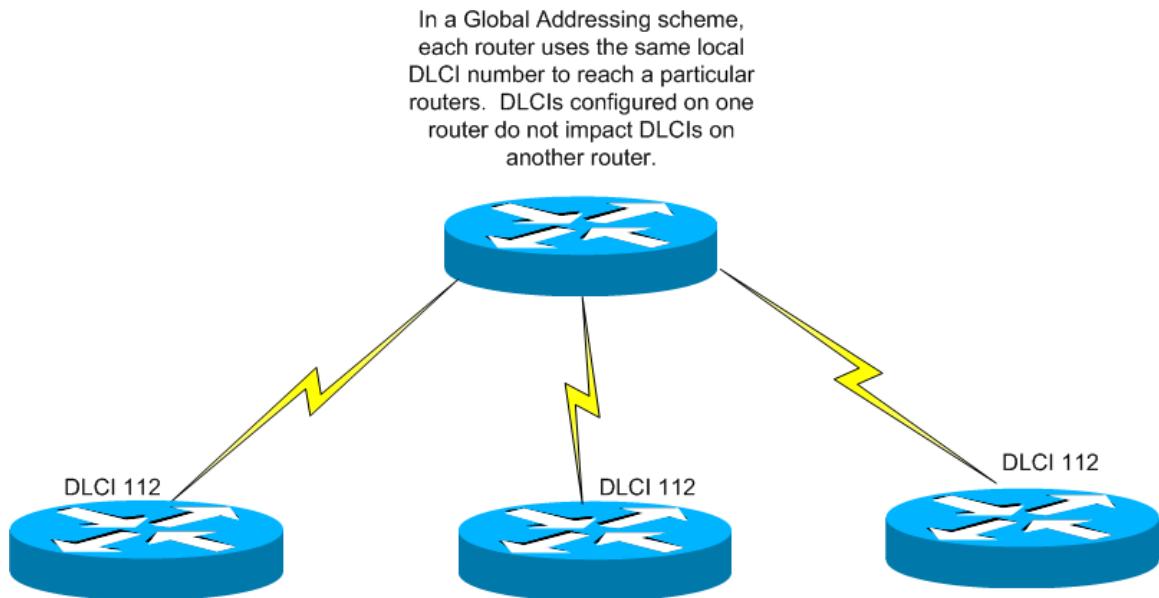
Frame Relay VCs use Data-Link Connection Identifiers (DLCIs) as their addresses. Unlike other Cisco technologies, VCs have only a single DLCI in their header. They do not have a source and destination.

The reason is that DLCIs have **local significance only**. The same DLCI can be used on different access links in the same network, but the same DLCI cannot be used on multiple access links on the same router.

Local Significance vs. Global Addressing

Cisco uses the term *global addressing* to describe a technique by which a router in a Frame network is reached via the same DLCI number from each router in the network. For example, in a 250-router network, the same DLCI number would be used to reach “Router A” by each router.

Global Addressing is an organizational tool that does **not** affect the fact that DLCIs have local significance only.



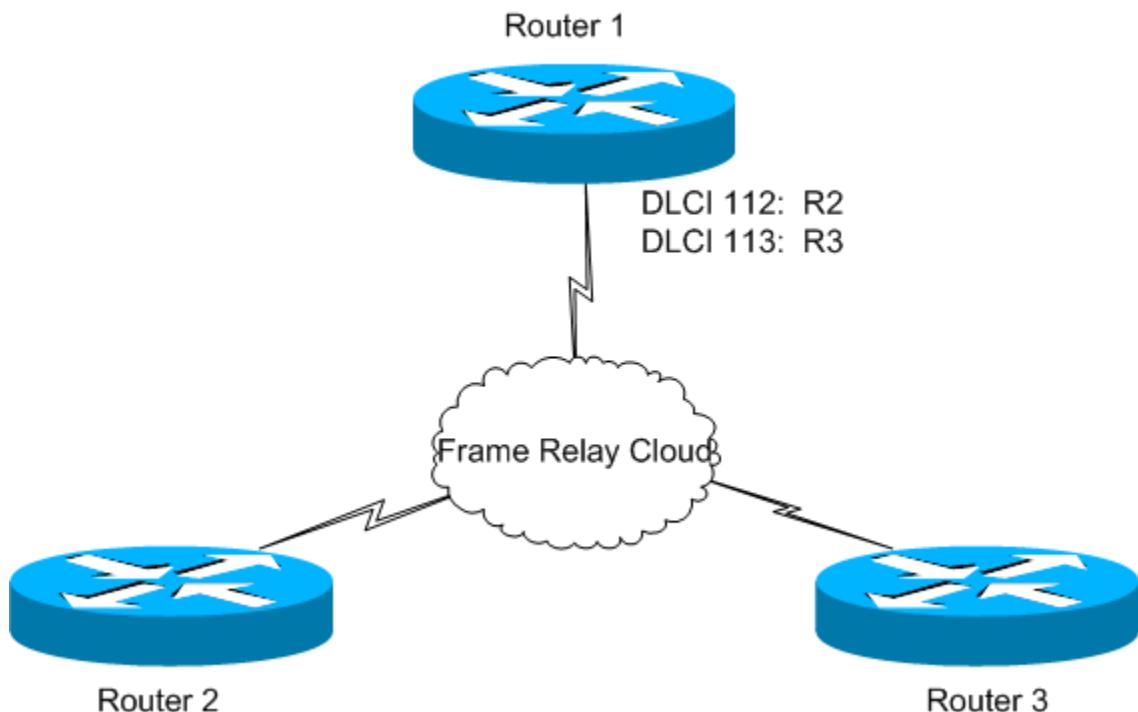
DLCI – to – IP Address Mapping

The locally significant DLCI must be mapped to the destination router's IP address. There are two options for this, Inverse ARP and static mapping.

What is ARP? What is Inverse ARP?

Address Resolution Protocol (ARP) is used by a router when the router knows the Layer 3 address (the IP address), but not the Layer 2 address (the MAC address). ARP dynamically maps that Layer 3 address to the Layer 2 address.

Inverse ARP performs the opposite function. In this case, the Layer 2 DLCI address is known, but the Layer 3 IP address is unknown. The Layer 3 addresses are learned once the PVC is up. The DLCI-IP address mapping then takes place.



In both the following examples, the single physical Serial interface on Router 1 is configured with two logical connections through the frame relay cloud, one to Router 2 and one to Router 3.

Inverse ARP runs by default once Frame Relay is enabled. By running **show frame-relay map** after enabling Frame Relay, two dynamic mappings are shown on this router. If a dynamic mapping is shown, Inverse ARP performed it.

Dynamic frame mappings are performed by Inverse ARP, enabled by default on a frame relay interface.

```
R1#show frame map
Serial0 (up): ip 200.1.1.2 dlci 122(0x7A,0x1CA0), dynamic,
               broadcast,, status defined, active
Serial0 (up): ip 200.1.1.3 dlci 123(0x7B,0x1CB0), dynamic,
               broadcast,, status defined, active
```

Static mappings require the use of a **frame map** statement. To use static mappings, turn Inverse ARP off with the **no frame-relay inverse-arp** statement, and configure a frame map statement for each remote destination that maps the local DLCI to the remote IP address. Frame Relay requires the **broadcast** keyword to send broadcasts to the remote device.

```
R1#conf t
R1(config)#interface serial0
R1(config-if)#no frame-relay inverse-arp
R1(config-if)#frame map ip 200.1.1.2 122 broadcast
R1(config-if)#frame map ip 200.1.1.3 123 broadcast
```

The syntax of the frame map statement maps the remote IP address to the local DLCI. Broadcasts will not be transmitted by default; the broadcast option must be configured.

```
R1#show frame map
Serial0 (up): ip 200.1.1.2 dlci 122(0x7A,0x1CA0), static,
               broadcast,
               CISCO, status defined, active
Serial0 (up): ip 200.1.1.3 dlci 123(0x7B,0x1CB0), static,
               broadcast,
               CISCO, status defined, active
```

The "static" status of the mapping indicates it was configured manually.

In The REAL World...

Like LMI Autosense, Inverse ARP sounds like a great idea, and doesn't always work well. You're much better off disabling Inverse ARP whenever possible and configuring static mappings. For exam purposes, know how to do both. As you progress up the Cisco certification ladder toward the CCIE, you'll find that disabling Inverse ARP becomes a habit.

Notes

Frame Relay Q & A

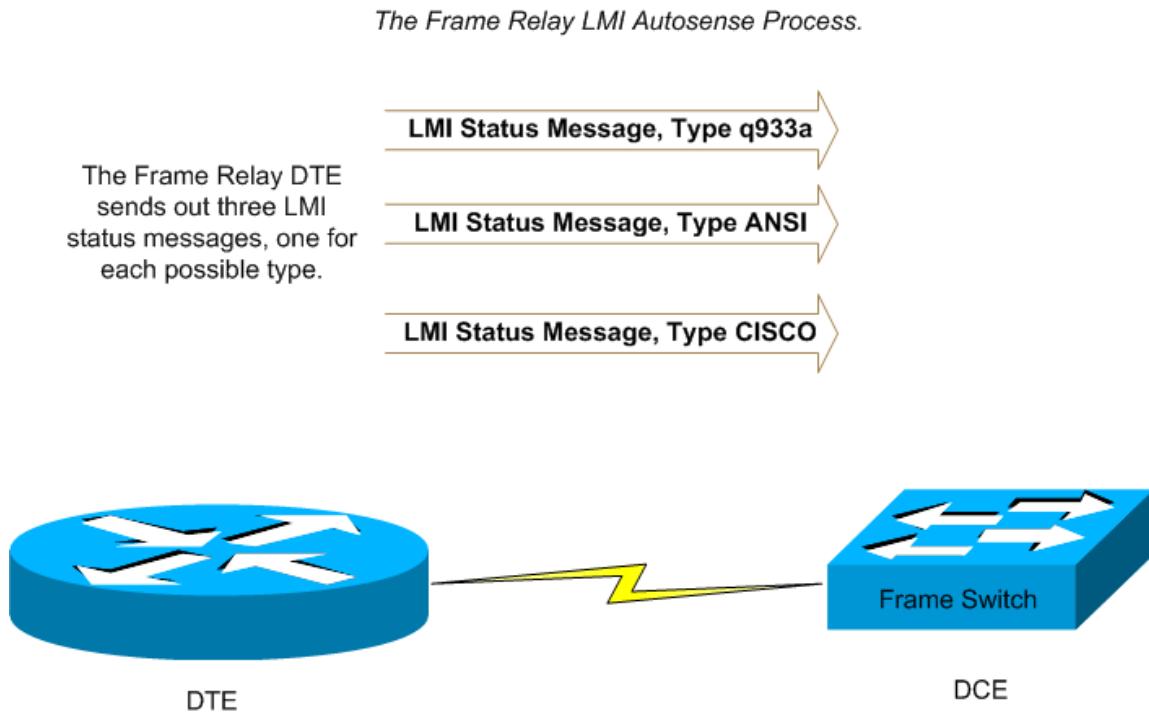
1. What statements are true of a Frame Relay "cloud"?
 - A. **A Frame Relay "cloud" generally consists of logical connections.**
 - B. **A Frame Relay "cloud" generally consists of physical connections.**
 - C. **The routers on the two sides of the cloud are DTEs.**
 - D. **The routers on the two sides of the cloud are DCEs.**
 - E. **The devices in the cloud itself are DTEs.**
 - F. **The devices in the cloud itself are DCEs.**
- ANSWER: A, C, F. A Frame Relay "cloud" consists of logical connections that pass through DCEs. The routers on both sides of the cloud are DTEs.
2. What statements are true of the Local Management Interface (LMI)? Choose three.
 - A. **Only the DCE is concerned with the LMI.**
 - B. **Only the DTE is concerned with the LMI.**
 - C. **Both the DCE and DTE are concerned with the LMI.**
 - D. **The LMIs are exchanged between the DCE and DTE.**
 - E. **The LMIs are exchanged between the two DTEs.**
 - F. **If LMIs are not received, the PVC will remain intact, but frames cannot be transmitted over that PVC.**
 - G. **If LMIs are not received, the PVC will go down.**
 - H. **The LMIs are status messages that do not affect the PVC.**

ANSWER: C, D, G. LMI messages serve as the "heartbeat" of Frame Relay. The LMIs are exchanged between the DTE and DCE; they do not travel all the way across a Frame Relay cloud.

3. Which of the following are LMI types? (Choose three.)
 - A. **cisco**
 - B. **ansi**
 - C. **frame**
 - D. **DTE**
 - E. **DCE**
 - F. **Q933a**

ANSWER: A, B, F. Cisco, ansi, and q933a are the LMI types.

4. Consider the following diagram.



LMI Autosense is in action. How many LMI status messages will the frame switch send back?

- A. Zero – LMI Autosense can't be used in this situation.
- B. One.
- C. Three.
- D. Four – the LMI types will be returned along with a negotiation packet.

ANSWER: B. The sending device will send three LMI status messages, one of each type. The Frame Switch will return only one LMI status message, indicating to the router what LMI type will be used.

5. Which of the following is true of Frame Relay encapsulation types? Choose two.

- A. The default is cisco.
- B. The default is ietf.
- C. The default is q933a.
- D. Only the DTEs care about the encapsulation type, and it must match on both sides of the cloud.
- E. Only the DCEs care about the encapsulation type, and it must match on all DCEs throughout the cloud.
- F. Both the DTEs and DCEs care about the encapsulation type, and it must match throughout the entire transmission.

ANSWER: A, D. "cisco" is the default encapsulation type, with the other available type being "ietf". Only the DTEs care about the encapsulation type, and it must match on both. Where the DCEs do care about the LMI type in use, they don't care about the encapsulation type.

Topic	Choices	Who Must Agree
Encapsulation Type	CISCO, IETF	Both Routers
LMI Type	CISCO, ANSI, Q933A	The router and its neighboring Frame Relay switch.

6. Which of the following is true of a Frame Relay DLCI?

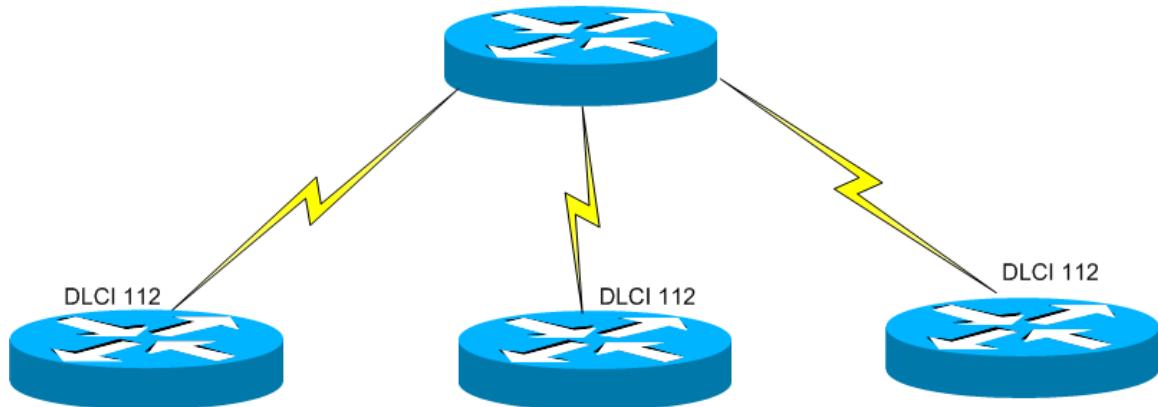
- A. "DLCI" stands for Data-Link Connection Identifier.
- B. "DLCI" stands for Data-Link Configuration Identifier.
- C. DLCIs have significance only on the router on which they are configured.
- D. DLCIs have significance throughout the cloud; no other router can use the same DLCI, hence the term "global addressing".
- E. Virtual Circuits have one DLCI only.
- F. Virtual Circuits have two DLCIs – one for the source and one for the destination.

ANSWER: A, D, E. DLCI stands for Data-Link Connection Identifier. DLCIs do have local significance only; a DLCI number can be used on as many different routers as you like. Virtual Circuits carry only the single DLCI number.

7. When configuring static mappings in Frame Relay, which of the following should be used? (Choose two.)

- A. The local IP address.**
- B. The remote IP address.**
- C. The local DLCI number.**
- D. The remote DLCI number.**
- E. The "broadcast" keyword.**
- F. Interesting traffic should be defined, so the PVC doesn't stay up when not needed.**

Answer: B, C. The remote IP address is mapped to the local DLCI number.



8. The three spoke routers are using the same DLCI number to reach the hub. What term best describes this numbering scheme?

- A. Misconfiguration – you can't use this numbering scheme.**
- B. Global Addressing**
- C. Local Significance**
- D. DLCI Multiplier**

E. Hub-and-spoke

ANSWER: B. The term “global addressing” refers to a DLCI numbering scheme where the remote routers all use the same DLCI number to reach a given destination. While this is a hub-and-spoke network, that term does not describe the numbering scheme.

9. You have decided not to configure manual frame mapping statements on your Frame Relay routers. What is true of the protocol that will perform this mapping dynamically?

- A. ARP will do this, and it's enabled by default.**
- B. ARP will do this, but it must be manually enabled.**
- C. Inverse ARP will do this, and it's enabled by default.**
- D. Inverse ARP will do this, but it must be manually enabled.**

ANSWER: C. Inverse ARP runs automatically when Frame Relay is enabled on an interface.

10. You have decided to manually map all of your Frame Relay DLCIs. Which of the following commands should you run before doing so?

- A. enable frame manual**
- B. no frame arp**
- C. no frame inverse-arp**
- D. enable frame arp**
- E. enable frame inverse-arp**

ANSWER: C. Inverse ARP will attempt to perform dynamic Frame Relay mappings when Frame Relay is enabled on the interface. When configuring static Frame Relay mappings, it's advisable to turn this default behavior off with “no frame inverse-arp”.

11. You have been called in to troubleshoot a Frame Relay connection. The PVCs are up, but routing protocol behavior on the PVCs has been unstable. You look at the configuration and see the following:

```
interface serial0
  no frame-relay inverse-arp
  frame map ip 200.1.1.2 122
  frame map ip 200.1.1.3 123
```

What is most likely the problem?

- A. Inverse ARP should not have been disabled.
- B. The "broadcast" keyword is missing.
- C. The "protocol" keyword is missing.
- D. The "pvc" keyword is missing.

ANSWER: B. Broadcasts will only travel over a Frame Relay network's static mappings if the "broadcast" keyword is present in the mapping statement. Leaving this out will prevent routing protocols that use broadcasts from working properly over the Frame Relay cloud.

```
R1#show frame map
```

```
Serial0 (up): ip 200.1.1.2 dlci 122(0x7A,0x1CA0), static,
               broadcast,
               CISCO, status defined, active
Serial0 (up): ip 200.1.1.3 dlci 123(0x7B,0x1CB0), static,
               broadcast,
               CISCO, status defined, active
```

12. Consider the above console output. Which of the following is false? Select all that apply.

- A. The Serial0 interface has been configured with two IP addresses, 200.1.1.2 and 200.1.1.3.
- B. Inverse ARP determined these mappings.
- C. Inverse ARP did not determine these mappings.
- D. The IP addresses shown are remote IP addresses.
- E. The DLCIs shown are remote DLCI numbers.

ANSWER: A, B, E. Note the question asked which statements are FALSE. Watch this carefully when you're passing the CCNA exam.

The IP addresses in the mapping statements are the addresses of the remote routers, so A and E are both false. B is false; the keyword "static" indicates that these mapping are manually configured. C and D are both true.

13. You are configuring Frame Relay on interface Serial0. You have just enabled Frame Relay encapsulation, as shown:

```
R1#conf t  
R1(config)#interface serial0  
R1(config-if)#encapsulation frame-relay
```

At this point in the configuration, what is true?

- A. Inverse ARP is enabled.
- B. Inverse ARP is disabled.
- C. The Frame Relay encapsulation type is "cisco".
- D. The Frame Relay encapsulation type is "ietf".
- E. The Frame Relay encapsulation type is "lmi".

ANSWER: A, C. By default, at this point in the configuration, Inverse ARP is enabled. The Frame Relay default encapsulation type is "cisco".

Notes

Frame Relay Lab

A hub-and-spoke Frame Relay network will now be configured, with R1 serving as the hub and R2 and R3 as the spokes. First, configure Frame Relay on R1's Serial0 interface with **encapsulation frame-relay**, and disable dynamic mapping with **no frame-relay inverse-arp**. After doing so, run **show frame map** on R1; no mappings should appear.

```
R1#conf t  
R1(config)#interface serial0  
R1(config)#ip address 172.12.123.1 255.255.255.0  
R1(config-if)#encapsulation frame-relay  
R1(config-if)#no frame-relay inverse-arp
```

```
R1#show frame map
```

```
R1#
```

If nothing appears after running “show frame map”, as shown here, no maps exist.

Configure two Permanent Virtual Circuits (PVC) on R1 with two **frame map** statements, mapping DLCI 122 to R2 and DLCI 123 to R3. Ensure that broadcasts will be sent over these virtual circuits with the **broadcast** keyword. Run **show frame map** after doing so.

Configuring frame map statements on the hub router.

```
R1#conf t  
R1(config)#interface serial0  
R1(config-if)#frame map ip 172.12.123.2 122 broadcast  
R1(config-if)#frame map ip 172.12.123.3 123 broadcast  
  
R1#show frame map  
Serial0 (up): ip 172.12.123.2 dlci 122(0x7A,0x1CA0), static,  
broadcast,  
CISCO, status defined, inactive  
Serial0 (up): ip 172.12.123.3 dlci 123(0x7B,0x1CB0), static,  
broadcast,  
CISCO, status defined, inactive
```

The mappings are inactive because frame-relay has not yet been configured on the remote routers R2 and R3.

R2's Serial interface will be configured as a multipoint interface. Remove any IP address from the interface as shown, and to prevent

dynamic mappings, disable Inverse ARP. Create a multipoint interface 0.123, and configure it with two frame map statements, one to reach R1 and one to reach R3. Both statements will use DLCI 221.

```
R2#conf t
R2(config)#interface serial0
R2(config-if)#no ip address
R2(config-if)#encapsulation frame-relay
R2(config-if)#no frame inverse-arp

R2(config-if)#interface s0.123 multipoint
R2(config-subif)#ip address 172.12.123.2 255.255.255.0
R2(config-subif)#frame map ip 172.12.123.1 221 broadcast
R2(config-subif)#frame map ip 172.12.123.3 221
```

A logical Serial interface can be either multipoint or point-to-point. When using a multipoint interface on a frame relay network, frame map statements are used just as they are on a physical interface. Enabling frame relay and disabling or enabling Inverse ARP are still done on the physical interface.

Note that the frame map statement for 172.12.123.3 does not include a broadcast statement. Routers do not forward broadcasts, so R1 would not forward a broadcast from R2 to R3. Therefore, there is no reason to send them.

In The Real World...

A lot of frame relay configurations in real-world networks, as well as most books on the subject, have “broadcast” in every single frame map statement, regardless of whether such a broadcast can ever reach the final destination. This isn’t necessarily wrong, but it is a waste of bandwidth and resources. Why send broadcasts toward a destination that will never see them?

Run **show frame map** on R2.

```
R2#show frame map

Serial0.123 (up): ip 172.12.123.1 dlci 221(0xDD,0x34D0), static,
    broadcast,
    CISCO, status defined, active
Serial0.123 (up): ip 172.12.123.3 dlci 221(0xDD,0x34D0), static,
    CISCO, status defined, active
```

The mapping is static, and since R1 is the remote end on this DLCI and has been configured for frame relay, the DLCI is active. For complete connectivity, R3 must now be configured.

A point-to-point interface will be configured on R3's Serial0 interface. Remove any IP address from the physical interface, enable Frame Relay on that interface, and disable dynamic mapping with **no frame-relay inverse-arp**. Configure the point-to-point interface as shown, with the **frame-relay interface-dlci** command.

```
R3#conf t
R3(config)#interface serial0
R3(config-if)#no ip address
R3(config-if)#encapsulation frame-relay
R3(config-if)#no frame-relay inverse-arp
R3(config-if)#interface serial 0.31 point-to-point
R3(config-subif)#ip address 172.12.123.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 321
```

*Point-to-point Serial interfaces on a frame relay network do not use dynamic or static mappings. A point-to-point interface has only one possible destination – the other end of the point-to-point connection. With only one possible destination, no mapping is necessary. Instead, the command **frame-relay interface-dlci** indicates the single DLCI that will be used by this interface.*

R3#show frame map

```
Serial0.31 (up): point-to-point dlci, dlci 321(0x141,0x5010), broadcast
status defined, active
```

“show frame map” shows no static or dynamic mappings.

From each router, **ping** the other two routers' Serial interfaces on the frame relay network. All pings will be successful.

Numbering Frame Relay Logical Interfaces

A good habit to develop now is to give your Frame Relay point-to-point and point-to-multipoint subinterfaces numbers that reflect what routers are connected. In this lab, the multipoint interface on R2 was numbered 0.123, indicating what routers are connected by it. The point-to-point interface on R3 was numbered 0.31, again indicating what routers are connected on that link. Developing this habit now will allow you to look at a interface number in more complex lab environments and know immediately what routers are connected by it.

On R1, change the frame LMI type to ANSI with the **frame-relay lmi-type** command. After about 30 seconds, the line will go down.

```

R1#conf t
R1(config)#interface serial0
R1(config-if)#frame-relay lmi-type ansi
00:46:40: %SYS-5-CONFIG_I: Configured from console by console
R1#
00:47:12: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 122 state changed to
INACTIVE
00:47:12: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 123 state changed to
INACTIVE
00:47:12: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 122 state changed to
DELETED
00:47:12: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 123 state changed to
DELETED
00:47:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to down

```

The LMI mismatch leads to the line going down and the DLCIs going inactive.

Run **show frame lmi** on R1.

```

R1#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0      Invalid Prot Disc 0
  Invalid dummy Call Ref 0     Invalid Msg Type 0
  Invalid Status Message 0     Invalid Lock Shift 0
  Invalid Information ID 0     Invalid Report IE Len 0
  Invalid Report Request 0    Invalid Keep IE Len 0
Num Status Enq. Sent 256      Num Status msgs Rcvd 240
Num Update Status Rcvd 0    Num Status Timeouts 16

```

The router is receiving LMI status messages, but when the LMI type was changed, the Status Timeouts began to accrue. This command gives an indication that there is a problem with the LMIs. The LMIs are the heartbeat of frame relay; without the right LMIs, the frame connection dies.

Run **debug frame lmi** on R1.

```
R1#debug frame lmi
```

Frame Relay LMI debugging is on

Displaying all Frame Relay LMI data

```
00:52:12: Serial0(out): StEnq, myseq 31, yourseen 0, DTE down
```

```
00:52:12: datagramstart = 0xE0183C, datagramsize = 14
```

```
00:52:12: FR encapsulation = 0x00010308
```

```
00:52:12: 00 75 95 01 01 00 03 02 1F 00
```

```
00:52:12:
```

```
00:52:22: Serial0(out): StEnq, myseq 32, yourseen 0, DTE down
```

```
00:52:22: datagramstart = 0xE0183C, datagramsize = 14
```

```
00:52:22: FR encapsulation = 0x00010308
```

```
00:52:22: 00 75 95 01 01 00 03 02 20 00
```

```
00:52:22:
```

```
00:52:32: Serial0(out): StEnq, myseq 33, yourseen 0, DTE down
```

```
00:52:32: datagramstart = 0xE0183C, datagramsize = 14
```

```
00:52:32: FR encapsulation = 0x00010308
```

```
00:52:32: 00 75 95 01 01 00 03 02 21 00
```

The “myseq” value continues to increase, but the “yourseen” value remains at 0. Between “debug frame lmi” and “show frame lmi”, it can be seen that LMI messages are being received from the DCE, but not accepted – another indicator of an LMI mismatch.

Leave **debug frame lmi** on while changing the LMI type back to Cisco.

```
R1#debug frame lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R1#conf t
R1(config)#interface serial0
R1(config-if)#frame-relay lmi-type cisco

00:56:22: Serial0(out): StEnq, myseq 1, yourseen 0, DTE down
00:56:22: datagramstart = 0xE0183C, datagramsize = 13
00:56:22: FR encapsulation = 0xFCF10309
00:56:22: 00 75 01 01 00 03 02 01 00
00:56:22: Serial0(in): Status, myseq 1
00:56:22: RT IE 1, length 1, type 0
00:56:22: KA IE 3, length 2, yourseq 1 , myseq 1
00:56:22: PVC IE 0x7 , length 0x6 , dlci 122, status 0x2 , bw 0
00:56:22: PVC IE 0x7 , length 0x6 , dlci 123, status 0x2 , bw 0
00:56:32: Serial0(out): StEnq, myseq 2, yourseen 1, DTE down
00:56:32: datagramstart = 0xE0183C, datagramsize = 13
00:56:32: FR encapsulation = 0xFCF10309
00:56:32: 00 75 01 01 01 03 02 02 01
00:56:32: Serial0(in): Status, myseq 2
00:56:32: RT IE 1, length 1, type 0
00:56:32: KA IE 3, length 2, yourseq 2 , myseq 2
00:56:32: PVC IE 0x7 , length 0x6 , dlci 122, status 0x2 , bw 0
00:56:32: PVC IE 0x7 , length 0x6 , dlci 123, status 0x2 , bw 0
00:56:42: Serial0(out): StEnq, myseq 3, yourseen 2, DTE up
00:56:42: datagramstart = 0xE0183C, datagramsize = 13
00:56:42: FR encapsulation = 0xFCF10309
00:56:42: 00 75 01 01 01 03 02 03 02
00:56:42: Serial0(in): Status, myseq 3
00:56:42: RT IE 1, length 1, type 1
00:56:42: KA IE 3, length 2, yourseq 3 , myseq 3
00:56:43: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
00:57:22: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 122 state changed to ACTIVE
00:57:22: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 123 state changed to ACTIVE

The incoming "myseq" packets are now being accepted, and the outgoing messages see the "yourseen" value begin to accrue. The DTE end of the connection goes up, the line protocol goes up soon after that, and finally the previously deleted DLCIs are again active.
```

Run **show frame pvc** on R1. Note the status for each DLCI, and the uptime.

```
R1#show frame pvc
```

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 122, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 20	output pkts 30	in bytes 2080
out bytes 3002	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	
out bcast pkts 1	out bcast bytes 30	
pvc create time 00:56:53, last time pvc status changed 00:06:16		

DLCI = 123, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 42	output pkts 20	in bytes 8262
out bytes 2080	dropped pkts 2	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	
out bcast pkts 0	out bcast bytes 0	
pvc create time 00:56:56, last time pvc status changed 00:06:18		

What are FECN, BECN, and DE?

The output of "show frame pvc" displays FECN, BECN, and DE packets coming in and out of the interface.

FECN and BECN are both Frame Relay congestion notifications. FECN indicates congestion was encountered in the direction in which the packet was traveling. If a FECN (pronounced "feckin")packet is received, the congestion was encountered on the way in.

BECN indicates congestion in the OPPOSITE direction in which the packet was received. If a BECN (pronounced "beckin")packet is received, the congestion was encountered in the outgoing direction.

DE stands for Discard Eligible. Under certain circumstances, a Frame packet can be discarded in case of network congestion if the DE bit is set.

Notes

Section Four: Point-to-Point Links and ISDN

ISDN is one of the most important topics you'll see on the CCNA exams, and in real life. You've got to know why an ISDN line goes down, and what makes it come up. You've got to know the commands that will give you this information.

PAP and CHAP cause engineers a lot of trouble. The theory is fairly simple, but again, you've got to know how to debug both of them. Between getting some hands-on work with ISDN and this chapter, you'll be ready for the many ISDN challenges on the CCNA exams.

*Chris Bryant
CCIE #12933*

Commands Reviewed In This Section and Labs:

Clock rate – Supplies clocking rate between two routers whose Serial interfaces are directly connected by a DTE/DCE cable. Configured on DCE interface only.

Debug ppp negotiation – Used to observe PAP and CHAP processes while authentication does (or does not) take place.

Dialer list – Indicates what protocol is considered interesting traffic. Can stand alone or link to an access-list.

Dialer group – Interface-level command indicating which dialer-list is determining the interesting traffic for this particular interface.

Dialer pool – Used with Dialer Profiles, configured on the logical interface. Indicates membership in a given pool of logical addresses.

Dialer pool-member – Used with Dialer Profiles, configured on the physical interface. Indicates what dialer pool contains the logical interface that will be bound to the physical interface.

Dialer load-threshold – Used with PPP Multilink to determine at what load level the additional lines will be brought up.

Show controller – Used to detect if a Serial interface has the DCE or DTE end of a DCE/DTE cable attached.

Show dialer – Lists current calls across ISDN link, and the source and destination addresses of the interesting traffic that brought the line up. Indicates time the link has been up and how long before a lack of interesting traffic will bring the line down.

Show isdn status -- Indicates type of ISDN switch the router has been configured to communicate with. Lists spids and whether they are valid.

Point-to-Point Serial Links

In the network at your job, most likely the serial interfaces on your Cisco routers are not connected to each other directly. They connect to a CSU/DSU, which supplies a clock rate to the router, allowing the line protocol to stay up.

In the world of Cisco exams, and in your practice lab, there are generally routers that have directly connected serial interfaces. These routers are connected to each other by a DCE/DTE cable; the DCE end of the cable will connect to the router that is acting as the CSU.



What's The Line Protocol?

You'll see a lot of discussion in CCNA and CCNP texts, but rarely does anyone actually say what the line protocol *is*. The Cisco IOS Command Reference defines the line protocol as "indicating whether the software processes that handle the line protocol consider the line usable (that is, keepalives are successful) or whether it has been taken down by an administrator."

Translation: When the line protocol is down, there's a problem with the keepalives or the encapsulation type.

To tell the DTE end from the DCE end before connecting it, look for a small label wrapped around one or both of the cable ends. That label will indicate whether that is the DCE or DTE end. If there is no label, the connector itself may have DTE or DCE imprinted on it.

After connecting the cable to the respective routers, use **show controller** to ensure the router sees the cable as a DCE or DTE.

R1#show controller serial 1

HD unit 1, idb = 0x107114, driver structure at 0x10C590
buffer size 1524 HD unit 1, **V.35 DTE cable**

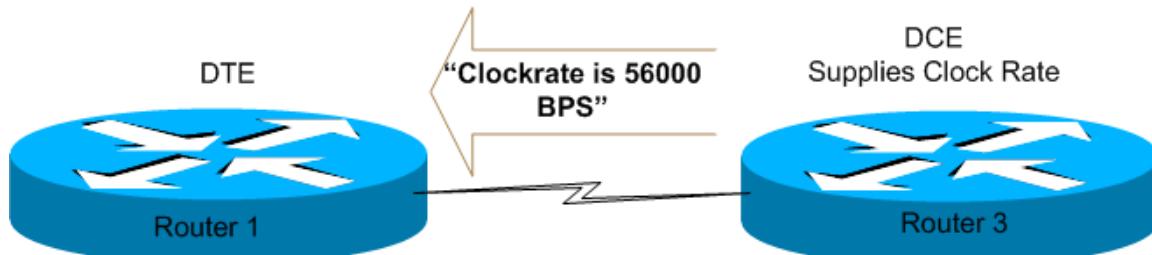
R3#show controller serial 1

HD unit 1, idb = 0xC7D1C, driver structure at 0xCCAA0
buffer size 1524 HD unit 1, **V.35 DCE cable**

The routers will not be able to communicate at this point, however. Remember that when a serial interface connects to a CSU/DSU, the interface receives clocking from that device. There is no CSU/DSU involved when two serial interfaces are directly connected; therefore, one of the devices must supply a clock rate to the other. The DCE interface must supply the clock rate to the DTE.

```
R3(config)#int serial 1
R3(config-if)#clockrate 56000
< The DCE interface now has a clockrate and will provide clocking to the remote DTE.>
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to up
R3#ping 172.12.13.1
```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.13.1, timeout is 2 seconds:
!!!!



In The REAL World...

For exam purposes, you need to memorize the fact that the DCE is the interface that needs to have the clock rate configured. When you're at your practice rack, you'll find out that you can't put the clockrate on the DTE, because the router won't let you!

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int serial1
R1(config-if)#clockrate 56000
%Error: This command applies only to DCE interfaces
R1(config-if)#+
```

HDLC vs. PPP: The Similarities and Differences

HDLC and PPP are the two data-link (Layer 2) protocols to consider when choosing an encapsulation method across a serial or ISDN link.

There are two major points of distinction between the two. First, HDLC is the default encapsulation for any serial point-to-point link. Second, HDLC supports synchronous links, but not asynchronous links. PPP supports both synchronous and asynchronous links.

Synchronous lines require the CSU/DSU at opposite ends of the same serial link to run at the same speed. To accomplish this, these devices will send **idle frames** to each other in order to keep the clockrate synchronized.

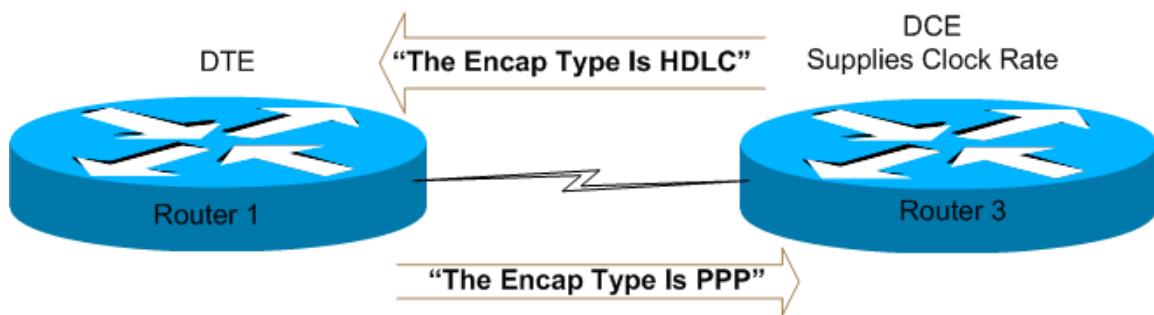
Asynchronous lines do not require the CSU/DSUs to run at the same speed, and idle frames are not transmitted.

As mentioned, HDLC is the default. Changing the encapsulation type to PPP is simple enough, but be prepared for the line to go down when doing so:

```
R1#conf t
R1(config)#interface serial1
R1(config-if)#encapsulation ppp
1w2d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
changed state to down
1w2d: %LINK-3-UPDOWN: Interface Serial1, changed state to up
< The line protocol never comes back up. >
```

Note that the line protocol went down. The physical interface went down, but the line protocol stayed down. The issue is that the opposite end of the connection, interface Serial1 on R3, is still running HDLC. Mismatched encapsulation types will result in the line protocol going down and staying down.

When the DTE and DCE disagree on the encapsulation type, the line protocol comes down.



Changing the encap type to PPP on R3 will result in the line protocol coming back up.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int s1
R3(config-if)#encapsulation ppp
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to up
```

A **show interface serial1** on r1 shows the encapsulation type is PPP and the line protocol is now up.

```
R1#show interface serial 1
Serial1 is up, line protocol is up
Hardware is HD64570
Internet address is 172.12.13.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive set (10 sec)
```

Troubleshooting Serial Lines With “Show Interface” and “Show Controller”

The use of “show interface” and “show controller” is a vital part of your troubleshooting skills. These commands will result in a lot of output, but when working with directly connected serial lines, the two factors to consider are encapsulation type and whether the DCE interface is configured to send clocking to the DTE. These are the two most common errors by far when troubleshooting directly connected serial lines, and these are the two commands to use in that situation.

Authenticating Over WAN Links With PPP

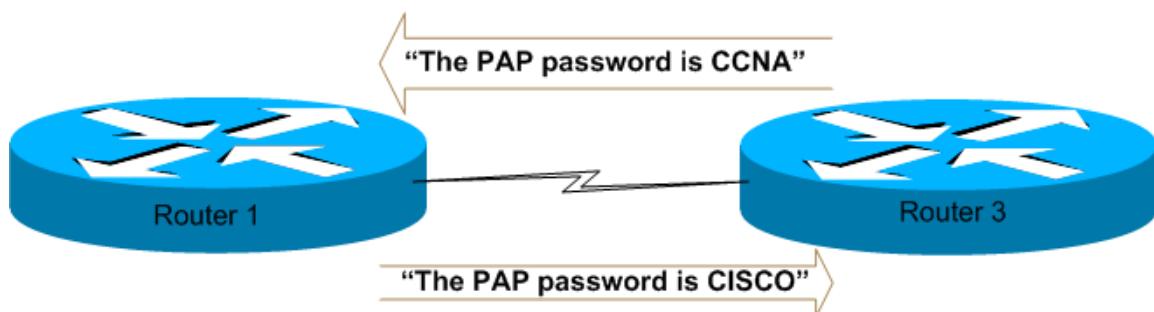
In today's security-conscious world, the need for authentication is apparent. HDLC doesn't give authentication options, but PPP gives us two: PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol).

Authenticating With PAP

PAP has a major shortcoming in that it sends both the username and password over the WAN link in clear-text. Anyone who is able to capture data traversing the WAN link would easily be able to determine the username and password being used.

PAP does have one advantage, however slight, over CHAP. PAP allows the routers at the end of the link to use different passwords, whereas with CHAP, the passwords must be the same for reasons examined later in this section.

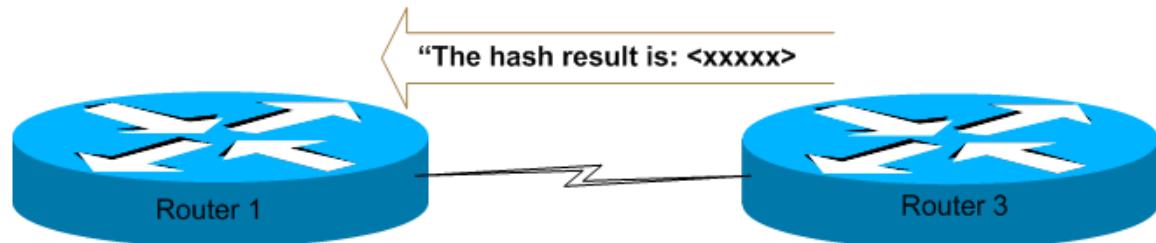
*The only real advantage to using PAP authentication
is that the passwords can be different.*



Authenticating with CHAP

CHAP does not pass any of this information over the link. Instead, CHAP runs a hash algorithm using the password and a random number. It is the result of this hash that is passed over the link. The remote router receives the hash result and then runs the exact same hash. The result must match the result it received from the other router. For this reason, the passwords must be the same. (The random number is determined by the router and is not manually configured.)

With CHAP authentication, the password is NEVER passed across the link. CHAP runs a hash algorithm using the password and a random number, and then sends the result of that hash to the remote router.



"I ran the same hash as Router 3 using the same password, and got the same result. I will notify R3 that the authentication is successful."

There are several examples of both PAP and CHAP in the ISDN section of the chapter, and in the accompanying lab exercises. The authentication process of both protocols is outlined there as well.

ISDN

ISDN (Integrated Services Digital Network) is often used by companies where a 24/7 connection is not necessary. Remote offices often use it to send data to a central location when the data is not needed immediately by that central location.

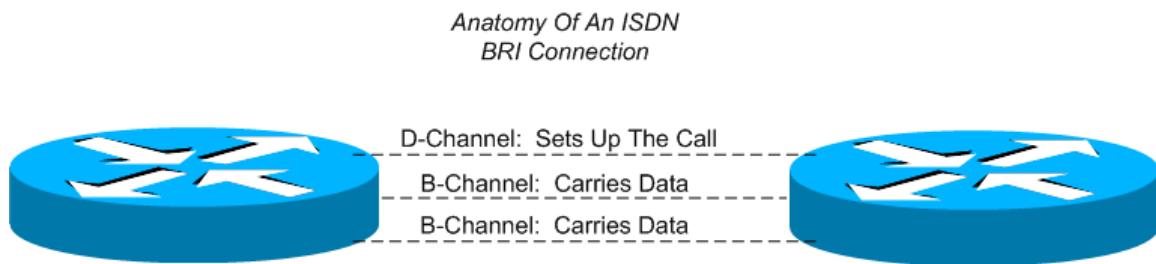
ISDN is also used for a backup line if the primary link to a location is down. ISDN can be configured to come up only if that primary link becomes unavailable for a certain amount of time, or if a certain route becomes unavailable.

With ISDN, it's vital to know under what circumstances the link will come up, and what will make the link go down. Why is that so important? Because the ISDN connection is a phone call, and businesses tend to frown on unending phone calls, particularly when they get the bill.

The ISDN Channels

ISDN links contain two kinds of interfaces, Basic Rate Interface (BRI) and Primary Rate Interface (PRI). The Cisco routers in the ICND course have BRI interfaces.

ISDN links also contain two kinds of channels: B-channels and D-channels. The number of B-channels depends on the type of interface. There will always be only one D-channel, but the capacity of that channel also depends on the type of interface.



There is also a difference between the number of B-channels found in a PRI in the United States and a PRI overseas, as shown in this chart:

	# B-Channels	B-Channel Capacity	# D-Channels	D-Channel Capacity
PRI (US)	23	64 kbps	1	64 kbps
BRI	2	64 kbps	1	16 kbps
PRI (Europe)	30	64 kbps	1	64 kbps

Dial-On-Demand Routing

An important concept when working with ISDN is that the line should only come up when there is data to send, and should go down in a reasonable time after the data is finished transmitting. This brings up another series of questions, though. What traffic should bring the link

up? How long should it remain up once that data is sent? These questions are answered when configuring dial-on-demand routing.

Before defining what traffic should bring the link up and when it should come down, there are some foundation tasks that must be performed and checked.

The first is to configure the ISDN switchtype that will be used. Without this command, no communication can possibly take place. The command **isdn switch-type** is used for this purpose, and can be configured globally or at the interface level. A value known as a Service Profile Identifier (SPID) will likely need to be configured at the interface level as well. Once this data is entered, always run **show isdn status** to verify that the switchtype has been configured and the spids are operational.

```
R1(config)#isdn switch-type basic-ni
< Global command defining ISDN switch-type. Can also be configured at the interface
level. When configured globally, this command may appear in the running
configuration at both the global and interface level. >
R1(config)#int bri0
R1(config-if)#isdn spid1 0835866101
R1(config-if)#isdn spid2 0835866301

R1#show isdn status
Global ISDN Switchtype = basic-ni < ISDN switch type is correctly configured.>
ISDN BRI0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 91, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
        TEI = 92, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Spid Status:
        TEI 91, ces = 1, state = 5(init)
        spid1 configured, no LDN, spid1 sent, spid1 valid < SPID 1 is good.>
        Endpoint ID Info: epsf = 0, usid = 1, tid = 1
        TEI 92, ces = 2, state = 5(init)
        spid2 configured, no LDN, spid2 sent, spid2 valid < SPID 2 is good.>
        Endpoint ID Info: epsf = 0, usid = 3, tid = 1
    Layer 3 Status:
        0 Active Layer 3 Call(s)
        Activated dsl 0 CCBs = 0
        The Free Channel Mask: 0x80000003
```

The SPIDs are valid and the configuration can continue.

Interesting Traffic and ISDN

Interesting Traffic is the Cisco term for traffic that will cause the router to dial another router. Interesting traffic is defined by the global **dialer-list** command, which may or may not refer to an access-list. Regardless of whether an access-list is involved, the **dialer-group** interface-level command links the physical interface to the dialer-list command defining interesting traffic.

Dialer-List With No Access List, Defining All IP Traffic As Interesting:

```
R1#conf t
R1(config)#dialer-list 1 protocol ip permit < defines all IP traffic as interesting>
R1(config)#int bri0
R1(config-if)#dialer-group 1 < links interface BRI to dialer-list 1 >
```

Dialer-List Calling An Access-List: Only ICMP Traffic is Interesting:

```
R1(config)#access-list 105 permit icmp any any
< The access-list permits all ICMP traffic, regardless of source of destination. >
R1(config)#dialer-list 1 protocol ip list 105
< The dialer-list option "list 105" means access-list 105 defines interesting traffic. >
R1(config)#interface bri0
R1(config-if)#dialer-group 1
```

If you're not familiar with access-lists at this point, suffice to say that the access list 105 shown permits ICMP traffic (pings, primarily) from any source and any destination. Cisco access-lists have an "implicit deny" at the end, meaning that traffic that is not expressly permitted will be denied. Access-lists are covered in another chapter.

What Traffic Travels Across The ISDN Link Once It Is Up ?

While interesting traffic is the only traffic that will actually cause the router to dial, once that link is up, by default any traffic can traverse the link.

How long does the link stay up? What makes the link stay up ?

Cisco uses an idle-timer to judge whether the line should stay up or be torn down. By default, this idle-timer is set for 120 seconds when interesting traffic crosses the link, and counts down until interesting traffic resets the timer or it reaches zero, at which point the link will be torn down.

It's important to note that while any traffic can cross the link once it's up, only interesting traffic resets the idle-timer. A flow of non-interesting traffic can be crossing the link, but if that timer expires while the flow is crossing, the connection will still be torn down. **Only interesting traffic resets the idle-timer.**

Manually Setting The Idle-Timer

Use the **dialer idle-timeout** command to alter this default. The command can also be used to reset the idle timer when interesting traffic enters or leaves the interface.

To use Cisco IOS Help to view a command's options, type a question mark and then hit <ENTER>. The IOS will list the available options. Help can also be used to see the available commands that begin with any given letter or letters:

```
R1#configure ?
memory      Configure from NV memory
network      Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
terminal      Configure from the terminal
```

Cisco commands use several different units of time and data. **Always use Cisco IOS Help to verify the correct unit when configuring a value of time or data.**

In The REAL World...

You should not expect IOS Help to be available on any questions on the CCNA exam. However, you should begin now to become comfortable with using IOS Help. It is an invaluable resource every Cisco engineer should be adept with.

Dialing The Remote Router

There are two methods of supplying the router with the number to dial when interesting traffic triggers a call. When configuring Legacy DDR on the physical interface, a **dialer map** is used.

Configuring A Dialer-Map On A Physical Interface.

```
R1(config)#int bri0  
R1(config-if)#dialer map ip 172.12.21.2 name R2 broadcast 8358662
```

Examining The Dialer Map Command:

“**dialer map**” – beginning of the command.

“**ip**” – Protocol mapped by this particular map.

“**172.12.21.2**” – Remote router IP address.

“**name**” – required command.

“**R2**” – remote router name

“**broadcast**” – required to allow broadcasts to traverse the ISDN line. Default behavior is for broadcasts to not go over the link. This default behavior will adversely affect routing protocol behavior.

“**8358662**” – number to dial to contact remote router

Dialing information can also be configured on a logical interface.

Dialer Profiles allow different dialing information to be applied to the logical interfaces. The logical interfaces may have different dialing destinations, different remote router names, etc., but they'll be using the same physical interface. *Dialer strings* are used on dialer profiles. Note that each logical interface has a different IP address, a different remote router to dial, and a different dialer string, but they will be using the same physical interface to dial out. The commands **dialer pool** and **dialer pool-member** are used to link the logical and physical interfaces. The number following each command must match for the logical interface to correctly bind to the physical interface.

Configuring a logical dialer profile and the physical BRI interface.

```
R1(config)#interface dialer0
R1(config-if)#ip address 172.16.1.1 255.255.255.0
R1(config-if)#encapsulation ppp
<. The encapsulation type is placed on both the logical and physical interfaces. >
R1(config-if)#dialer remote-name Remote0
<name of remote router>
R1(config-if)#dialer pool 1
< places logical interface into dialer pool >
R1(config-if)#dialer string 5551212
< number dialed to contact router Remote0 >
R1(config-if)#dialer-group 1
< links logical interface to dialer-list 1 >

R1(config)#interface dialer1
R1(config-if)#ip address 172.16.1.2 255.255.255.0
R1(config-if)#encapsulation ppp
R1(config-if)#dialer remote-name Remote1
R1(config-if)#dialer pool 1
R1(config-if)#dialer string 5551234
R1(config-if)#dialer-group 1

R1(config)#interface bri0
R1(config-if)#no ip address
< With dialer profiles, IP addresses are assigned to logical interfaces. >
R1(config-if)#encapsulation ppp
< The encapsulation type is place on both the logical and physical interfaces.>
R1(config-if)#dialer pool-member 1
< The number associated with this command should match the number configured
with the dialer pool number on the logical dialer interfaces. >
R1(config-if)#isdn spid1 0835866101
R1(config-if)#isdn spid2 0835866301
<The SPIDs are always placed on the physical interface only.>
```

When configuring dialer profiles, the encapsulation type should be placed on both the physical BRI interface and the logical dialer interfaces. The SPIDs are configured on the physical interface as well.

Troubleshooting ISDN

ISDN can be tricky to configure at times, but there are several helpful commands to help identify any issues that may arise.

The biggest issue with ISDN, whether on the job or in the lab, is when the line comes up when you don't want it to, or when you're not sure what is causing the line to come up in the first place. As you progress through your CCNA, CCNP, and CCIE studies, this situation will occur more often. Become familiar with **show dialer**, and continue to use it, as knowing what traffic is bringing the line up is more than half the battle.

Examining the output of “show dialer”.

R1#show dialer

BRI0 - dialer type = ISDN

Dial String	Successes	Failures	Last called	Last status
8358662	1	0	00:00:59	successful

0 incoming call(s) have been screened.

0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is data link layer up

Dial reason: ip (s=172.12.21.1, d=172.12.21.2) < *s=Source, d=Destination* >

Time until disconnect 62 secs

< *Idle timer, reset by interesting traffic only* >

Connected to 8358662 (R2)

BRI0:2 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is idle

The output of this command shows that one call went through successfully, and two failed; there is currently a call on BRI0:1; the source of the interesting traffic is 172.12.21.1, destined for 172.12.21.2; the call has 117 seconds left unless interesting traffic resets the timer; and finally, the number dialed is 8358662 on R2.

Show ISDN Status

Show isdn status should be run immediately after configuring the ISDN switchtype and the spids, which are usually the first two things configured when running ISDN. This command indicates that you have correctly set the switchtype, and under the Spid Sent section, indicates the validity of the spids.

The output of “show isdn status” indicates the configuration is correct. The ISDN switch-type has been configured, the SPIDs are valid, and there is an active Layer 3 call on the line.

```
R1#show isdn status
Global ISDN Switchtype = basic-ni <ISDN switch-type set correctly>
ISDN BRI0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 93, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    TEI = 94, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Spid Status:
    TEI 93, ces = 1, state = 5(init)
        spid1 configured, no LDN, spid1 sent, spid1 valid
        Endpoint ID Info: epsf = 0, usid = 1, tid = 1
    TEI 94, ces = 2, state = 5(init)
        spid2 configured, no LDN, spid2 sent, spid2 valid
        Endpoint ID Info: epsf = 0, usid = 3, tid = 1
Layer 3 Status:
    1 Active Layer 3 Call(s) <one call is currently active on the line>
Activated dsl 0 CCBs = 1
    CCB:callid=0x800D, sapi=0x0, ces=0x1, B-chan=1
The Free Channel Mask: 0x80000002
Total Allocated ISDN CCBs = 1
```

If the SPIDs have been entered incorrectly, **show isdn status** will indicate it.

"show isdn status" indicates the SPIDS are incorrectly configured.

```
R1#show isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 97, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Spid Status:
        TEI 97, ces = 1, state = 6(not initialized)
        spid1 configured, no LDN, spid1 sent, spid1 NOT valid
        TEI Not Assigned, ces = 2, state = 1(terminal down)
        spid2 configured, no LDN, spid2 NOT sent, spid2 NOT valid
Layer 3 Status:
    0 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 0
    The Free Channel Mask: 0x80000003
    Total Allocated ISDN CCBs = 0
```

A readout that indicates a spid has not been sent, or that the spid is invalid, generally indicates that a spid was entered incorrectly. In this case, go back to the running configuration and double-check the spid entries under the BRI interface. Change the spid to the correct value, open and shut the interface, and run **show isdn status** again.

```
R1#show isdn status
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 98, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    TEI = 99, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Spid Status:
        TEI 98, ces = 1, state = 5(init)
        spid1 configured, no LDN, spid1 sent, spid1 valid
        Endpoint ID Info: epsf = 0, usid = 1, tid = 1
        TEI 99, ces = 2, state = 5(init)
        spid2 configured, no LDN, spid2 sent, spid2 valid
        Endpoint ID Info: epsf = 0, usid = 3, tid = 1
```

The spids now show as valid.

Show ISDN History

```
R1#show isdn history
```

ISDN CALL HISTORY

History table has a maximum of 100 entries.

History table data is retained for a maximum of 15 Minutes.

Call Type	Calling Number	Called Number	Remote Name	Seconds Used	Seconds Left	Seconds Idle	Charges Units/Currency
Out		8358662	R2	121			0
Out		8358662	R2	121			0

To see what calls have been made in the last 15 minutes, run **show isdn history**. This command can help you diagnose the problem when an ISDN line is continually going up and down ("flapping").

ISDN and Point-to-Point Q&A

1. You are preparing to configure a point-to-point ISDN link. You are concerned about the number of addresses available to you, and you want to conserve addresses. What subnet mask should you use on the ISDN link in order to conserve addresses as much as possible while having a fully functional link?

- A. **255.255.255.252**
- B. **255.255.255.254**
- C. **255.255.255.0**
- D. **/31**
- E. **/30**
- F. **/32**

Answers: A, E. For a point-to-point link, two valid host addresses are needed. Both A and E deliver two valid host addresses. Pronounced "slash 30", /30 is simply another way to indicate a subnet mask of 255.255.255.252. This is the tightest subnet mask you can use and still have two usable host addresses.

C would result in 254 addresses, far too many for a point-to-point link. B and D are not valid masks, since having one host bit would result in zero usable host addresses.

2. In ISDN, which of the following is true of a "B" channel? Choose three.

- A. **The "B" channel bears data.**
- B. **The "B" channel sends an indication that a new call is being created.**
- C. **The "B" channel's capacity is 64 kbps.**
- D. **The "B" channel's capacity is 16 kbps.**
- E. **There is one "B" channel in a BRI interface.**
- F. **There are two "B" channels in a BRI interface.**

Answers: A, C, F. The purpose of the B-channel is to bear (transport) the data. A B-channel's capacity is 64 kbps, and there are two of them in a BRI interface. Answers B, D, and E are descriptive of a D-channel.

3. Call setup and teardown messages are defined by what protocol?

- A. IP
- B. TCP
- C. Q.931
- D. Q.921
- E. ISDN

Answer: C. Q.931 defines call setup and teardown messages.

4. By default, what traffic causes a BRI interface to dial?

- A. Any IP traffic.
- B. Any traffic specifically defined for the IP address on the other side of the link.
- C. None.
- D. Traffic defined in the dialer map statement.

ANSWER: C. Traffic that causes the BRI interface to dial is referred to as *interesting traffic*, and by default no traffic is interesting traffic.

5. Which of the following can be used to define interesting traffic?
Choose three.

- A. An access-list
- B. A dialer-map statement.
- C. A dialer-list statement on the BRI interface.
- D. A dialer-list statement, globally configured.
- E. A dialer-group statement on the BRI interface.
- F. A dialer-group statement, globally configured.

Answers: A, D, and E. An access-list is not required, but can be used to limit the traffic allowed by the global command **dialer-list**. The dialer-group statement links the interface to the dialer-list, and is configured at the interface level. The example below shows an access-list configured to allow traffic with source address 10.0.0.0 /8; the dialer-list then allows IP traffic matching that list; finally, the interface command dialer-group 1 links to the dialer-list command, as shown:

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 1 permit 10.0.0.0 0.255.255.255
R1(config)#dialer-list 1 proto ip list 1
R1(config)#int bri0
R1(config-if)#dialer-group 1
```

6. Which router command defines the type of ISDN switch?

- A. **isdn switch-type**
- B. **ppp switch-type**
- C. **dialer-map**
- D. **dialer switch-type**

ANSWER: A. isdn switch-type is used to indicate the type of switch being used. The command can be configured globally or at the interface level.

7. Your router is displaying the following information:

```
Global ISDN Switchtype = basic-ni
ISDN BRI0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 96, Ces = 1, SAPI = 0, State =
MULTIPLE_FRAME_ESTABLISHED
    TEI = 97, Ces = 2, SAPI = 0, State =
MULTIPLE_FRAME_ESTABLISHED
Spid Status:
    TEI 96, ces = 1, state = 5(init)
        spid1 configured, no LDN, spid1 sent, spid1 valid
        Endpoint ID Info: epsf = 0, usid = 1, tid = 1
    TEI 97, ces = 2, state = 5(init)
        spid2 configured, no LDN, spid2 sent, spid2 valid
        Endpoint ID Info: epsf = 0, usid = 3, tid = 1
```

What command caused the router to give this display?

- A. **show isdn switchtype**
- B. **show interface bri0 interface detail**
- C. **show isdn status**
- D. **show dialer**
- E. **show isdn spids**
- F. **debug isdn switch**

Answer: C. show isdn status displays the type of ISDN switch that has been configured on the router, and also gives a warning if no switchtype has been configured. It also shows Layer 2 status and the validity of the spids.

8. You receive a report that the ISDN connection is not working properly. You run show isdn status and see this display:

```
R1#show isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
    DEACTIVATED
Layer 2 Status:
    TEI = 96, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
    TEI = 97, Ces = 2, SAPI = 0, State = TEI_ASSIGNED
Spid Status:
    TEI 96, ces = 1, state = 5(init)
        spid1 configured, no LDN, spid1 sent, spid1 valid
        Endpoint ID Info: epsf = 0, usid = 1, tid = 1
    TEI 97, ces = 2, state = 5(init)
        spid2 configured, no LDN, spid2 sent, spid2 valid
        Endpoint ID Info: epsf = 0, usid = 3, tid = 1
Layer 3 Status:
    0 Active Layer 3 Call(s)
```

What is the most likely source of the problem?

- A. **Layer 3 is not properly configured.**
- B. **The switchtype is invalid.**
- C. **The assigned TEIs at Layer 2 are misconfigured.**
- D. **The interface is down.**

ANSWER: D. If Layer 1, the Physical layer, is deactivated, either the interface has been shut down or another physical problem exists. The switchtype is valid, and you will not manually assign TEI numbers, so those answers would not be correct. The Layer 3 status shown is expected; since Layer 1 is deactivated, there can't be any Layer 3 calls.

9. What is the default encapsulation on an ISDN link?

- A. LCP
- B. PPP
- C. ISDN
- D. HDLC

ANSWER: D. HDLC is the default encapsulation on an ISDN or serial link.

10. Which of the following statements best describes the concept of interesting traffic?

- A. Any traffic crossing an ISDN line is by default considered interesting traffic.
- B. Any traffic that is encapsulated using PPP is interesting traffic, as long as CHAP is used.
- C. Any traffic that causes a router to dial a remote ISDN router is interesting traffic.
- D. Any traffic that causes a router to immediately disconnect from the remote router due to framing errors is considered interesting traffic.

ANSWER: C. Interesting traffic is traffic that matches criteria set forth by the dialer-list command (and perhaps an accompanying access-list), which will cause the router to dial the remote router.

11. By default, once an ISDN link has been established between two routers, what traffic can then cross the link?

- A. All traffic.
- B. Interesting traffic only.
- C. Traffic expressly permitted by an access-list
- D. Traffic implicitly permitted by an access-list.
- E. Traffic meeting the criteria of the "dialer map" command.

ANSWER: A. By default, all traffic can be transported over an ISDN link once interesting traffic has brought the line up. You can use an access-list to deny or permit traffic, but that is not the default.

12. There is a timer that defines how long an ISDN link will stay up once the connection is established. What is the default value for this timer, and what traffic will reset the timer?

- A. **The default is one minute, and any traffic resets the timer.**
- B. **The default is two minutes, and any traffic resets the timer.**
- C. **The default is one minute, and the timer cannot be reset.**
- D. **The default is two minutes, and the timer cannot be reset.**
- E. **The default is one minute, and only interesting traffic resets the timer.**
- F. **The default is two minutes, and only interesting traffic resets the timer.**

ANSWER: F. While any traffic can traverse the link once interesting traffic brings it up, the link stays up for only two minutes unless interesting traffic again crosses the link, which will reset the idle-timer.

13. Your company uses ISDN at its remote sites to transfer small amounts of vital data to your company headquarters. You notice that the majority of these calls last less than a minute, but our phone bills are for two-minute calls. The ISDN link is not used for any other purpose at the remote sites.

You need to cut these phone bills. Which of the following commands is the most appropriate to use in this situation?

- A. **dialer fast-idle**
- B. **dialer idle-timeout**
- C. **dialer string**
- D. **isdn timeout**
- E. **dialer load-threshold**

ANSWER: B. Your remote sites are bringing up the ISDN link and transmitting a small amount of data. The link will then wait the default two minutes, and if no other interesting traffic crosses the link, the link will be torn down.

Since this is the only interesting traffic coming from the remote site, the line will wait the default two minutes before coming down. Your company is being billed for those two minutes time and again. You can use dialer idle-timeout to adjust the time the line will wait in the absence of interesting traffic before the line is brought down.

14. Which of the following defines interesting traffic as being traffic sourcing from network 192.168.15.0 /24?
- A. **access-list 5 permit 192.168.15.0 0.0.0.255**
dialer-list 1 proto ip list 5
interface bri0
dialer-group 1
 - B. **access-list 5 deny any**
access-list 5 permit 192.168.15.0 0.0.0.255
dialer-list 1 proto ip list 5
interface bri0
dialer-group 1
 - C. **access-list 5 permit 192.168.15.0 0.0.0.255**
dialer-group 1 protocol ip list 5
interface bri0
dialer-list 1
 - D. **access-list 5 permit isdn 192.168.15.0 0.0.0.255**
dialer-list 1 protocol ip list 5
interface bri0
dialer-list 1
 - E. **access-list 5 permit 192.168.15.0 0.0.255.255**
dialer-list 5 protocol ip list 5
interface bri0
dialer-group 5

ANSWER: A. Answer "b" cannot be correct; it leads its access-list off with a "deny any" statement, which would match all traffic. The second line of that list would never be reached. Answer "c" cannot be correct; it has the "dialer-group" and "dialer-list" commands in the wrong places. (dialer-group is an interface-level command, where dialer-list is a global command.) Answer "d" cannot be correct; the access-list shown cannot be written. Answer "e" cannot be correct; the wildcard mask is incorrect.

15. You wish to use an authentication scheme across your ISDN link. Your boss insists that the passwords and usernames be sent in unencrypted form. You have the following connection on your local router:

```
hostname R1  
username R2 password ccna
```

```
interface BRI0  
ip address 172.12.21.1 255.255.255.252  
dialer map ip 172.12.21.2 name R2 broadcast 8358662  
no ip directed-broadcast  
encapsulation ppp  
dialer-group 1  
isdn switch-type basic-ni  
isdn spid1 0835866101  
isdn spid2 0835866301  
ppp authentication pap
```

Assuming the remote router is correctly configured, and the same password is in use on both routers, what additional configuration is needed on this router?

- A. A password needs to be added to the dialer map statement.**
- B. A password needs to be added to the hostname statement.**
- C. The command “ppp pap sent-username R1 password ccna” needs to be added to the global configuration.**
- D. The command “ppp pap sent-username R2 password ccna” needs to be added to the interface configuration.**
- E. The command “ppp pap password ccna” needs to be added to the global configuration.**
- F. The command “ppp pap password ccna” needs to be added to the interface configuration.**

ANSWER: C. When using PAP (clear-text) authentication, the interface-level command `ppp pap sent-username` must be used: We know this is R1 due to the “hostname” command.

```
R1(config)#int bri0  
R1(config-if)#ppp pap sent-username R1 password ccna
```

16. Which of the following is true of CHAP? Choose two.
- A. CHAP provides both clear-text and encrypted authentication.
 - B. CHAP provides clear-text authentication only.
 - C. CHAP provides encrypted authentication only.
 - D. CHAP requires passwords be the same on both routers.
 - E. CHAP does not require passwords to be the same on both routers.

ANSWER: C, D. CHAP provides an encrypted authentication scheme where the passwords are never passed over the link in clear-text. CHAP never passes the passwords over the link at all; rather, a hash algorithm is run, and the result of that operation is passed over the link. This operation requires that the password be the same on both routers in order for CHAP authentication to operate correctly.

17. Which of the following commands can contain a phone number for the router to dial? Choose two.

- A. dialer string
- B. isdn dial
- C. dialer call
- D. dialer map
- E. call line
- F. isdn dial string

ANSWER: A, D. Dialer string is primarily used in dialer profiles, where dialer map is typically used in Legacy DDR configurations. Both contain the phone number to dial upon the receipt of interesting traffic.

18. What command defines how long an ISDN link will take to timeout when all lines are in use and new interesting traffic arrives?

- A. dialer idle-timeout
- B. dialer map
- C. dialer fast-idle
- D. isdn caller

ANSWER: C. **Dialer fast-idle** is used only when all lines are in use, Chris Bryant, CCIE #12933
www.thebryantadvantage.com
© 2004 The Bryant Advantage

and new interesting traffic arrives. When one of the busy lines goes idle, this command defines how long that line will remain idle before it becomes available to the new interesting traffic. Do not confuse this command with **dialer idle-timeout**, which does not consider if lines are all busy or not.

19. When using dialer profiles, which of the following commands is configured on the physical interface?

- A. **dialer pool**
- B. **dialer pool-member**
- C. **ip address**
- D. **No commands are configured on the physical interface with dialer profiles; the dialer profile is a logical interface.**

ANSWER: B. The **dialer pool-member** command is always configured under the physical interface; it links to the **dialer pool** command that will be configured on the dialer profile itself. The IP address will be configured on the dialer profile, not the physical interface.

20. You would like the second B-channel to come up when the load on the first B-channel reached 50% of capacity. Which of the following configurations would accomplish this?

- A. **interface bri0
encapsulation ppp
ppp multilink
ppp multilink load 50**
- B. **interface bri0
encapsulation ppp
ppp multilink
ppp load-threshold percent 50**
- C. **interface bri0
encapsulation ppp
ppp multilink
dialer load-threshold 50**
- D. **interface bri0
encapsulation ppp
ppp multilink**

dialer load-threshold 127

**E. interface bri 0
 encapsulation ppp
 ppp multilink 127**

ANSWER: D. The commands **encapsulation ppp** and **ppp multilink** are required. The next command is **dialer load-threshold**. However, the value entered with this command **represents a percentage of 255, NOT 100**.

To arrive at the correct value, simply multiply the percentage at which you want the second line to come up by 255. In this question, the desired percentage is 50%. In decimal format, that is .50, or .5. Multiply that by 255 and you get 127.5. You can then round that number up or down, as you must enter a round number with this command.

21. You have two Cisco routers with their Serial0 interfaces directly connected via a DTE/DCE cable. You have determined that R1 has the DTE end of the cable attached, and R2 has the DCE cable attached. You apply IP addresses to the interfaces, and then open the interfaces. You notice that you cannot ping R1's Serial0 interface from R2, and vice versa. What additional command is needed, and where should it be applied?

- A. Apply the command bandwidth 64 to R1's Serial0 interface.**
- B. Apply the command bandwidth 64 to R2's Serial0 interface.**
- C. Apply the command clockrate 56000 to R2's Serial0 interface.**
- D. Apply the command clockrate 56000 to R1's Serial0 interface.**

ANSWER: C. The interface with the DCE end of the cable directly attached must supply the clockrate to the other interface, or the line protocol will not come up.

22. Which statement best describes how CHAP handles passwords?
- A. The password is sent in clear-text and can be different on each router.
 - B. The password is sent in clear-text and must be the same on each router.
 - C. The password is sent in encrypted form and can be different on each router.
 - D. The password is sent in encrypted form and must be the same on each router.
 - E. The password is sent in hashed form and can be different on each router.
 - F. The password is sent in hashed form and must be the same on each router.

ANSWER: F. CHAP does not simply encrypt the password. The router will run a hash algorithm using the configured password and a random number, and transmit the results of that hash to the remote router. The remote router will run the same hash and will compare the results. If the results are the same, authentication will occur. For this hash to be the same, both routers must use the same password.

23. What statement best describes how Legacy DDR handles broadcasts?
- A. Like all other traffic, broadcast traffic will cross the ISDN link once interesting traffic brings it up.
 - B. To protect the integrity of the link, broadcasts cannot cross the ISDN link under any circumstances.
 - C. Broadcasts can cross the link if the keyword "broadcast" is applied to the dialer map.
 - D. Broadcasts can cross the link by using the "isdn broadcast" command.

ANSWER: C. Broadcasts can go across the link, but only by adding "broadcast" to the dialer map statement.

ISDN / Point-To-Point Lab

R1 and R3 have a directly connected serial interface connected by a DTE/DCE cable. Using **show controller**, determine which router is the DCE.

“show controller” displays the DTE and DCE ends of the connection. The output of these commands has been truncated for clarity.

R1#show controller serial 1

HD unit 1, idb = 0x107114, driver structure at 0x10C590
buffer size 1524 HD unit 1, **V.35 DTE cable**

R3#show controller serial 1

HD unit 1, idb = 0xC7D1C, driver structure at 0xCCAA0
buffer size 1524 HD unit 1, **V.35 DCE cable**

Ping R1's serial interface from R3.

R3#ping 172.12.13.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.12.13.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

The escape sequence for pings is CTRL-SHIFT-6 performed twice in succession.

The ping fails. Run **show interface serial1** to see why.

R3#show interface serial1

Serial1 is up, line protocol is down

Hardware is HD64570

Internet address is 172.12.13.3/24

The truncated output of “show interface serial1” shows the physical interface is up, but the line protocol is down.

The line protocol is down because the DCE end of the cable must supply a clock rate to the DTE end. To resolve this, configure **clock rate 56000** on R3's Serial interface. Once the line protocol is up, run **show interface serial1** again to verify, and ping R1's Serial interface again. The ping will succeed.

```
R3#conf t
R3(config)#interface serial1
R3(config-if)#clock rate 56000

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to up

R3#show interface serial1
Serial1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.12.13.3/24
```

Once the DCE supplies a clock rate to the DTE, the line comes up.

```
R3#ping 172.12.13.1
```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/36 ms

The ping is successful.

The two BRI interfaces will now be configured with PPP PAP authentication. Configure the appropriate IP addresses and SPIDs on R1 and R2, as well as the appropriate ISDN switch-type. Run **show isdn status** after doing so to verify the switch-type is correctly configured and that the SPIDs are operational.

Configuring basic ISDN information – switch type, IP address, and spids.

```
R1#conf t
R1(config)#isdn switch-type basic-ni
R1(config)#interface bri0
R1(config-if)#ip address 172.12.21.1 255.255.255.252
R1(config-if)#isdn spid1 0835866101
R1(config-if)#isdn spid2 0835866301
R1(config-if)#no shut
```

```
R2#conf t
R2(config)#isdn switch-type basic-ni
R2(config)#interface bri0
R2(config-if)#ip address 172.12.21.2 255.255.255.252
R2(config-if)#isdn spid1 0835866201
R2(config-if)#isdn spid2 0835866401
R2(config-if)#no shut
```

The IP addresses, ISDN switch-type, and SPIDs are configured.

R1#show isdn status

Global ISDN Switchtype = basic-ni

ISDN BRI0 interface

 dsl 0, interface ISDN Switchtype = basic-ni

Layer 1 Status:

 ACTIVE

Layer 2 Status:

 TEI = 71, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED

 TEI = 72, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED

Spid Status:

spid1 configured, no LDN, spid1 sent, spid1 valid

 TEI 72, ces = 2, state = 5(init)

spid2 configured, no LDN, spid2 sent, spid2 valid

R2#show isdn status

The current ISDN Switchtype = basic-ni1

ISDN BRI0 interface

Layer 1 Status:

 ACTIVE

Layer 2 Status:

 TEI = 73, State = MULTIPLE_FRAME_ESTABLISHED

 TEI = 74, State = MULTIPLE_FRAME_ESTABLISHED

Spid Status:

 TEI 73, ces = 1, state = 5(init)

spid1 configured, no LDN, spid1 sent, spid1 valid

 TEI 74, ces = 2, state = 5(init)

spid2 configured, no LDN, spid2 sent, spid2 valid

“show isdn status” verifies the SPIDs are valid.

Troubleshooting ISDN with “show isdn status”

When you have problems arise with ISDN, show isdn status is the first command you should run. It displays a wealth of ISDN information.

Consider the examples shown in this lab. Reading from top to bottom, the first output of the command indicates what type of ISDN switch was configured on the router, if any. If you do not have a switchtype configured, your ISDN configuration will fail. In this case, your “show isdn status” output would look like this:

```
R2#show isdn status
**** No ISDN Switchtype currently defined ****
ISDN BRI0 interface
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
```

Simply configure the correct switchtype and then run “show isdn status” again. You will then see the correct switchtype identified here.

SPIDs can be temperamental as well. Once in a while, they will show as invalid when they have actually been configured correctly. This generally happens when SPIDs are changed, which rarely happens outside a lab environment.

This output of “show isdn status” displays a problem with the SPIDs:

```
R2#show isdn status
The current ISDN Switchtype = basic-ni1
ISDN BRI0 interface
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    Layer 2 NOT Activated
  Spid Status:
    TEI Not Assigned, ces = 1, state = 3(await establishment)
      spid1 configured, no LDN, spid1 NOT sent, spid1 NOT valid
    TEI Not Assigned, ces = 2, state = 1(terminal down)
      spid2 configured, no LDN, spid2 NOT sent, spid2 NOT valid
```

The first thing to check is the numbers configured as the SPIDs; a single error will render the SPIDs invalid.

If this fails, simply shut down the interface and then open it up again. This will resolve the issue if the SPIDs are correct. Do so and run “show isdn status” again to verify SPID validity.

Configure dialer map statements on R1 and R2, each mapping to the other router's BRI interface. Ping R1's BRI interface from R2.

```
R1#conf t  
R1(config)#interface bri0  
R1(config-if)#dialer map ip 172.12.21.2 name R2 broadcast 8358662
```

```
R2#conf t  
R2(config)#interface bri0  
R2(config-if)#dialer map ip 172.12.21.1 name R1 broadcast 8358661
```

```
R2#ping 172.12.21.1  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.21.1, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

The dialer map configuration is correct, but the pings do not go through.

The ping fails because there is no interesting traffic defined that will bring the line up. Using the **dialer-list** and **dialer-group** commands, allow any IP traffic to bring up the line. Ping R1 from R2. After the ping goes through, run **show dialer** to see what packets brought the line up.

*All IP traffic is defined as interesting traffic by the **dialer-list** command, and that list is called by the **dialer-group** command. The ping packets bring the line up.*

```
R1#conf t  
R1(config)#dialer-list 1 protocol ip permit  
R1(config)#interface bri0  
R1(config-if)#dialer-group 1
```

```
R2#conf t  
R2(config)#dialer-list 1 protocol ip permit  
R2(config)#interface bri0  
R2(config-if)#dialer-group 1
```

```
R2#ping 172.12.21.1  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.21.1, timeout is 2 seconds:  
.!!!!  
Success rate is 80 percent (4/5), round-trip min/avg/max = 36/37/40 ms  
R2#  
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up  
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up  
R2#  
%ISDN-6-CONNECT: Interface BRI0:1 is now connected to 8358661 R1
```

R2#**show dialer**

BRI0 - dialer type = ISDN

Dial String	Successes	Failures	Last called	Last status
8358661	2	0	00:00:04	successful

0 incoming call(s) have been screened.

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is data link layer up

Dial reason: ip (s=172.12.21.2, d=172.12.21.1)

Time until disconnect 117 secs

Connected to 8358661 (R1)

The dial reason in the output of “show dialer” clearly shows the source (s) and destination (d) of the packet that caused the line to dial. While it was obvious here why the line went up, routing protocols send multicasts and broadcasts that can cause such a line to dial and stay dialed for days, weeks, or even months at a time, which costs a great deal of money. This command is vital in diagnosing any issue involving an ISDN line that dials and stays up.

The routers will now authenticate each other with PAP over the ISDN link. Configure the global command **username / password** on each router, naming the remote router as the username and the password the remote router will be sending as the password. Use **encapsulation ppp** and **ppp authentication pap** to enable each router to authenticate the other. Have R1 send a password of CCNA and R2 to send a password of CISCO. Use the **ppp pap sent-username** command as shown in the following illustration.

Examining the configuration of PPP and PAP on R1 and R2.

```
R1#conf t
R1(config)#username R2 password CISCO
< The username and password the remote router will send is configured here for authentication. >
R1(config)#interface bri0
R1(config-if)#encapsulation ppp
11:21:19: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 71 changed to down
11:21:19: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 72 changed to down
11:21:19: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 71 changed to down
11:21:21: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 75 changed to up
11:21:21: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 76 changed to up
< The encapsulation is changed from the default of HDLC to PPP. The TEI will go down and then should come back up. If they do not come back up within half a minute, open and close the BRI interfaces. >
R1(config-if)#ppp authentication pap
< PAP authentication will be used to authenticate the remote router. >
R1(config-if)#ppp pap sent-username R1 password CCNA
< This is the username and password R1 will send R2. This is a mandatory command when using PAP authentication. >

R2#conf t
R2(config)#username R1 password CCNA
R2(config)#interface bri0
R2(config-if)#encapsulation ppp
ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 74 changed to down
ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 73 changed to down
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 77 changed to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 78 changed to up
R2(config-if)#ppp authentication pap
R2(config-if)#ppp pap sent-username R2 password CISCO
```

On R2, run the global command **debug ppp negotiation**, and ping R1's BRI interface.

Examining the PPP PAP negotiation from R2's perspective. The output of "debug ppp negotiation" has been truncated to show the PAP authentication process clearly.

R2#**debug ppp negotiation**

PPP protocol negotiation debugging is on

R2#**ping 172.12.21.1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.12.21.1, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 36/37/40 ms

%LINK-3-UPDOWN: Interface BRI0:1, changed state to up

BR0:1 PPP: Phase is AUTHENTICATING, by both

< Both routers are authenticating the other. >

BR0:1 PAP: O AUTH-REQ id 1 len 13 from "R2"

< R2 is sending an authentication request to R1. >

BR0:1 PAP: I AUTH-ACK id 1 len 5

< The "I" indicates an incoming packet; the remote route is acknowledging the authentication request. >

BR0:1 PAP: I AUTH-REQ id 1 len 12 from "R1"

< A PAP authentication request has been received from R1. >

BR0:1 PAP: Authenticating peer R1

< R1 is being authenticated. >

BR0:1 PAP: O AUTH-ACK id 1 len 5

< An acknowledgment of the PAP authentication request from R1 is sent. >

PAP authentication will now be removed and CHAP authentication configured in its place. On both physical interfaces, run **no encapsulation ppp** under the BRI interfaces, and remove the **username / password** statements.

Removing PPP encapsulation will also remove PAP authentication commands.

```
R1#conf t
R1(config)#no username R2 password CISCO
R1(config)#int bri0
R1(config-if)#no encapsulation ppp
20:59:31: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 75 changed to
down
20:59:31: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 76 changed to
down
20:59:31: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 75 changed to
down
20:59:34: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 79 changed to up
20:59:34: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 80 changed to up

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#no username R1 password CCNA
R2(config)#int bri0
R2(config-if)#no encapsulation ppp
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 78 changed to down
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 77 changed to down
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 77 changed to down
```

After changing the encapsulation type back to HDLC with the “no encapsulation ppp” command, the TEI goes down but does not come back up. This indicates the SPIDs are being seen as invalid. “show isdn status” shows this to be true:

```
R2#show isdn status
Layer 2 Status:
    Layer 2 NOT Activated
Spid Status:
    TEI Not Assigned, ces = 1, state = 3(await establishment)
        spid1 configured, no LDN, spid1 NOT sent, spid1 NOT valid
    TEI Not Assigned, ces = 2, state = 1(terminal down)
        spid2 configured, no LDN, spid2 NOT sent, spid2 NOT valid
```

The SPID values did not change, but are being seen as invalid. This is a common situation after changing encapsulation types. Simply shut the interface with “shut” and open it with “no shut”, and the SPIDs will be fine.

Opening and shutting the BRI interface after changing encapsulation types restores the SPID status to “valid”.

```
R2(config)#interface bri0
R2(config-if)#shut
%LINK-5-CHANGED: Interface BRI0, changed state to administratively down
R2(config-if)#no shut
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:2, changed state to down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:2, changed state to down
%LINK-3-UPDOWN: Interface BRI0, changed state to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 81 changed to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 82 changed to up
```

```
R2#show isdn status
Layer 2 Status:
  TEI = 81, State = MULTIPLE_FRAME_ESTABLISHED
  TEI = 82, State = MULTIPLE_FRAME_ESTABLISHED
Spid Status:
  TEI 81, ces = 1, state = 5(init)
    spid1 configured, no LDN, spid1 sent, spid1 valid
    Endpoint ID Info: epsf = 0, usid = 2, tid = 1
  TEI 82, ces = 2, state = 5(init)
    spid2 configured, no LDN, spid2 sent, spid2 valid
    Endpoint ID Info: epsf = 0, usid = 4, tid = 1
```

Configure the routers for CHAP authentication. The switch-type, SPIDs, dialer map statements, and dialer-lists have already been configured. On both R1 and R2, configure a **username / password** statement with the password **GOTMYCCNA**. Configure both routers for PPP encapsulation and CHAP authentication with the **encapsulation ppp** and **ppp authentication chap** commands.

Configuring PPP encapsulation and CHAP authentication on both R1 and R2.

```
R1#conf t
R1(config)#username R2 password GOTMYCCNA
R1(config)#interface bri0
R1(config-if)#encapsulation ppp
21:17:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 79 changed to
down
21:17:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 80 changed to
down
21:17:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 79 changed to
down
21:17:56: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 83 changed to up
21:17:56: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 84 changed to up
R1(config-if)#ppp authentication chap
```

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#username R1 password GOTMYCCNA
R2(config)#interface bri0
R2(config-if)#encapsulation ppp
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 82 changed to down
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 81 changed to down
%ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 81 changed to down
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 85 changed to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 86 changed to up
R2(config-if)#ppp authentication chap
```

It Bears Repeating... CHAP and PAP Passwords

CHAP: Both routers must use the same password.

PAP: Routers can use different passwords, configured with the **ppp pap sent-username** command.

Run **debug ppp negotiation** on R2, and ping R1's BRI interface.
After the line comes up, run **show dialer** to see the cause of the call.

Examining the CHAP authentication process with “debug ppp negotiation”.

```
R2#debug ppp negotiation
PPP protocol negotiation debugging is on
R2#ping 172.12.21.1

BR0:1 PPP: Phase is AUTHENTICATING, by both
< Both routers are authenticating the other with CHAP. >
BR0:1 CHAP: O CHALLENGE id 1 len 23 from "R2"
< R2 is sending a challenge to the remote router. O = "outgoing". >
BR0:1 CHAP: I CHALLENGE id 1 len 23 from "R1"
< A CHAP challenge has come in from R1. I = "incoming">
BR0:1 CHAP: O RESPONSE id 1 len 23 from "R2"
< R2 is sending a response to the challenge. >
BR0:1 CHAP: I SUCCESS id 1 len 4
< A success message has come in , indicating the challenge was successfully met. >
BR0:1 CHAP: I RESPONSE id 1 len 23 from "R1"
< A response to the challenge sent to R1 is now coming in. >
BR0:1 CHAP: O SUCCESS id 1 len 4
< The challenge was successfully met by R1, and a success message is sent. >
```

```
R2#show dialer
BRI0 - dialer type = ISDN
```

Dial String	Successes	Failures	Last called	Last status
8358661	4	0	00:00:12	successfu

0 incoming call(s) have been screened.

```
BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=172.12.21.2, d=172.12.21.1)
Time until disconnect 109 secs
Connected to 8358661 (R1)
```

```
BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
```

The ping packet from R2 was the cause of the line dialing.

Using **ppp multilink** and **dialer load-threshold**, configure the ISDN interface on R1 to bring up the second B-channel when the first B-channel reaches 50% of its outbound capacity.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface bri0
R1(config-if)#ppp multilink
R1(config-if)#dialer load-thresh 127 ?
    either  Threshold decision based on max of inbound and outbound traffic
    inbound  Threshold decision based on inbound traffic only
    outbound Threshold decision based on outbound traffic only
<cr>
R1(config-if)#dialer load-thresh 127 outbound
```

The IOS Help feature is activated by entering part of all of a command and entering "?" (without the quotation marks). IOS Help is very helpful in seeing all the options and values a command may have.

PPP was already running on the BRI interface, so the only command needed to activate multilink is the **ppp multilink** command. The router must now be told at what point to bring up the additional link.

The **dialer load-threshold** command is used for this purpose, but care must be used in entering this command. The possible values for this command are 1 – 255, **not** 1 – 100. The value entered with this command represents a percentage of 255, not 100.

To configure the desired value for this threshold, take the percentage of line capacity at which the link should come up, convert it to decimal, and multiply by 255.

In this configuration, the second B-channel should come up when the first line reaches 50% of its capacity. 50% is .50 in decimal format; multiply 255 by .50, and the result is 127.5. The value of **dialer load-threshold** must be a round number, so round the result up or down before entering.

A dialer profile will now be configured on R1. On the BRI interface, remove the following: the PPP encapsulation type, the dialer-map statement, the dialer-group statement, the dialer-load statement, the IP address, and any commands referencing PAP or CHAP authentication.

The **ISDN switch-type** command and **username / password** command should remain.

Removing the appropriate statements from the BRI interface on R1.

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#interface bri0  
R1(config-if)#no encapsulation ppp  
21:45:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 83 changed to  
down  
21:45:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 84 changed to  
down  
21:45:54: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 83 changed to  
down  
21:45:56: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 87 changed to up  
21:45:56: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 88 changed to up  
R1(config-if)#no dialer map ip 172.12.21.2 name R2 broadcast 8358662  
R1(config-if)#no dialer-group 1  
R1(config-if)#no dialer load-threshold 127 outbound  
R1(config-if)#no ip address
```

Make sure the TEI comes back up after going down. If it does not, shut and reopen the BRI interface.

After removing these statements, the running config should show this for the BRI interface:

```
interface BRI0  
no ip address  
isdn switch-type basic-ni  
isdn spid1 0835866101  
isdn spid2 0835866301
```

The SPIDs are always configured on the physical interface.

Configure a dialer profile with the command **interface dialer 1** on R1. The IP address that was on the BRI interface will be placed on this logical interface. Use **dialer remote-name** to indicate the name of

the remote router to be dialed, and **dialer string** to configure the number to be dialed.

*The logical interface **dialer1** is configured.*

```
R1#conf t
R1(config)#interface dialer 1
R1(config-if)#ip address 172.12.21.1 255.255.255.252
R1(config-if)#dialer remote-name R2
R1(config-if)#dialer string 8358662
```

The dialer-list defining all IP traffic as interesting is still present. Place the **dialer-group 1** command on interface dialer1 so interesting traffic will bring the logical interface up.

```
R1#conf t
R1(config)#interface dialer1
R1(config-if)#dialer-group 1
```

The physical BRI interface and logical Dialer interface must now be linked. Configure Dialer1 with the **dialer pool 1** command, then make the BRI interface a member of that pool with the **dialer pool-member 1** command.

```
R1#conf t
R1(config)#interface dialer1
R1(config-if)#dialer pool 1

R1#conf t
R1(config)#interface bri0
R1(config-if)#dialer pool-member 1
```

R2 is still using PPP encapsulation and CHAP authentication; R1 must also. On **both** the physical and logical interfaces, configure **encapsulation ppp** and **ppp authentication chap**.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface bri0
R1(config-if)#encapsulation ppp
22:05:47: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 87 changed to
down
22:05:47: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 88 changed to
down
22:05:47: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 87 changed to
down
22:05:50: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 89 changed to up
22:05:50: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 90 changed to up
R1(config-if)#ppp authentication chap

R1(config)#interface dialer1
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
```

When the encapsulation type is changed on the physical interface, the TEI goes up and down, as expected. If the TEI doesn't come back up, open and shut the physical interface. No such "up / down" behavior will occur when the encapsulation type is configured on the logical interface.

```

R1#debug ppp negotiation
PPP protocol negotiation debugging is on
R1#ping 172.12.21.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.21.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 36/36/36 ms

22:12:07: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
22:12:07: %DIALER-6-BIND: Interface BRI0:1 bound to profile Dialer1
22:12:07: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 8358662
22:12:07: BR0:1 PPP: Phase is AUTHENTICATING, by both
22:12:07: BR0:1 CHAP: O CHALLENGE id 3 len 23 from "R1"
22:12:07: BR0:1 CHAP: I CHALLENGE id 3 len 23 from "R2"
22:12:07: BR0:1 CHAP: O RESPONSE id 3 len 23 from "R1"
22:12:07: BR0:1 CHAP: I SUCCESS id 3 len 4
22:12:07: BR0:1 CHAP: I RESPONSE id 3 len 23 from "R2"
22:12:07: BR0:1 CHAP: O SUCCESS id 3 len 4
22:12:07: BR0:1 PPP: Phase is UP

```

< The expected series of challenges, responses, and successes occur. >

```

R1#show dialer
BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=172.12.21.1, d=172.12.21.2)
Interface bound to profile Dialer1
Time until disconnect 112 secs
Current call connected 00:00:10
Connected to 8358662 (R2)

```

```

Dialer1 - dialer type = DIALER PROFILE
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up

```

The BRI physical interface is bound to Dialer1, the logical interface, and the status of the Dialer Profile is up as well.

Using IOS Help To Avoid ISDN Misconfigurations

There is no standard for how values are entered with various Cisco commands. Some are entered in seconds, some are entered in minutes; some are entered as percentages of 100, some as percentages of other numbers. When you begin learning about Cisco Quality Of Service, which is basically a way to allot bandwidth throughout your network, you'll see several different measurements of data used with those commands.

A good habit to get into now is to use Cisco IOS Help to make absolutely sure you know what unit of time or data is being used with a particular command. Particularly infamous is the dialer load-threshold command, which uses a value that is a percentage of 255, but many people rush through it thinking it's a percentage of 100:

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int dialer1
R1(config-if)#dialer load-threshold ?
  <1-255> Load threshold to place another call

R1(config-if)#dialer load-threshold
```

On occasion, the IOS will not indicate what the value is, as with dialer idle-timeout:

```
R1(config)#
2d22h: %SYS-5-CONFIG_I: Configured from console by console
R1(config)#int dialer1
R1(config-if)#dialer idle-timeout ?
  <1-2147483> Idle timeout before disconnecting a call

R1(config-if)#dialer idle-timeout
```

Notice that the IOS Help doesn't indicate whether that value is seconds or minutes. That doesn't happen often, but it does happen. If this happens, simply look the command up at www.cisco.com/univercd to check what the value represents.

Good habits start at the CCNA level. Develop the habit now of double-checking Cisco's measurement for any numerical value you enter. It only takes a moment and can spare you a lot of trouble over a misconfiguration.

Notes

Section Five: Binary Math and Subnetting

The Importance Of Knowing Binary Math

Do not rely on chart memorization when converting binary values to decimal, and vice versa, or when subnetting. Performing binary math conversions is simply a matter of addition and subtraction, and true mastery of binary math is achieved through practice, not memorization. The ability to perform these conversions and to subnet is imperative not only for the CCNA exam, but for success on CCNP and CCIE exams, and in a real-world job. CCNAs who truly understand binary math have a huge advantage over those who memorized a chart long enough to get through the exam and then promptly forgot it.

Converting Decimal Values To Binary

Consider the default mask for a Class A network, 255.0.0.0. Broken down at the bit level, the value is derived in this fashion:

	128	64	32	16	8	4	2	1
First Octet	1	1	1	1	1	1	1	1
Second Octet	0	0	0	0	0	0	0	0
Third Octet	0	0	0	0	0	0	0	0
Fourth Octet	0	0	0	0	0	0	0	0

To convert a decimal value into a binary value, take the binary number and, working from left to right, determine whether subtracting the value of the binary column from the decimal value would result in a positive remainder or negative number.

- A. If subtracting the binary value from the decimal value would result in a positive remainder, subtract that value, put a "1" under that binary value column, and repeat the operation with the next binary value until you reach the end of the column or your binary value is zero.

- B. If subtracting the binary value from the decimal value would result in a negative number, place a "0" in the binary column, and repeat the operation with the next column's value.

That sounds complicated, but after going through some examples, it becomes obvious that decimal-to-binary conversion is simply a matter of addition and subtraction.

With subnetting, decimal values will always be broken down into binary using this chart:

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Taking the example of a Class A mask, 255.0.0.0, draw a chart with those values at the top, and put four rows under it, one for each octet.

	128	64	32	16	8	4	2	1
First Octet								
Second Octet								
Third Octet								
Fourth Octet								

The first octet value is 255. Working from left to right, can 128 be subtracted from 255? Yes, and it leaves 127. Put a "1" under "128" in the first row.

	128	64	32	16	8	4	2	1
First Octet	1							

Move to the next value, left to right. Can 64 be subtracted from 127? Yes, and it leaves 63. Put a "1" under "64" in the first row.

	128	64	32	16	8	4	2	1
First Octet	1	1						

Repeating this operation from left to right, every column has a "1" in it to represent the value to 255. The decimal value 255 is successfully converted to the binary value 11111111.

When the decimal-to-binary conversion results in a remainder of zero before finishing the left-to-right operation, all remaining columns will have a value of zero.

For example, consider the subnet mask 248.0.0.0. Working from left to right, convert this mask to binary value.

	128	64	32	16	8	4	2	1
First Octet								

Work from left to right. Can 128 be subtracted from 248? Yes, with a remainder of 120. Place a "1" in the "128" column.

	128	64	32	16	8	4	2	1
First Octet	1							

Moving left to right, the next binary value to consider is 64. Can 64 be subtracted from 120? Yes, with a remainder of 56. Place a "1" in the "64" column.

	128	64	32	16	8	4	2	1
First Octet	1	1						

The next binary value to consider is 32. Can 32 be subtracted from 56? Yes, with a remainder of 24. Place a "1" in the "32" column.

	128	64	32	16	8	4	2	1
First Octet	1	1	1					

The next value is 16. Can 16 be subtracted from 24? Yes, with a remainder of 8. Place a "1" in the "16" column.

	128	64	32	16	8	4	2	1
First Octet	1	1	1	1				

The remainder is 8; the next value to consider is 8. Can 8 be subtracted from 8? Yes, with a remainder of zero. Place a "1" in the "8" column.

	128	64	32	16	8	4	2	1
First Octet	1	1	1	1	1			

The remainder is now zero, so the decimal value conversion has completed. Place a "0" in each of the remaining columns.

	128	64	32	16	8	4	2	1

First Octet	1	1	1	1	1	0	0	0
-------------	---	---	---	---	---	---	---	---

The decimal 248 has been successfully converted into the binary number 11111000. The second, third, and fourth octets are all represented by the decimal 0, which is represented by the binary value 00000000. The conversion of 248.0.0.0 into binary value results in a final value of 11111000 00000000 00000000 00000000.

Another example: Consider the subnet mask 240.0.0.0. Create the chart with the binary bit values, and work from left to right to convert the decimal value 240 to binary:

	128	64	32	16	8	4	2	1
First Octet								

Subtracting 128 from 240 results in a remainder of 112. Place a "1" in the "128" column:

	128	64	32	16	8	4	2	1
First Octet	1							

Subtracting 64 from the remainder 112 results in 48. Place a "1" in the "64" column.

	128	64	32	16	8	4	2	1
First Octet	1	1						

Subtracting 32 from the remainder 48 results in 16. Place a "1" in the "32" column.

	128	64	32	16	8	4	2	1
First Octet	1	1	1					

Subtracting 16 from the remainder 16 results in zero. Place a "1" in the "16" column.

	128	64	32	16	8	4	2	1
First Octet	1	1	1	1				

The remainder is now zero, indicating that the decimal-to-binary conversion is complete. Place a "0" in all remaining columns to complete the binary number.

	128	64	32	16	8	4	2	1

First Octet	1	1	1	1	0	0	0	0
-------------	---	---	---	---	---	---	---	---

The conversion is complete. The decimal value 240 has been successfully converted to binary value 11110000. The last three octets are all zeroes; the binary conversion for the entire mask is 11110000 00000000 00000000 00000000.

Converting Binary To Dotted Decimal

Converting a binary value to a dotted decimal value is comparatively simple. Consider this binary value: 11111111 11110001 00000000 00000000.

Draw the same chart used for decimal-to-binary conversion:

	128	64	32	16	8	4	2	1
First Octet								
Second Octet								
Third Octet								
Fourth Octet								

Simply fill in the chart bit by bit with the 1s and 0s as presented to you.

	128	64	32	16	8	4	2	1
First Octet	1	1	1	1	1	1	1	1
Second Octet	1	1	1	1	0	0	0	1
Third Octet	0	0	0	0	0	0	0	0
Fourth Octet	0	0	0	0	0	0	0	0

For each octet, add all values represented by "1". For the first octet, the decimal value is $128+64+32+16+8+4+2+1$, which equals 255. The second octet is $128+64+32+16+1$, equaling 241. The third and fourth octets are all zeroes, so our dotted decimal value from the supplied binary value is 255.241.0.0.

IP Addressing And Subnetting

To fully understand subnetting, it is vital to know how many network and host bits are present in Class A, Class B, and Class C networks.

	First Octet	Network Bits	Host Bits	Default Mask
Class A	1 – 126	8	24	255.0.0.0
Class B	128 – 191	16	16	255.255.0.0

Class C	192 - 223	24	8	255.255.255.0
---------	-----------	----	---	---------------

"Network bits" indicates how many bits represent the network number when the dotted decimal number is converted to binary. "Host bits" indicates how many bits represent hosts.

For example, the network address 17.1.1.1 falls into the Class A category. Breaking the dotted decimal down into bits, the first 8 bits of this address represent the network, and the remaining 24 bits represent the host:

Network Address 17.1.1.1 Broken Down Into Network And Host Bits:

Dotted Decimal	17	1	1	1
Octets In Binary	00010001	00000001	00000001	00000001
Network / Host Bits	Network	Host	Host	Host

The Class A network address 17.1.1.1 has eight network bits and 24 host bits.

Breaking down network address 150.10.10.10, a Class B network address:

Dotted Decimal	150	10	10	10
Octets In Binary	10010110	00001010	00001010	00001010
Network / Host Bits	Network	Network	Host	Host

The Class B network address 150.10.10.10 has 16 network bits and 16 host bits.

Breaking down network address 200.24.24.24, a Class C network address:

Dotted Decimal	200	24	24	24
Octets In Binary	11001000	00011000	00011000	00011000
Network / Host Bits	Network	Network	Network	Host

The Class C network address 200.24.24.24. has 24 network bits and 8 host bits.

Subnetting is simply a process of "borrowing" host bits in order to create the subnet portion of an address. Note in the following

examples that the network portion of the address never changes. The subnet field always borrows from the host bits.

Why Use Subnetting?

Subnetting is a highly effective method of conserving IP addresses. Consider a point-to-point ISDN connection with two host addresses, one on each side of the connection. Using an entire Class C address range for this network segment would be a waste of addresses. A default Class C network mask of 255.255.255.0 yields 254 usable host addresses, but only two are needed for this small network.

Subnetting allows the use of a "tighter" subnet mask than the default; that is, one that yields a smaller amount of network addresses. The benefit is that the addresses that would have been wasted are now still usable by other segments of the network.

Determining The Number Of Valid Subnets

To determine the number of valid subnets for a given network number and mask, use this formula:

$$\text{Number of subnets} = (2 \text{ squared by the number of subnet bits}) - 2$$

The number of subnet bits is determined by examining the default network mask for that class, and comparing it to the actual network mask. Taking network 172.20.20.0 255.255.255.0 as an example, the default mask for this Class B network is 255.255.0.0. Write out the default mask and the actual mask in binary:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Default Mask 255.255.0.0	11111111	11111111	00000000	00000000
Subnet Mask 255.255.255.0	11111111	11111111	11111111	00000000
Type and Number Of Bits	All Network Bits	All Network Bits	All Subnet Bits	All Host Bits

The process for determining the network, subnet, and host bits is as follows:

- A. The rules for Class A, Class B, and Class C determine the network portion.
- B. Compare the remaining bits of the default and actual mask. The portion where the Default Mask and Subnet Mask both have zeroes is the host bits portion of the address.
- C. The remaining bits – where the Default Mask has zeroes but the Subnet Mask has one – are the subnet bits. These are the bits that are being “borrowed” from the host bits.

After determining there are eight subnet bits, the formula dictates that the number of subnets equals (2 to the 8th power) – 2, which is 254. The network 172.20.20.0 255.255.255.0 has 254 usable subnets.

Prefix Notation

Prefix notation is a quicker way of stating what the subnet mask of a network is. Note that the subnet mask consists of a number of consecutive 1s. Prefix notation is simply a slash followed by the number of 1s in the subnet mask. The subnet mask 255.255.255.0 is instead referred to as “slash 24” when spoken, or “/24” when written. This is the generally accepted way subnet masks are spoken and written about.

Why Subtract Two?

The two subnets you are subtracting at the end of this formula are the “zero subnet” (all binary zeroes) and the “broadcast subnet” (all binary ones).

Both the “all-zeroes” and “all-ones” subnets are available for use on a Cisco router, but Cisco recommends you not use them. For exam purposes, use the formula as shown and do not consider either of these networks to be valid.

It's relatively easy to spot the subnet masks when the mask ends where the octets ends, such as 255.255.0.0 for a Class A network or 255.255.255.0 for a Class B network. For a network mask such as

255.255.240.0, determining the number of subnet bits is done by again comparing the default mask to the subnet mask, bit by bit.

Consider network 95.0.0.0 using the subnet mask 255.255.240.0. This is a Class A network with a default mask of 255.0.0.0. Write out the default mask and the subnet mask and compare the bits:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Default Mask 255.0.0.0	11111111	00000000	00000000	00000000
Subnet Mask 255.255.240.0	11111111	11111111	11110000	00000000
Network / Subnet / Host Bits	8 Network Bits	8 Subnet Bits	4 Subnet, 4 Host Bits	8 Host Bits

Review the rules for network, subnet, and host bits:

- A. The rules for Class A, Class B, and Class C determine the network portion.
- B. Compare the remaining bits of the default and actual mask. The portion where the Default Mask and Subnet Mask both have zeroes is the host bits portion of the address.
- C. The remaining bits – where the Default Mask has zeroes but the Subnet Mask has one – are the subnet bits. These are the bits that are being “borrowed” from the host bits.

The network is Class A, so the first 8 bits are the network portion. The last 12 bits of both masks are zeroes, so those are the host bits. The remaining 12 bits – the bits that are zeroes in the Default Mask and ones in the Subnet Mask – are subnet bits.

Multiply 2 to the 12th power per the subnet mask formula, then subtract 2. The result is that the network 95.0.0.0 255.255.240.0 (or 95.0.0.0 /20) has 4094 usable subnets.

Determining The Number Of Valid Hosts

To determine the number of valid hosts, a somewhat similar formula is used:

Valid Hosts On A Subnet = (2 squared by number of host bits) – 2

The method that was used to determine the number of subnet bits is the same that is used to determine the number of host bits. Consider the network 150.50.50.0 /24. This is a Class B network with a default mask of 255.255.0.0 (/16). Convert the Default Mask and Subnet Mask into binary and determine the network, subnet, and host bits.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Default Mask 255.255.0.0	11111111	11111111	00000000	00000000
Subnet Mask 255.255.255.0	11111111	11111111	11111111	00000000
Number And Type Of Bits	All Network	All Network	All Subnet	All Host

150.50.50.0 /24 consists of 16 network bits because of its Class B classification. The Default and Subnet masks have all zeroes in all 8 bits of the final octet; these 8 bits are the host bits. The bits where the Default Mask has zeroes and the Subnet Mask has ones are the subnet bits.

Using the formula to determine the number of valid hosts, (2 to the 8th power) minus 2 yields 254. The network 150.50.50.0 /24 has 254 valid hosts.

Why Subtract Two?

The formulas for determining the number of valid hosts and for determining the number of valid subnets both subtract two at the end. For the “valid subnets” formula, the two subnets being subtracted are the “all-zeroes” and “all-ones” subnets. When determining valid hosts, the two hosts subtracted from the final answer account for the network number and the broadcast address for that subnet, neither of which should be considered a valid host.

Determining The Subnet Number Of A Given IP Address

Given an IP address and subnet mask, determining the subnet it resides on is accomplished by performing a Boolean AND operation.

First, the IP address and its subnet mask will be converted to binary. The Boolean AND is simply a bit-by-bit comparison of the address and the subnet mask.

If both bits are 1, the result of the Boolean AND is 1. If a 0 is set for that bit on either the subnet mask or IP address, or both, the result of the AND is a 0.

Consider network address 178.56.21.9 /24. Convert the IP address and the subnet mask into binary, remembering that /24 is equivalent to 255.255.255.0:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 178.56.21.9	10110010	00111000	00010101	00001001
SNMask 255.255.255.0	11111111	11111111	11111111	00000000
AND Result	10110010	00111000	00010101	00000000

Converting the AND result back to decimal format will yield the subnet this address resides on.

	128	64	32	16	8	4	2	1	Sum
1 st Octet	1	0	1	1	0	0	1	0	178
2 nd Octet	0	0	1	1	1	0	0	0	56
3 rd Octet	0	0	0	1	0	1	0	1	21
4 th Octet	0	0	0	0	0	0	0	0	0

The decimal dotted subnet this IP address resides on is 178.56.21.0 /24.

Another example of determining the subnet of an IP address/ subnet mask combination: Consider network address 200.154.150.89 /27. A /27 mask refers to the first 27 bits of the binary address being 1s, which would look like this:

11111111 11111111 11111111 11100000 = 255.255.255.224

Convert the IP address and the subnet mask to binary, and run a Boolean AND on them:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IPAdd 200.154.150.89	11001000	10011010	10010110	01011001

SNMask 255.255.255.224	11111111	11111111	11111111	11100000
AND Result	11001000	10011010	10010110	01000000

Convert the AND result to decimal: 200.154.150.64. The IP address 200.154.150.89 /27 resides on subnet 200.154.150.64 /27.

Determining The Range Of Valid Host Addresses On A Subnet

To determine the range of valid host addresses on a subnet, first determine how many overall host addresses are on that subnet. The first address in the range is the network number and is not a valid host address; the final address in the range is the broadcast address for that subnet and is not a valid host address. All addresses between the two are valid host addresses.

Consider the network 200.154.150.64 /27. To determine the valid host addresses on this subnet, first determine how many host bits there are by converting the dotted decimal IP address and mask to binary.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 200.154.150.64	11001000	10011010	10010110	01000000
Mask 255.255.255.224	11111111	11111111	11111111	11100000
Host Bits				5 Host

The subnet mask /27 indicates that there are only 5 host bits. (32 overall bits – 27 bits in the mask.) If those last five bits are all zeroes, the resulting address is 200.154.150.64. This is the network address and is not valid for hosts.

If those last five bits are ones, the resulting address is 200.154.150.95. This is the broadcast address, again not valid for hosts.

The range of addresses between the two, 200.154.150.65 – 94, is the acceptable range for host addresses on this subnet.

Meeting Stated Design Requirements

Consider this question:

"Your network uses Class B network 165.10.0.0. You need at least 150 subnets that have no more than 200 hosts apiece. Which of the following subnet masks should you use?"

Remember the formulas for determining the number of subnets, or the number of valid hosts per subnet:

Number of subnets = (2 to the n th power) – 2, where n equals the number of subnet bits.

Number of valid hosts on a subnet = (2 to the n th power) – 2, where n equals the number of host bits.

Network 165.10.0.0 is a Class B network with a default mask of 255.255.0.0. Sixteen bits are being used for the network address, leaving another 16 bits to be divided between the subnet and host bits.

A direct method of determining how many bits are needed for the subnet mask is to simply write 2x2 on paper, yielding 4, and continue doubling until you reach the necessary number of subnets. The number of twos you have to involve will be the number of subnet bits.

Example: For 150 subnets:

$$2 \times 2 = 4 - 2 = 2$$

$$2 \times 2 \times 2 = 8 - 2 = 6$$

$$2 \times 2 \times 2 \times 2 = 16 - 2 = 14$$

$$2 \times 2 \times 2 \times 2 \times 2 = 32 - 2 = 30$$

$$2 \times 2 \times 2 \times 2 \times 2 \times 2 = 64 - 2 = 62$$

$$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 128 - 2 = 126$$

$$2 \times 2 = 256 - 2 = 254$$

This only takes a few seconds to do on paper and is an excellent illustration of how many subnet bits are needed. Remember to

Chris Bryant, CCIE #12933

150

www.thebryantadvantage.com

© 2004 The Bryant Advantage

subtract two at the end of the formula to account for the all-zeroes and all-ones subnets.

Eight subnet bits will give 254 usable subnets. The question asked for 150, so that requirement is covered. However, since there would be eight bits left for the hosts, that would also yield 254 hosts. The question asked for the subnets to allow for no more than 200 hosts.

Using seven bits for the hosts would result in 126 hosts. That would leave nine subnet bits, resulting in 510 usable subnets. Since the question requested a minimum number of subnets, and made no mention of a maximum number, the correct response would be to have nine subnet bits and seven host bits, resulting in a mask of 255.255.255.128, or /25.

Binary Math Q&A

1. What is the binary equivalent of the IP address 217.39.41.200?

- A. **11011001 00110011 00101001 11110000**
- B. **11011001 00100111 00101001 11001001**
- C. **11011001 00100111 00101001 11001000**
- D. **11111001 00100111 00101000 11001000**

Answer: C. Use this chart to convert a dotted decimal address to its binary equivalent.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
1st Octet: 217	1	1	0	1	1	0	0	1	11011001
2 nd Octet: 39	0	0	1	0	0	1	1	1	00100111
3 rd Octet: 41	0	0	1	0	1	0	0	1	00101001
4th Octet: 200	1	1	0	0	1	0	0	0	11001000

2. Which of the following is true of a Class A address? Choose two.

- A. **It has a default mask of 255.0.0.0.**
- B. **It has a default mask of 255.255.0.0**
- C. **It has a default mask of /8.**
- D. **It has a default mask of /16.**
- E. **It has a default mask of /6.**

ANSWER: A, C. Class A addresses have a default mask of 255.0.0.0, also referred to as "slash eight" or /8, since the mask has 8 bits.

3. What is the binary equivalent of the IP address 145.68.245.251?

- A. **11110000 10101111 11001000 11111011**
- B. **10010001 01000100 11110101 11111011**
- C. **11001011 01110101 11111100 11110111**
- D. **11110101 00011111 11111010 11110101**

ANSWER: B. Use the following chart to convert a dotted decimal address to its binary equivalent.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
1 st Octet: 145	1	0	0	1	0	0	0	1	10010001
2 nd Octet: 68	0	1	0	0	0	1	0	0	01000100
3 rd Octet: 245	1	1	1	1	0	1	0	1	11110101
4 th Octet: 251	1	1	1	1	1	0	1	1	11111011

4. What network mask is indicated by the term "slash sixteen"? Choose two.

- A. **255.255.0.0**
- B. **0.0.255.255**
- C. **255.255.255.0**
- D. **The default mask for a Class A network.**
- E. **The default mask for a Class B network.**
- F. **The default mask for a Class C network.**

ANSWER: A, E. "Slash sixteen" refers to a mask of /16, meaning the first 16 bits are set to 1. The dotted decimal equivalent of /16 is 255.255.0.0, which is also the default network mask for a Class B network.

5. What is the binary equivalent of the IP address 212.48.1.254?

- A. **11110011 11101110 01111111 11111110**
- B. **11110111 00111100 01111111 00111010**
- C. **11010100 00110000 00000001 11111110**
- D. **11010100 00110000 11111110 00000001**

ANSWER: C. Use this chart to convert an IP address to its binary equivalent.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
1 st Octet: 212	1	1	0	1	0	1	0	0	11010100
2 nd Octet: 48	0	0	1	1	0	0	0	0	00110000
3 rd Octet: 1	0	0	0	0	0	0	0	1	00000001
4 th Octet: 254	1	1	1	1	1	1	1	0	11111110

6. You are configuring a point-to-point ISDN connection. To conserve addresses, you want to assign this subnet the smallest amount of host addresses possible. What subnet mask should be used for this subnet?

- A. /30
- B. /31
- C. /32
- D. 255.255.255.252
- E. 255.255.255.253
- F. 255.255.255.254

ANSWER: A, D. A subnet mask of 255.255.255.252, also referred to as /30 or "slash thirty", will result in two usable host addresses. This will be enough to have one valid address on each end of the ISDN connection.

7. What is the binary equivalent of the IP address 100.101.45.32?

- A. 01100100 01100101 00101101 00100000
- B. 01100110 01101101 00101010 00100000
- C. 01101010 01101010 01010100 00100000
- D. 01101110 01101011 01110110 00100000

ANSWER: A. Use this chart to convert a dotted decimal address to its binary equivalent.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
1 st Octet: 100	0	1	1	0	0	1	0	0	01100100
2 nd Octet: 101	0	1	1	0	0	1	0	1	01100101
3 rd Octet: 45	0	0	1	0	1	1	0	1	00101101
4 th Octet: 32	0	0	1	0	0	0	0	0	00100000

8 Which of the following is true of a Class C network? Choose three.

- A. Class C networks have a default network mask of 255.255.255.0.
- B. Class C networks have a default network mask of 255.255.0.0.
- C. Class C networks have a default network mask of /24.
- D. Class C networks have a default network mask of /20.
- E. 200.200.200.0 is a Class C network.
- F. 150.150.150.0 is a Class C network.
- G. 10.10.10.0 is a Class C network.

ANSWER: A, C, E. Class C networks have a default network mask of 255.255.255.0, or /24, and have a first-octet range of 192 – 223.

9. What statement is true of the network 175.10.10.0 /24?

- A. It is a Class B network with sixteen network bits, sixteen subnet bits, and no host bits.
- B. It is a Class B network with sixteen network bits, eight subnet bits, and eight host bits.
- C. It is a Class B network with eight network bits, sixteen subnet bits, and eight host bits.
- D. With the information given, it is not possible to determine how many subnet bits the network has.

ANSWER: B. This is a Class B network with sixteen network bits, eight subnet bits, and eight host bits.

The rules for determining the network, subnet, and host bits:

- A. The rules for Class A, Class B, and Class C determine the network portion.
- B. Compare the remaining bits of the default and actual mask. The portion where the Default Mask and Subnet Mask both have zeroes is the host bits portion of the address.
- C. The remaining bits – where the Default Mask has zeroes but the Subnet Mask has one – are the subnet bits. These are the bits that are being “borrowed” from the host bits.

The default network mask for a Class B network is 255.255.0.0.
 Convert the Default Mask and the Subnet Mask (255.255.255.0) into binary:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Default Mask: 255.255.0.0	11111111	11111111	00000000	00000000
Subnet Mask: 255.255.255.0	11111111	11111111	11111111	00000000
Type / Number Of Bits	8 Network	8 Network	8 Subnet	8 Host

Total: 16 Network bits, 8 subnet bits, and 8 host bits.

10. What is the binary equivalent of the dotted decimal address 123.45.67.89?

- A. 01111011 00111111 01000011 01011001
- B. 01111011 00111000 01110001 01010010
- C. 01111111 00110101 01100011 00101010
- D. 01111011 00101011 01000011 01011001

ANSWER: D. Use this chart to convert a dotted decimal number to its binary equivalent.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
1 st Octet: 123	0	1	1	1	1	0	1	1	01111011
2 nd Octet: 45	0	0	1	0	1	0	1	1	00101011
3 rd Octet: 67	0	1	0	0	0	0	1	1	01000011
4 th Octet: 89	0	1	0	1	1	0	0	1	01011001

11. The calculation for finding the number of valid subnets in a given subnet is $(2 \text{ to the } n\text{th power}) - 2$. What value does the "n" represent?

- A. The number of subnet masks.
- B. The number of subnet bits.
- C. The number of host bits.
- D. The number of network bits.

ANSWER: B. To calculate the number of valid subnets, find the number of subnet bits and put that in for "n" in the formula (2 to the nth power) – 2.

12. The calculation for finding the number of valid subnets in a given subnet is (2 to the nth power) – 2. Why is 2 subtracted from that value?

- A. Two subnets, the all-ones and all-broadcast subnets, cannot be used.
- B. Two subnets, the all-zeroes and all-ones subnets, cannot be used.
- C. It is good design practice to keep two subnets reserved for future use.
- D. Two subnets must be used by the IOS for management purposes and cannot be populated by hosts.

ANSWER: B. The all-zeroes and all-ones subnets cannot be used, and should not be counted among the valid subnets.

13. What is the dotted decimal equivalent of the binary IP address 11001001 00001111 01001011 00011111?

- A. 201.16.75.31
- B. 201.15.75.31
- C. 217.17.67.31
- D. 215.15.175.32

ANSWER: B. Use this chart to convert a binary IP address to its dotted decimal equivalent.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11001001	1	1	0	0	1	0	0	1	201
2 nd : 00001111	0	0	0	0	1	1	1	1	15
3 rd : 01001011	0	1	0	0	1	0	1	1	75
4 th : 00001111	0	0	0	1	1	1	1	1	31

14. What is the dotted decimal equivalent of the binary IP address
11001001 00110011 11001100 00111111?

- A. 201.59.204.63**
- B. 201.51.220.63**
- C. 223.51.204.63**
- D. 201.51.204.63**
- E. 201.51.204.127**

ANSWER: D. Use this chart to convert a binary IP address to its dotted decimal equivalent.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11001001	1	1	0	0	1	0	0	1	201
2 nd : 00110011	0	0	1	1	0	0	1	1	51
3 rd : 11001100	1	1	0	0	1	1	0	0	204
4 th : 00111111	0	0	1	1	1	1	1	1	63

15. What is the dotted decimal IP address represented by the binary number 11001111 00111110 11110010 00000011?

- A. 239.62.242.3**
- B. 207.62.250.3**
- C. 207.62.242.3**
- D. 207.63.242.3**

ANSWER: C. Use this chart to convert a binary IP address to its dotted decimal equivalent.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11001111	1	1	0	0	1	1	1	1	207
2 nd : 0111110	0	0	1	1	1	1	1	0	62
3 rd : 11110010	1	1	1	1	0	0	1	0	242
4 th : 00000011	0	0	0	0	0	0	1	1	3

16. What is the dotted decimal IP address represented by the binary number 11110011 00010011 11111110 01010101?
- A. 243.19.254.85**
 - B. 247.19.254.85**
 - C. 243.19.255.85**
 - D. 243.23.254.85**
 - E. 247.19.255.93**

ANSWER: A. Use this chart to convert a binary IP address to its dotted decimal equivalent.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11110011	1	1	1	1	0	0	1	1	243
2 nd : 00010011	0	0	0	1	0	0	1	1	19
3 rd : 11111110	1	1	1	1	1	1	1	0	254
4 th : 01010101	0	1	0	1	0	1	0	1	85

17. What is the dotted decimal IP address represented by the binary number 11000001 11010101 00100101 10001011?
- A. 193.229.45.139**
 - B. 193.229.45.143**
 - C. 194.229.37.147**
 - D. 193.229.37.139**
 - E. 194.229.37.139**

ANSWER: D. Use this chart to convert a binary number to its dotted decimal equivalent.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11000001	1	1	0	0	0	0	0	1	193
2 nd : 11010101	1	1	0	1	0	1	0	1	229
3 rd : 00100101	0	0	1	0	0	1	0	1	37
4 th : 10001011	1	0	0	0	1	0	1	1	139

18. What is the dotted decimal IP address represented by the binary number 11110010 11110001 00101010 01001010?

- A. 241.242.42.72**
- B. 242.241.42.72**
- C. 244.245.45.75**
- D. 242.234.46.62**
- E. 244.245.68.72**

ANSWER: B. Use this chart to convert a binary number to a dotted decimal value.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11110010	1	1	1	1	0	0	1	0	242
2 nd : 11110001	1	1	1	1	0	0	0	1	241
3 rd : 00101010	0	0	1	0	1	0	1	0	42
4 th : 01001010	0	1	0	0	1	0	1	0	72

19. What is the dotted decimal IP address represented by the binary number 11111110 00001010 11100101 01010110?

- A. 254.10.237.86**
- B. 254.10.228.87**
- C. 254.10.229.86**
- D. 254.12.237.86**
- E. 254.12.228.87**
- F. 254.12.229.86**

ANSWER: C. Use this chart to convert binary to dotted decimal.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11111110	1	1	1	1	1	1	1	0	254
2 nd : 00001010	0	0	0	0	1	0	1	0	10
3 rd : 11100101	1	1	1	0	0	1	0	1	229
4 th : 01010110	0	1	0	1	0	1	1	0	86

20. What is the dotted decimal address represented by the binary number 11001010 01010101 10100001 10000111?

- A. 206.85.161.135
- B. 202.85.161.135
- C. 202.85.169.135
- D. 202.85.161.143
- E. 206.85.161.143
- F. 206.92.161.143
- G. 206.92.169.135

ANSWER: B. Use this chart to convert binary to dotted decimal.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 11001010	1	1	0	0	1	0	1	0	202
2 nd : 01010101	0	1	0	1	0	1	0	1	85
3 rd : 10100001	1	0	1	0	0	0	0	1	161
4 th : 10000111	1	0	0	0	0	1	1	1	135

21. What is the dotted decimal address represented by the binary number 00010011 10001101 01010011 00100111?

- A. 19.141.83.39
- B. 23.141.83.39
- C. 19.149.83.39
- D. 19.149.84.47
- E. 19.149.91.47
- F. 27.149.91.47
- G. 27.141.83.53

ANSWER: A. Use this chart to convert binary to dotted decimal.

	128	64	32	16	8	4	2	1	Dotted Total For Octet
1 st : 00010011	0	0	0	1	0	0	1	1	19
2 nd : 10001101	1	0	0	0	1	1	0	1	141
3 rd : 01010011	0	1	0	1	0	0	1	1	83
4 th : 00100111	0	0	1	0	0	1	1	1	39

22. What subnet does IP address 199.37.22.38 /27 reside on?
Choose two.

- A. 199.37.22.0 255.255.255.224
- B. 199.37.22.16 255.255.255.240
- C. 199.37.22.32 255.255.255.224
- D. 199.37.22.64 255.255.255.240
- E. 199.37.22.0 /27
- F. 199.37.22.16 /27
- G. 199.37.22.32 /27
- H. 199.37.22.64 /27

ANSWER: C, G. To determine where the subnet where a particular IP address can be found, convert both the IP address and the subnet mask into binary. The subnet mask /27 refers to the binary makeup of the mask; the first 27 bits are ones, the remaining 5 bits are zeroes.

The Boolean AND operation is then performed. The Boolean AND is a bit-by-bit comparison of the IP address and subnet mask. If both numbers are "1", the AND result is "1". Any other combination results in a "0".

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 199.37.22.38	11000111	00100101	00010110	00100110
Subnet Mask	11111111	11111111	11111111	11100000
Boolean AND Result	11000111	00100101	00010110	00100000

Simply convert the Boolean AND result back to dotted decimal to yield the subnet the IP address will be found on.

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 11000111	1	1	0	0	0	1	1	1	199
2 nd Octet: 00100101	0	0	1	0	0	1	0	1	37
3 rd Octet: 00010110	0	0	0	1	0	1	1	0	22
4 th Octet: 00100000	0	0	1	0	0	0	0	0	32

The subnet is 199.37.22.32 /27, which is also expressed as 199.37.22.32 255.255.255.224.

In The REAL World...

Any type of binary math calculation is a little clumsy at first, but hang in there. You will end up miles ahead of those who memorized a chart. Trust me.

23. What subnet is the address 200.17.49.200 /23 a member of?

- A. 200.17.48.0 /23
- B. 200.17.49.0 /23
- C. 200.17.47.0 /23
- D. 200.17.48.128 /23
- E. 200.17.48.0 255.255.254.0
- F. 200.17.49.0 255.255.254.0
- G. 200.17.47.0 255.255.254.0
- H. 200.17.48.128 255.255.254.0

ANSWER: A, E.

As before, to determine the subnet upon which this IP address resides, convert both the IP address and the subnet mask into binary. The subnet mask /23 has the first 23 bits set at "1" and the remaining bits at "0". Then perform the Boolean AND. If the corresponding bits of the IP address and subnet mask are set to "1", the AND result is "1". Any other combination results in a "0".

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 200.17.48.200	11001000	00010001	00110001	11001000
Subnet Mask	11111111	11111111	11111110	11100000
Boolean AND Result	11001000	00010001	00110000	00000000

Converting the Boolean AND result back to dotted decimal yields the correct subnet number.

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 11001000	1	1	0	0	1	0	0	0	200
2 nd Octet: 00010001	0	0	0	1	0	0	0	1	17
3 rd Octet: 00110000	0	0	1	1	0	0	0	0	48
4 th Octet: 00000000	0	0	0	0	0	0	0	0	0

The subnet is 200.17.48.0 /23, which is also expressed as 200.17.48.0 255.255.254.0.

24. What subnet is the address 10.17.2.14 /18 a member of?

- A. **10.17.0.0 /18**
- B. **10.17.2.0 /18**
- C. **10.17.2.8 /18**
- D. **10.17.1.0 /18**

ANSWER: A.

Following the same procedure as the last two questions, we arrive at this chart:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 10.17.2.14	00001010	00010001	00000010	00001110
Subnet Mask	11111111	11111111	11110000	00000000
Boolean AND Result	00001010	00010001	00000000	00000000

The AND Result is now converted from binary to dotted decimal:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 00001010	0	0	0	0	1	0	1	0	10
2 nd Octet: 00010001	0	0	0	1	0	0	0	1	17
3 rd Octet: 00000010	0	0	0	0	0	0	0	0	0
4 th Octet: 00000000	0	0	0	0	0	0	0	0	0

The subnet is 10.17.0.0 /18, or 10.17.0.0 255.255.192.0.

25. What is the range of valid host addresses existing on the subnet 200.100.80.128 /25?

- A. 200.100.80.128 – 200.100.80.255
- B. 200.100.80.128 – 200.100.80.253
- C. 200.100.80.129 – 200.200.80.254
- D. 200.100.80.129 – 200.100.80.255
- E. 200.100.80.129 – 200.100.80.253

ANSWER: C. To resolve this type of question, simply write out the network number and subnet mask, and note the location and number of the host bits.

Network 200.100.80.128	11001000	10011010	10010110	10000000
Mask 255.255.255.128 (/25)	11111111	11111111	11111111	10000000
Host Bits				7 Host

If all the host bits are "0", the resulting address is 200.100.80.128. This is the network address.

If all the host bits are "1", the resulting address is 200.100.80.255. This is the broadcast address.

All addresses between the network address and broadcast address are valid host addresses. The range is 200.100.80.129 – 200.100.80.254.

Chris Bryant, CCIE #12933

165

www.thebryantadvantage.com

© 2004 The Bryant Advantage

26. What subnet is the address 178.45.12.200 /22 a member of?

- A. 178.45.12.0 /22
- B. 178.45.12.8 /22
- C. 178.45.20.0 /22
- D. 178.45.24.0 /22

Following the same procedure as the last two questions, we arrive at this chart:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 178.45.12.200	10110010	00101101	00001100	11001000
Subnet Mask	11111111	11111111	11111100	00000000
Boolean AND Result	10110010	00101101	00001100	00000000

The AND result is now converted from binary to dotted decimal:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 10110010	1	0	1	1	0	0	1	0	178
2 nd Octet: 00101101	0	0	1	0	1	1	0	1	45
3 rd Octet: 00001100	0	0	0	0	1	1	0	0	12
4 th Octet: 00000000	0	0	0	0	0	0	0	0	0

The subnet is 178.45.12.0 /22, or 178.45.12.0 255.255.252.0.

27. What subnet is the address 37.22.41.89 /29 a member of?
 Short answer question, no choices given.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 37.22.41.89	00100101	00010110	00101001	01011001
Subnet Mask	11111111	11111111	11111111	11111000
Boolean AND Result	00100101	00010110	00101001	01011000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 00100101	0	0	1	0	0	1	0	1	37
2 nd Octet: 00010110	0	0	0	1	0	1	1	0	22
3 rd Octet: 00101001	0	0	1	0	1	0	0	1	41
4 th Octet: 01011000	0	1	0	1	1	0	0	0	88

The subnet address is 37.22.41.88 /29, or 37.22.41.88
 255.255.255.252.

28. What subnet is the address 217.23.45.175 /25 a member of?
 Short answer, no choices given.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 217.23.45.175	11011001	00010111	00101101	10101111
Subnet Mask	11111111	11111111	11111111	10000000
Boolean AND Result	11011001	00010111	00101101	10000000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 11011001	1	1	0	1	1	0	0	1	217
2 nd Octet: 00010111	0	0	0	1	0	1	1	1	23
3 rd Octet: 00101101	0	0	1	0	1	1	0	1	45
4 th Octet: 10000000	1	0	0	0	0	0	0	0	128

The subnet address is 217.23.45.128 /25, or 217.23.45.128
255.255.255.128.

29. What is the range of valid host addresses existing on the subnet 10.1.1.16 255.255.255.240? No choices are given.

To resolve this type of question, simply write out the network number and subnet mask, and note how many host bits there are.

Network 10.1.1.16	00001010	00000001	00000001	0001 0000
Mask 255.255.255.240	11111111	11111111	11111111	11110000
Host Bits				4 Host Bits

If all the host bits are "0", the resulting address is 10.1.1.16. This is the network address.

If all the host bits are "1", the resulting address is 10.1.1.31. This is the broadcast address.

All addresses between the network address and broadcast address are valid host addresses. The range is 10.1.1.16 – 10.1.1.30.

30. You are planning a network's addressing scheme. You will be using network 172.16.0.0. You must allow for 150 subnets, and each subnet must allow at least 175 hosts. Which is the most appropriate subnet mask for you to use?
- A. 255.255.255.0
 - B. 255.255.255.128
 - C. 255.255.128.0
 - D. 255.255.255.1
 - E. 255.255.255.192

ANSWER: A. We begin this type of question by determining the default mask for this network. 172.16.0.0 is a Class B network, so we know the network mask is 255.255.0.0. This leaves us 16 bits for the subnet and host bits.

You can memorize a "powers of two" chart, but personally I have always found it easier to simply jot down "2 x 2" on a piece of paper and keep going until I get the required number of subnets or hosts. Don't forget that you must subtract two from the total, regardless of whether you are determining the number of host bits or the number of subnet bits.

For example, here we need 150 subnets. On a piece of paper, I would write out the following. (You can also do this before the exam begins if you choose.)

2 x 2	x 2	x 2	x 2	x 2	x 2	x 2	x 2	x 2	x 2
4	8	16	32	64	128	256	1024	2048	4196

Here, I can quickly see that for 30 subnets, I need to multiply two by itself five times, which gives me 32. As always, I subtract two from that, leaving 30 subnets.

Use whatever method makes you more comfortable. If you are going to use this method or write out a "powers of two" table, do so before beginning the exam.

Whichever method you use, we can see that for 150 subnets, we will need to multiply two by itself 8 times. This yields 256, and even after subtracting two, that leaves 254 subnets.

That would leave us with 8 host bits, which obviously gives us 254 host addresses, more than enough for the requirement to be met.

We now have a network mask of 255.255.0.0, 8 subnet bits, and 8 host bits. The subnet bits are "1" in binary, giving us a final subnet mask of 255.255.255.0.

	Octet 1	Octet 2	Octet 3	Octet 4
Class B Network Mask	11111111	11111111		
8 Subnet Bits			11111111	
8 Host Bits				00000000
Final Subnet Mask	11111111	11111111	11111111	00000000

31. You are planning a networking addressing scheme. You have been provided with the network number 235.17.18.0. You need at least 14 subnets. Your network manager has mandated that none of the subnets should ever contain more than 10 hosts. What is the most appropriate subnet mask for you to use?

- A. **255.255.255.128**
- B. **255.255.255.192**
- C. **255.255.192.0**
- D. **255.255.255.240**
- E. **255.255.255.248**

ANSWER: E. As before, we begin by determining the default network mask for this network number. The network provided is a Class C network, giving us a default mask of 255.255.255.0. This leaves us only the fourth octet to divide between the subnet bits and the host bits.

The first requirement is that we have 14 subnets. Using either a "powers of two" chart or the aforementioned method, we quickly determine that 4 subnet bits will yield 14 valid subnets. ($2 \times 2 \times 2 \times 2 = 16$; $16 - 2 = 14$.)

That would leave four bits for the host addresses. However, the second requirement did not specify a *minimum* number of host

addresses; it specified a *maximum*. If we have four host bits, that leaves us with 14 valid host addresses. (2 to the 4^{th} power is 16 ; $16 - 2 = 14$.) This violates the second requirement.

In order to meet the second requirement, we must have only three host bits. Two to the 3^{rd} power is 8 ; $8 - 2 = 6$. That would give us five subnet bits, which would yield 30 valid subnets. That's a lot more than 14, but it does fulfill the first requirement.

With five subnet bits and three host bits, we come up with a subnet mask of 255.255.255.248.

	Octet 1	Octet 2	Octet 3	Octet 4
Class C Network Mask	11111111	11111111	11111111	
5 Subnet Bits				11111
3 Host Bits				000
Final Subnet Mask	11111111	11111111	11111111	11111000

Watch the requirements on this kind of question. It would have been easy to just look at the first requirement and say, "Okay, four bits gives me 14 valid subnets, the mask is 255.255.255.240, next question." Read the requirements of any Cisco exam question **carefully**.

32. You are planning a network's addressing scheme. You will be using network 11.0.0.0. You must allow for 4000 subnets, but no more than 4200. Each subnet will have between 500 and 600 users. Which is the most appropriate subnet mask for you to use?
- A. 255.224.0.0**
 - B. 255.255.224.0**
 - C. 255.255.255.224**
 - D. None of the above.**

ANSWER: C. Again, we have an unusual set of requirements. We have only a minimum and maximum number of subnets, but a range of users. This range is an implicit requirement that we're going to need at least 600 valid host addresses on each subnet.

First, we determine the default mask. We're working with a Class A network, so the default mask is 255.0.0.0. We have 8 network bits and 24 bits to split between the subnet bits and the host bits.

By using the "powers of two" chart or the method I showed you earlier, we determine that for at least 4000 subnets, we will need 11 subnet bits. 2 multiplied by itself 11 times yields 4196; $4196 - 2 = 4194$. The first requirement stated that we have at least 4000 valid subnets, but no more than 4200. This number of subnet bits is the only number that meets both requirements.

2×2	$\times 2$								
4	8	16	32	64	128	256	1024	2048	4196

With eight network bits and eleven subnet bits, that leaves 13 host bits, which more than allows for the range of 500 to 600 valid host addresses. In binary, the final subnet mask is 255.255.224.0.

	Octet 1	Octet 2	Octet 3	Octet 4
Class A Network Mask	11111111			
11 Subnet Bits		11111111	111	
13 Host Bits			00000	000000000
Final Subnet Mask	11111111	11111111	11100000	000000000

33. Which of the following is not on the same subnet as IP address 148.14.23.238 /28?
- A. **148.14.23.225**
 - B. **148.14.23.230**
 - C. **148.14.23.235**
 - D. **148.14.23.240**

ANSWER: D. This question format is actually asking you two things. First, what subnet does that address/mask combination reside on? Secondly, in order to determine which of those addresses does not reside on that subnet, you've got to know how to determine the valid range of host addresses.

Since you've practiced binary math, you know how to perform both of these operations. First, we'll determine the subnet upon which the IP address 148.14.23.238 /28 resides.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 148.14.23.238	10010100	00001110	00010111	11101110
Subnet Mask /28	11111111	11111111	11111111	11110000
Boolean AND Result	10010100	00001110	00010111	11100000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 10010100	1	0	0	1	0	1	0	0	148
2 nd Octet: 00001110	0	0	0	0	1	1	1	0	14
3 rd Octet: 00010111	0	0	0	1	0	1	1	1	23
4 th Octet: 11100000	1	1	1	0	0	0	0	0	224

We now know that IP address 148.14.23.238 /28 resides on subnet 148.14.23.224 /28. Now we need to determine the valid host addresses for that subnet, like we did for earlier questions.

To do so, simply write out the network number and subnet mask, and note how many host bits there are.

Network 148.14.23.224	10010100	00001110	00010111	1110 0000
Mask 255.255.255.240 (/28)	11111111	11111111	11111111	11110000
Host Bits				4 Host Bits

There are four host bits. If each one is set to "0", the result is 148.14.23.224. This is the subnet number.

If each host bit is set to "1", the result is 148.14.23.239. This is the broadcast address for this subnet.

The addresses between these two are the valid host addresses. The range of valid host addresses is 148.14.23.225 – 148.14.23.238. Compare this range to the choices given and see which one does not belong.

34. Which of the following is not on the same subnet as IP address 10.1.1.200 /27?
- A. 10.1.1.193
 - B. 10.1.1.194
 - C. 10.1.1.195
 - D. 10.1.1.210
 - E. 10.1.1.220
 - F. 10.1.1.225

This question format is actually asking us two things. First, we must determine what subnet the given IP address is on. Second, we must determine the valid range of IP addresses for that subnet in order to decide which one of these five addresses does not fall in this range.

First, we'll determine the subnet upon which this IP address falls.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 10.1.1.200	00001010	00000001	00000001	11001000
Subnet Mask /27	11111111	11111111	11111111	11100000
Boolean AND Result	00001010	00000001	00000001	11000000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 00001010	0	0	0	0	1	0	1	0	10
2 nd Octet: 00000001	0	0	0	0	0	0	0	1	1
3 rd Octet: 00000001	0	0	0	0	0	0	0	1	1
4 th Octet: 11000000	1	1	0	0	0	0	0	0	192

We now know that the given IP address can be found on subnet 10.1.1.192 /27.

We now have to determine what the valid IP addresses are on this subnet. To do so, simply write out the network number and subnet mask, and note how many host bits there are.

Network 10.1.1.192	00001010	00000001	00000001	110 00000
Mask 255.255.255.224 (/27)	11111111	11111111	11111111	11100000
Host Bits				5 Host Bits

There are five host bits. If each is set to "0", the result is 10.1.1.192. This is the subnet address.

If each is set to "1", the result is 10.1.1.223. This is the broadcast address for the subnet. All addresses between these two are valid addresses, giving us a range of 10.1.1.193 – 10.1.1.222. Compare this range to the choices presented in the question.

35. What dotted decimal address is represented by the binary number 00111100 00101010 10101111 01110111? Short answer, no choices given.

To convert from binary to dotted decimal, use this chart.

	128	64	32	16	8	4	2	1	Decimal Total For Octet
1 st : 00111100	0	0	1	1	1	1	0	0	60
2 nd : 00101010	0	0	1	0	1	0	1	0	42
3 rd : 10101111	1	0	1	0	1	1	1	1	170
4 th : 01110111	0	1	1	1	0	1	1	1	119

The dotted decimal IP address is 60.42.170.119.

36. What subnet is the address 217.25.49.130 /26 a member of?
Short answer question, no choices given.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 217.25.49.130	11011001	00011001	00110001	10000010
Subnet Mask /26	11111111	11111111	11111111	11000000
Boolean AND Result	11011001	00011001	00110001	10000000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 11011001	1	1	0	1	1	0	0	1	217
2 nd Octet: 00011001	0	0	0	1	1	0	0	1	25
3 rd Octet: 00110001	0	0	1	1	0	0	0	1	49
4 th Octet: 10000000	1	0	0	0	0	0	0	0	128

The subnet this IP address can be found on is 217.25.49.128 /26.

37. What subnet is the address 111.12.116.201 /30 a member of?
 Short answer question, no choices given.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 111.12.116.201	01101111	00001100	01110010	11001001
Subnet Mask /30	11111111	11111111	11111111	11111100
Boolean AND Result	01101111	00001100	01110010	11001000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 01101111	0	1	1	0	1	1	1	1	111
2 nd Octet: 00001100	0	0	0	0	1	1	0	0	12
3 rd Octet: 01110010	0	1	1	1	0	0	1	0	116
4 th Octet: 11001000	1	1	0	0	1	0	0	0	200

The listed address resides on subnet 111.12.116.200 /30.

38. What is the range of valid host addresses existing on the subnet 110.1.1.192 /27?

- A. 110.1.1.192 – 110.1.1.224
- B. 110.1.1.192 – 110.1.1.223
- C. 110.1.1.192 – 110.1.1.224
- D. 110.1.1.193 – 110.1.1.222
- E. 110.1.1.194 – 110.1.1.222

ANSWER: D.

To resolve this type of question, simply write out the network number and subnet mask, and note how many host bits there are.

Network 110.1.1.192	01101110	00000001	00000001	11000000
Mask 255.255.255.224 (/27)	11111111	11111111	11111111	11100000
Host Bits				5 Host

If all the host bits are "0", the resulting address is 110.1.1.192. This is the network address.

If all the host bits are "1", the resulting address is 110.1.1.223. This is the broadcast address.

All addresses between the network address and broadcast address are valid host addresses. The range is 110.1.1.193 – 110.1.1.222.

39. Which of the following is not on the same subnet as IP address 222.48.209.82 /27? Choose all that apply.
- A. 222.48.209.60
 - B. 222.48.209.70
 - C. 222.48.209.80
 - D. 222.48.209.90
 - E. 222.48.209.100

ANSWER: A, E. Again, there are two tasks hidden in one question. First, we must determine what subnet the given IP address is on. Second, we must determine the valid range of IP addresses for that subnet in order to decide which one of these five addresses does not fall in this range.

First, we'll determine the subnet upon which this IP address falls.

When presented with this question format, express the IP address and mask in binary format and run the Boolean AND operation.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address 222.48.209.82	11011110	00110000	11010001	01010010
Subnet Mask /27	11111111	11111111	11111111	11100000
Boolean AND Result	11011110	00110000	11010001	01000000

Now convert the Boolean AND result to dotted decimal, and you have the subnet:

	128	64	32	16	8	4	2	1	Decimal Sum Of Octet
1 st Octet: 11011110	1	1	0	1	1	1	1	0	222
2 nd Octet: 00110000	0	0	1	1	0	0	0	0	48
3 rd Octet: 11010001	1	1	0	1	0	0	0	1	209
4 th Octet: 01000000	0	1	0	0	0	0	0	0	64

So far, so good. We know the subnet in question is 222.48.209.64 /27. Now we must determine the valid range of host addresses on that subnet. To do so, simply write out the network number and subnet mask, and note how many host bits there are.

Network 222.48.209.64	11011110	00110000	11010001	01000000
Mask 255.255.255.224 (/27)	11111111	11111111	11111111	11100000
Host Bits				5 Host Bits

There are five host bits. If each bit is set to "0", the result is 222.48.209.64. This is the subnet address. If each bit is set to "1", the result is 222.48.209.95. This is the broadcast address for this subnet.

All addresses in between these two are valid, giving us a range of 222.48.209.65 – 222.48.209.94. Compare this range to the choices given in the question to determine which one(s) are not on this subnet.

40. What is the binary equivalent of the dotted decimal IP address 145.89.205.18? Short answer, no choices are given.

As always, use this chart to convert dotted decimal to binary.

	128	64	32	16	8	4	2	1	Entire Octet In Binary
First Octet: 145	1	0	0	1	0	0	0	1	10010001
Second Octet: 89	0	1	0	1	1	0	0	1	01011001
Third Octet: 205	1	1	0	0	1	1	0	1	11001101
Fourth Octet: 18	0	0	0	1	0	0	1	0	00010010

The binary equivalent of 145.89.205.18 is 10010001 01011001
11001101 00010010.

Nice work!

Section Six: Initial Router Configuration

Terminology Used In This Chapter:

ROM: Read-Only Memory. ROM stores the router's bootstrap startup program, operating system software, and power-on diagnostic test programs.

Flash Memory: Generally referred to as "flash", the IOS images are held here. Flash is erasable and reprogrammable ROM. Flash memory content is retained by the router on power-down or reload.

RAM: Random-Access Memory. Stores operational information such as routing tables and the running configuration file. RAM contents are lost when the router is powered down or reloaded.

NVRAM: Non-volatile RAM. NVRAM holds the router's startup configuration file. NVRAM contents are not lost when the router is powered down or reloaded.

IOS: Internetwork Operating System. The router's operating system software.

The Router Boot Process

When a Cisco router powers up, it first runs a POST (Power-On Self Test). The POST is a series of diagnostic tests designed to verify the basic operation of the network interfaces, memory, and the CPU.

After the router passes the POST, it looks for a source from which to load a valid Internetwork Operating System (IOS). The router has three sources from which it can load an IOS image:

1. Flash memory (the default).
2. A TFTP server. (Trivial File Transfer Protocol)
3. Read-Only Memory (ROM)

For the router to look for the IOS from a TFTP server or from ROM, a change must be made to the *configuration register*, discussed in a later section.

Once the IOS is found, the router looks for a valid startup configuration file. By default, the router will look for the startup configuration file in Non-volatile RAM (NVRAM).

The router can be configured to load the startup configuration file from a TFTP server as well. If the startup file cannot be loaded from the TFTP server, the router will attempt to load it from NVRAM.

If no valid startup configuration file is found, the router enters *setup mode*, where the router run the *system configuration dialogue*, a series of questions involving basic router setup. This mode requires user input.

In The REAL World...

When configuring a router from scratch, most engineers prefer to do so from the Command Line Interface (CLI) than going through Setup mode.

The first question the router will ask is, “Would you like to enter the initial configuration dialog?” To avoid going through Setup mode and configure the router from the CLI, answer “N” to that question, and “Y” to the next question, “Would you like to terminate autoinstall?”

After you’ve configured a few routers in Setup mode and from the CLI, decide for yourself which method you prefer.

"User Exec" and "Enable" Modes

A prompt appears to press RETURN (the <ENTER> key) to begin, interface status messages appear, and the prompt looks like this:

Press RETURN to get started!

```
Cisco Internetwork Operating System Software
Copyright (c) 1986-2000 by cisco Systems, Inc.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to down
%LINK-5-CHANGED: Interface Ethernet0, changed state to administratively down
%LINK-5-CHANGED: Interface Serial0, changed state to administratively down
%LINK-5-CHANGED: Interface Serial1, changed state to administratively down
%LINK-5-CHANGED: Interface TokenRing0, changed state to administratively down
Router>
```

The router is in user EXEC mode.

This mode is **user exec mode**. The main use for this mode is to use it as a platform to log into **privileged exec mode**, the mode in which the startup and running configuration of the router can be changed.

```
Router>
Router>enable
Router#
```

The enable command takes the user from user exec to privileged exec mode. Privileged exec mode is indicated by the "#" following the router's hostname, which is currently "router".

To go back to user exec from privileged exec, use the **logout** command.

```
Router#logout
```

Router con0 is now available

```
Press RETURN to get started.
Router>
```

The user is back in user exec mode.

Router Modes

Global configuration mode is entered by typing **configure terminal** in privileged exec mode. (**config t** and **conf t** do the job.)

Once a command is set in global configuration mode, it is set **once** and affects **the entire router**.

Examining the configuration of a global command, beginning in user exec mode.

```
Router>enable  
Router#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#hostname R3  
R3(config)#^Z  
R3#
```

Looking at that configuration from the top line down:

*The **enable** command takes the router into privileged exec mode, where changes to the running configuration can be made.*

***conf t** is short for **configure terminal**, meaning that all commands that are entered after this are written to the running configuration, kept in RAM.*

*The global command **hostname** is run. Note that it takes effect immediately, as seen on the next line of the configuration.*

CTRL-Z appears as ^Z on the screen. This command takes the user back to privileged exec mode, as seen on the last line.

The **hostname** command is now written to the running configuration, which is kept in RAM; this will be lost if the router loses power before the current running configuration is saved.

To copy the running configuration over the current starting configuration, thereby retaining changes made since the last save, run **copy run start** in global configuration mode.

```
R3#copy run start  
Building configuration...  
[OK]
```

Interface Configuration Mode is entered from global configuration mode. Use the **interface** command followed by the interface to be configured:

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface ?
  Async      Async interface
  BVI        Bridge-Group Virtual Interface
  Dialer     Dialer interface
  Ethernet   IEEE 802.3
  Group-Async Async Group interface
  Lex         Lex interface
  Loopback   Loopback interface
  Null       Null interface
  Serial     Serial
  TokenRing  IEEE 802.5
  Tunnel     Tunnel interface
  Virtual-Template Virtual Template interface

R3(config)#interface serial0
R3(config-if)#

```

*Cisco IOS Help is accessed with the “?” to display all possible options of the **interface** command. CCNA candidates will be primarily concerned with the Dialer, Ethernet, Loopback, and Serial interfaces.*

At the bottom of the configuration, after entering “interface serial0” to configure that interface, the prompt changes to “config-if”, indicating the router is now in interface mode.

IOS HELP

IOS Help is called with the question mark symbol. By typing the “?” after a command and a space, the various options for that command are shown, as was illustrated in the previous example.

IOS Help can also be accessed by typing part of a command and entering a question mark without leaving a space between the two.

Examining both uses of IOS Help.

R3#**conf?**
Configure

*The question mark is entered directly after **conf**, which then lists all possible commands in this mode that begin with that string of letters.*

R3#**conf ?**

memory	Configure from NV memory
network	Configure from a TFTP network host
overwrite-network	Overwrite NV memory from TFTP network host
terminal	Configure from the terminal

When the question mark is preceded by a command AND a space, IOS Help will then list all available options for the command.

Configuring Router Passwords

The first two passwords to configure are the **enable secret** and **enable password**. If the names sound alike, that's because they have the same function. The user will be prompted to enter this password when entering privileged exec mode. The *enable password* is for older routers, also referred to as "legacy routers". The *enable secret* password will be used by the majority of the users.

If both passwords are in effect, the **enable password** will not be used.

*Examining the configuration and operation of the **enable secret** password.*

R3#**conf t**
R3(config)#**enable secret GETYOURCCNA**
R3(config)#^Z
R3#**logout**

*The **enable secret** password has been set. Users will be prompted for this password when attempting to enter privileged exec mode. To test this, the current user has been logged out with the **logout** command.*

R3 con0 is now available
Press RETURN to get started.

R3>en
Password:
R3#

C *The user was prompted for the enable secret password before being allowed into privileged exec mode. The password does not appear as it is being keyed in.*

W

A password can also be set for the console. Enter *line configuration mode* with the command **line console 0**, enter **login** to have the user prompted for a password when logging on to the console, and the **password** command is used to set the password.

Examining the logon process for a router with a console and enable secret password.

First, the console password is configured:

R3#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R3(config)#**line console 0**

R3(config-line)#**login**

R3(config-line)#**password CISCO**

R3(config-line)#^Z

R3(config)#logout

R3 con0 is now available

Press RETURN to get started.

User Access Verification

Password: < CISCO was entered here >

R3>en

Password: < GETYOURCCNA was entered here. >

R3#

The user is now prompted for the console password before user exec mode can be accessed. After entering that password, the user is prompted for the enable secret password to enter privileged exec mode.

Encrypting All Router Passwords In The Running Configuration

After configuring a console password and a telnet password, the passwords appear in the running configuration – *in clear-text*.

```
R3#show config
< output truncated for clarity >
!
line con 0
password GETYOURCCNA
login
line aux 0
line vty 0 4
password CISCO
login
!
end
```

By default, only the enable secret password will be encrypted in the running configuration. To encrypt all passwords in the running config, use the global command **service password-encryption**.

```
R3#conf t
R3(config)#service password-encryption

R3#show config
service password-encryption
!
line con 0
password 7 10692C2D3C3827392F27040A
login
line aux 0
line vty 0 4
password 7 14343B382F2B
login
!
end
```

The passwords are now encrypted and cannot be read from the running configuration.

Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) runs by default between all directly connected Cisco devices.

```
R2#show cdp neighbor
```

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

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
R1	BRI0	167	R	2521	Dialer1

Show cdp neighbor displays all directly connected Cisco routers and switches. CDP is Cisco-proprietary, so it will not display non-Cisco devices.

This command is particularly helpful when troubleshooting Cisco switches. There's no need to trace wiring in a rack of Cisco devices to see what routers are connected to a Cisco switch when **show cdp neighbor** can be used.

CDP can be disabled at both the global and interface level. To disable CDP at the interface level, run **no cdp enable** on the interface, and **cdp enable** to turn it back on.

By default, the **cdp timer** defines how often CDP packets are transmitted, and **cdp holdtime** defines how long a device will hold a received packet.

To turn CDP off for the entire router, run **no cdp run**. To view the current global status of CDP, run **show cdp**.

Displaying CDP information, and disabling CDP at the interface and global level.

R2#**show cdp**

Global CDP information:

Sending CDP packets every **60** seconds

Sending a holdtime value of **180** seconds

CDP is running by default.

R2#conf t

R2(config)#**cdp timer 45**

R2(config)#**cdp holdtime 100**

The CDP timers are changed.

R2#**show cdp**

Global CDP information:

Sending CDP packets every 45 seconds

Sending a holdtime value of 100 seconds

The CDP values have been successfully changed. “show cdp interface” will give the timer information for each interface on the router.

R2#conf t

R2(config)#interface bri0

R2(config-if)#no cdp enable

CDP is disabled on the BRI interface. This does NOT have to be done to keep the line from dialing, as will be shown.

R2#conf t

R2(config)#no cdp run

CDP is disabled globally.

R2#**show cdp**

% CDP is not enabled

CDP has been successfully disabled.

CDP Packets Do NOT Make An ISDN Link Dial Or Stay Up!

Occasionally, ISDN links that dial “mysteriously” or stay up after dialing are blamed on CDP packets making the line dial. This does NOT happen.

CDP has been reenabled on R2 at both the global and BRI interface level. R1 and R2 do not see themselves as directly connected, according to **show cdp neighbor**. This is because the line is down.

R2#show cdp

Global CDP information:

Sending CDP packets every 45 seconds

Sending a holdtime value of 100 seconds

R2#show cdp nei

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

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
< No CDP Neighbors are shown. >					

A **ping** is sent from R1 to R2, bringing the line up. (IP traffic is still defined as interesting.) R1 and R2 will now exchange CDP packets and **show cdp neighbors** displays the opposite router:

R1#**ping 172.12.21.2**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.12.21.2, timeout is 2 seconds:
.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 36/36/36

R1#show cdp neighbor

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
R2	Dialer1	84	R	2500	BRI0

No interesting traffic is sent over the line, but CDP packets are still going over the line. The CDP packets clearly do not keep the line up. When the CDP holddown timer expires, the routers no longer see each other as CDP neighbors.

R1#show dialer

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is idle

R1#show cdp neighbor

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
< When the CDP holdtime expires, the neighbor R2 is no longer seen. The ISDN line is down, and CDP neither kept the line up nor made it dial. >					

Physical Interfaces, Logical Interfaces, and IP Addressing

The Frame Relay section and labs gave you a taste of the different kinds of network interfaces that are available. Let's take a closer look at each.

Physical interfaces are just that. If you're putting an IP address on or configuring a "whole" interface, such as Ethernet0 or Serial1, you're configuring a physical interface.

There are two main types of logical interfaces, multipoint and point-to-point. Logical interfaces give you the ability to put a single physical interface into more than one network, or to allow for future network additions.

Let's say you have a single Serial interface going out to your Frame Relay cloud. You know that in the future, you may have to configure a point-to-point link on that interface, but for now, you need to be able to communicate with two other devices on that same link. You can configure a multipoint interface to the two devices you need to communicate with now, and you will be able to configure a point-to-point logical interface in the future.

Certain routing issues do arise when multipoint and point-to-point links are configured. You will configure both kinds of interfaces and be shown these issues in the labs for the routing protocol section.

Initial Router Configuration Q&A

1. Upon bootup, which of the following is the first task a Cisco router performs?

- A. The router looks for a valid IOS.
- B. The router looks for a valid startup configuration file.
- C. The router runs a power-on self-test.
- D. The router enters startup mode.

ANSWER: C. The power-on self-test, or POST, is a series of diagnostic tests designed to verify the basic operation of the memory, CPU, and network interfaces.

2. From what three sources will the router attempt to load the IOS?

- A. Flash memory.
- B. An FTP server.
- C. A TFTP server.
- D. RAM
- E. ROM
- F. The IOS server

ANSWER: A, C, E. The router will first attempt to load the IOS from Flash; it will then look to a TFTP server and Read-Only Memory if necessary.

3. By default, the router looks where for a valid startup configuration file?

- A. Flash memory.
- B. An FTP server.
- C. A TFTP server.
- D. RAM
- E. ROM
- F. NVRAM

ANSWER: F. NVRAM, or non-volatile RAM, is by default the first place the router looks for the startup configuration file.

4. You have configured the router to look for its valid configuration file from a TFTP server. For some reason, the router is unable to do so on startup. By default, what will happen next?

- A. The router will send a broadcast, looking for other TFTP servers.
- B. The router will load the valid configuration file from NVRAM.
- C. The router will enter setup mode.
- D. The router will continue to look for the TFTP server until the administrator enters CTRL-ALT-BREAK at the prompt.

ANSWER: B. If the router is configured to load its configuration file from a TFTP server, but it unable to do so, the router will then attempt to load the file from NVRAM.

5. You have configured the router to look for its valid configuration file from a TFTP server. For some reason, the router is unable to do so on startup. The router then looks to NVRAM, and cannot find a file there. What is the default behavior of the router in this situation?

- A. The router will enter setup mode.
- B. The router will enter safe mode.
- C. The router will look to RAM for a configuration file.
- D. The router will look to ROM for a configuration file.
- E. The router will auto-reload and go through the process again until a valid file is found from either the TFTP server or in NVRAM.

ANSWER: A. If a valid configuration file cannot be found from the named TFTP server, or in NVRAM, the router will enter setup mode.

6. You have just put a brand new Cisco router on the network. You start it up and go through setup mode. By default, what mode will the router be in after setup mode?

- A. user exec
- B. privilege exec
- C. interface
- D. console
- E. router

ANSWER: A. By default, the router will go into user exec mode.

7. You have just entered the command “hostname R4” in global mode. When will this command take effect?

- A. Immediately.
- B. Upon reboot.
- C. After the running configuration is written to memory.
- D. After the running configuration is written to the TFTP server.

ANSWER: A. Global commands are set only once and affect the entire router. The “hostname” command takes place immediately:

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R3
R3(config)#^Z
```

8. Looking at the router configuration, you notice that both an enable secret and enable password have been set. What is the effect when both are set?

- A. Neither will be applied.
- B. Both will be applied.
- C. The enable secret will not be applied.
- D. The enable password will not be applied.

ANSWER: D. When both the “enable secret” and “enable password” are set, the “enable secret” password will be used.

9. What mode does the command “enable secret” protect?

- A. user exec
- B. privilege exec
- C. console
- D. line
- E. router

ANSWER: B. Both “enable secret” and “enable password” are used to prompt the user for a password before the user can enter privileged exec mode.

10. You wish to use password protection for the router console. What command would be entered first to do so?

- A. **login**
- B. **login local**
- C. **password**
- D. **enable password**
- E. **line console 0**
- F. **line vty 0 4**

ANSWER: E. You would enter line console mode first, before entering a login type and password:

```
R3#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R3(config)#line console 0  
R3(config-line)#login  
R3(config-line)#password CISCO  
R3(config-line)#^Z
```

This configuration will allow any user to access the console who knows the password "CISCO".

11. You have created a password for console access. You run "show running-config" and notice that the password is in clear-text. What is your appropriate action?

- A. **Run the command "service password-encryption console" to encrypt the console password.**
- B. **Run the command "service password-encryption" to encrypt the passwords in the running configuration.**
- C. **Do nothing. The console access password, by default, cannot be encrypted.**
- D. **Do nothing. The console access password can be encrypted, but doing so violates Cisco best practices.**

ANSWER: B. The command "service password-encryption" will encrypt all passwords in the running configuration. It does not have the option to just encrypt the console password.

12. You have used the "enable secret" command to set a password for privileged exec mode. You run "show running-config" and notice that the password is in clear-text. What is your appropriate action?

- A. Run the command "encrypt secret" to encrypt this password.
- B. Run the command "service password-encryption" to encrypt the passwords in the running configuration.
- C. Do nothing. The enable secret password, by default, cannot be encrypted.
- D. This situation cannot happen.

ANSWER: D. By default, the enable secret password is encrypted and will be seen as such in the running configuration. Only the "enable password" would ever be seen in clear-text.

13. Consider the following console readout:

R2# < command removed >
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater
Device ID Local Intrfce Holdtme Capability Platform Port ID
R1 BRI0 167 R 2521 Dialer1

Which of the following three statements are true?

- A. The command used here was "show direct neighbors".
- B. The command used here was "show cdp neighbors".
- C. The command used here was "show igmp neighbors".
- D. The protocol in use is Cisco-proprietary.
- E. The protocol in use is not Cisco-proprietary.
- F. The protocol in use runs globally and on open interfaces by default.
- G. The protocol in use does not run globally or on open interfaces by default.

ANSWER: B, D, F. The command used here is "show cdp neighbors". It will show any directly connected Cisco device, along with

information such as the remote device's hostname and what remote interface is directly connected.

Notice that I said "directly connected Cisco device". CDP (Cisco Discovery Protocol) is Cisco-proprietary, meaning that if the two directly connected devices are not both Cisco devices, CDP will not work.

By default, CDP runs globally and on open interfaces by default.

14. You wish to turn Cisco Discovery Protocol off on your BRI interface. What interface-level command will you use to do so?

- A. **no cdp run**
- B. **no cdp enable**
- C. **cdp no run**
- D. **cdp disable**

ANSWER: B. The interface-level command is "no cdp enable". To prevent CDP from running globally on a router, run "no cdp run".

Consider the following router console readout:

R2# < command removed >

Global CDP information:

Sending CDP packets every 45 seconds
Sending a holdtime value of 100 seconds

15. Which two of the following statements are true?

- A. **The command run was "show cdp".**
- B. **The command run was "show cdp timers".**
- C. **The command run was "show cdp values".**
- D. **The default CDP value for sending CDP packets has been changed, but the holdtime is still at the default.**
- E. **The default CDP value for CDP holdtime has been changed, but the CDP send time is still at the default.**
- F. **The default CDP values have both been changed.**
- G. **The default CDP values are both at the default.**

ANSWER: A, D. "show cdp" shows the global settings for the CDP timers. To see the interface-specific timers, run "show cdp interface". CDP timers can be changed, and have been. The default values for

CDP is that CDP packets are sent every 60 seconds, and the default holdtime is 180 seconds.

16. You notice that anytime you misspell a router command, such as in the following screen, you get a strange message at the console:

R2#contin

Translating "contin"...domain server (255.255.255.255)

% Unknown command or computer name, or unable to find computer address

Every time this happens, the highlighted line appears for close to a minute, and you're tired of waiting like that any time you misspell a router command. What command will prevent this from happening?

- A. **no ip lookup**
- B. **no ip domain-lookup**
- C. **no ip translation**
- D. **no ip domain server**
- E. **no ip translation broadcast**

ANSWER: B. The default behavior of a Cisco router is to attempt to resolve this unknown command via Domain Name System (DNS). The "255.255.255.255." you see is the router broadcasting for a DNS server to resolve the entry "contin". This can take up to a minute before the router finally tells you it was unable to resolve the name.

"no ip domain-lookup" will change this default behavior. Should you need to use DNS lookups in the future, simply enter "ip domain-lookup" to reenable DNS lookups.

Section Seven: RIP, IGRP, And Static Routing

There's a lot going on in this section! We leave Layer Two for Layer Three, which is the OSI layer at which routing takes place. First, we'll look at static routing. You've got to know how to configure a static route and a static default route.

You'll then be introduced to two versions of RIP, then IGRP. During that introduction, you'll learn many important characteristics of distance vector protocols.

When you practice with these protocols, I urge you to begin practicing your debugs now. Debugs are not only important exam topics, but they show you what's actually happening when you enter your routing commands. That's the foundation you need to build now to help you with your future Cisco studies.

*Chris Bryant
CCIE #12933*

Commands Introduced In This Chapter And Labs Include:

debug ip packet – shows source and destination of packets as they enter and leave the router.

debug ip rip – shows RIP routes received and sent; also displays RIP version and authentication mode, if any.

ip route – Used to configure static routes.

key chain – Used to create key chains for RIP version 2 authentication.

network – Indicates what interfaces will be running a given protocol.

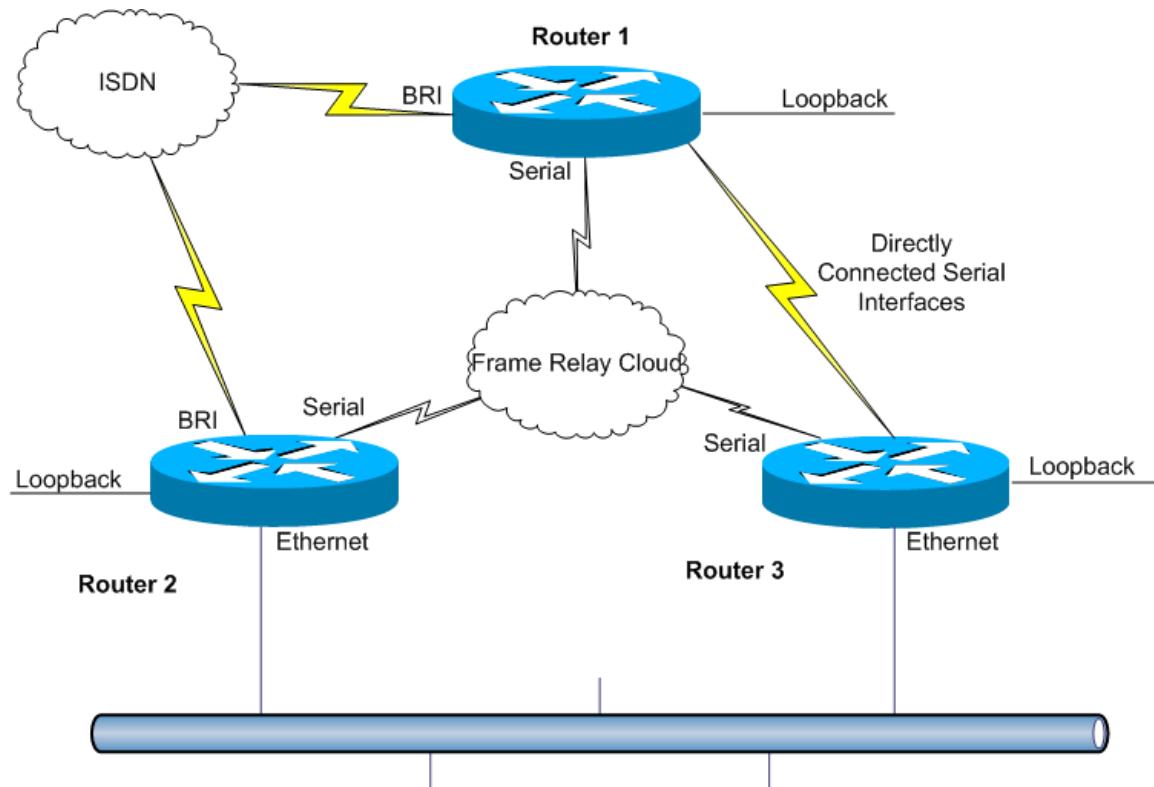
no auto-summary – Disables default auto-summarization in RIP version 2.

router rip – Enables RIP on the router.

show ip protocols – Displays protocols running on the router, route sources.

show ip route – Displays routing table.

The network used in this chapter:



Network Type	Network / Subnet Mask
Ethernet	172.23.23.0 /27
ISDN	172.12.21.0 /30
Serial to Frame Relay Cloud	172.12.123.0 /24
Directly Connected Serial Interfaces	172.12.13.0 /24
Router 1 Loopback Address	1.1.1.1 / 27
Router 2 Loopback Address	2.2.2.2 /27
Router 3 Loopback Address	3.3.3.3 / 27

Static Routes

A router uses a **routing table** to determine what interface a packet should leave that router on to reach a given destination. Routers run

protocols such as RIP and IGRP to exchange routing information and build their routing tables. Before any protocol is enabled, the table will show directly connected interfaces and the subnets they're connected to. Static routes inform the router to send packets destined for a remote network out a directly connected interface.

When a router is first connected to the network, **show ip route** will display the router interfaces and the subnets they are connected to. (Make sure the interfaces are open or the subnets will not appear in the routing table.)

```
R3#show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

172.12.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.12.13.0/24 is directly connected, Serial1
C 172.23.23.0/27 is directly connected, Ethernet0
C 172.12.123.0/24 is directly connected, Serial0.31
3.0.0.0/27 is subnetted, 1 subnets
C 3.3.3.0 is directly connected, Loopback0

The "C" next to the two routes indicate that these routes are directly connected, as does the description next to the subnets. The interface listed at the end of the route is the directly connected interface itself.

Adding A Static Route

A static route is added with the **ip route** command.

Each router has a **loopback address**. A loopback address is a logical interface with an IP address. Unlike physical interfaces, a loopback address can only go "down" if the router itself goes down.

A **ping** is used to test the ability of a router to send packets to a remote destination, *and* for those ping packets to return to the source. R3 now attempts to ping R2's loopback address. The ping fails. The

first step in finding out why a ping does not work is to check the local routing table with **show ip route**:

```
R3#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/4)
R3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR

Gateway of last resort is not set

      172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.12.13.0/24 is directly connected, Serial1
C        172.23.23.0/27 is directly connected, Ethernet0
C        172.12.123.0/24 is directly connected, Serial0.31
      3.0.0.0/27 is subnetted, 1 subnets
C        3.3.3.0 is directly connected, Loopback0
```

The router looks in its routing table for a match for network 2.0.0.0 and sees none. The router does not know how to send a packet to the destination; as a result, the ping fails.

The manual addition of a static route will allow R3 to send a ping to R2 via R1. To see if the packet leaves the frame successfully, **debug ip packet** is run, and the ping is sent again. After configuring the static route, the ping still fails. Why?

Examining Output Of “debug ip packet” and “ping 2.2.2.2” on R3:

```
R3#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R3(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.1
```

The Syntax Of The Static Route Command “ip route”:

2.2.2.2: Destination of the route.

255.255.255.255 A /32 subnet mask, indicating only packets exactly matching this destination should use this static route. A /24 mask (255.255.255.0) would mean that any packets for destination network 2.2.2.0 could use this route.

172.12.123.1: The next-hop address to use to reach the destination.

```
R3#debug ip packet  
IP packet debugging is on  
R3#ping 2.2.2.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

U.U.U

The ping code “U” indicates the destination is unreachable.

Success rate is 0 percent (0/5)

R3#

```
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending  
IP: s=172.12.123.1 (Serial0.31), d=172.12.123.3 (Serial0.31), len 56, rcvd 3  
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending  
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending  
IP: s=172.12.123.1 (Serial0.31), d=172.12.123.3 (Serial0.31), len 56, rcvd 3  
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending  
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending  
IP: s=172.12.123.1 (Serial0.31), d=172.12.123.3 (Serial0.31), len 56, rcvd 3
```

The output of “debug ip packet” indicates that the packets are indeed leaving the router out the desired interface. Packets are also being received from 172.12.123.1, the next-hop address over the Frame Relay network.

WARNING: Do not run “debug ip packet” unless it becomes absolutely necessary. The output of this command can overwhelm the router and force it to lock up.

With no routing protocol yet in place, a static route was added to R3 to allow the router to send packets to 2.2.2.2. The packets are leaving the router, but are not reaching the source. R1 is sending packets to R3 indicating the destination is unreachable.

Just as R3 did not know what interface could be used to reach 2.2.2.2 before the static route was configured, neither does R1. An additional static route must be added to R1 before the pings will go through, as

right now R1 receives the pings from R3 and has no route in its routing table for 2.2.2.2.

R1 has no route for 2.2.2.2 in its routing table, so it cannot route R3's pings.

R1#show ip route

```
172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C    172.12.13.0/24 is directly connected, Serial1
C    172.12.21.0/30 is directly connected, Dialer1
C    172.12.123.0/24 is directly connected, Serial0
```

A static route is added to R1 for destination 2.2.2.2, using 172.12.123.2 as its next hop.

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.2

R1 now has a static route in its routing table to 2.2.2.2. Static routes are indicated by “S”.

R1#show ip route

```
2.0.0.0/32 is subnetted, 1 subnets
S  2.2.2.2 [1/0] via 172.12.123.2
172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C    172.12.13.0/24 is directly connected, Serial1
C    172.12.21.0/30 is directly connected, Dialer1
C    172.12.123.0/24 is directly connected, Serial0
```

Pings from R3 to 2.2.2.2 are now successful. “debug ip packet” shows the pings entering and leaving the router.

R3#ping 2.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 136/138/148 ms

R3#

```
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending
IP: s=172.12.123.1 (Serial0.31), d=172.12.123.3 (Serial0.31), len 56, rcvd 3
IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending
IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending
IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending
IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3
IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending
IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3
```

Configuring A Static Default Route

A *default route* is the route a packet should take if it has no more specific entry in the routing table. Some protocols put default routes in the routing table by default, and a static default route can also be configured with the **ip route** command:

Examining The Static Default Route Command “ip route”

```
R1#conf t  
R1(config)#ip route 0.0.0.0 0.0.0.0 172.12.123.2
```

The route “0.0.0.0 0.0.0.0” matches all routes that have no more specific match in the routing table.

```
R1#show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 172.12.123.2 to network 0.0.0.0

The “gateway of last resort”, the destination to which all packets with no more-specific match in the routing table will be sent to , is now set to 172.12.123.2.

```
172.12.0.0/16 is variably subnetted, 4 subnets, 3 masks  
C    172.12.13.0/24 is directly connected, Serial1  
C    172.12.21.0/30 is directly connected, Dialer1  
C    172.12.21.2/32 is directly connected, Dialer1  
C    172.12.123.0/24 is directly connected, Serial0  
S*  0.0.0.0/0 [1/0] via 172.12.123.2
```

The manually configured default route appears in the routing table. Note the asterisk next to the “S” in the left-hand column, indicating that this route is a candidate to be the default route.

Distance Vector Protocols And Their Behavior

RIP and IGRP are **distance vector protocols**, so called because they use **hop count** to determine the best path to a destination.

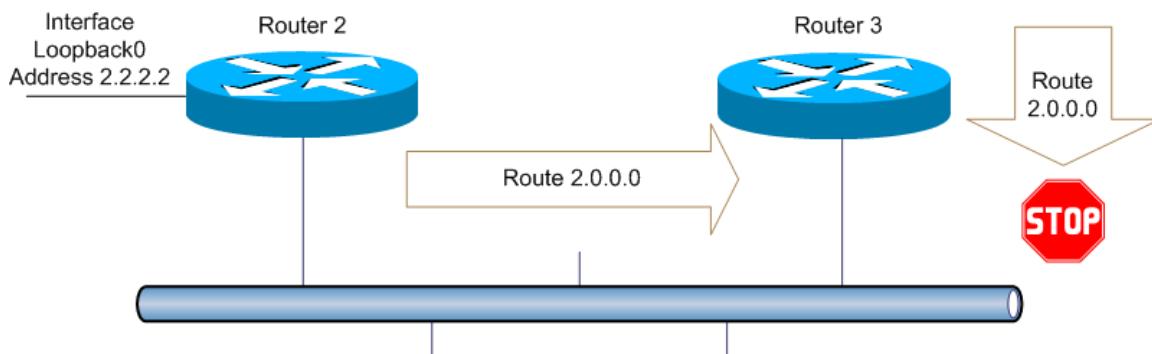
Before examining and configuring RIP and IGRP, it is important to understand the underlying methodology of distance vector protocols.

A **routing loop** occurs when an overall path to a destination results not in the packet reaching the destination, but instead the packet enters a "loop" where the packet is routed in an unending circle. Loops generally occur due to router misconfiguration or poor network design.

Distance Vector protocols use several different methods to prevent routing loops from occurring.

Split Horizon

Split Horizon is a simple yet powerful loop-avoidance feature. The rule of split horizon is that a route cannot be advertised out the same interface upon which the advertisement for that same network was originally received.



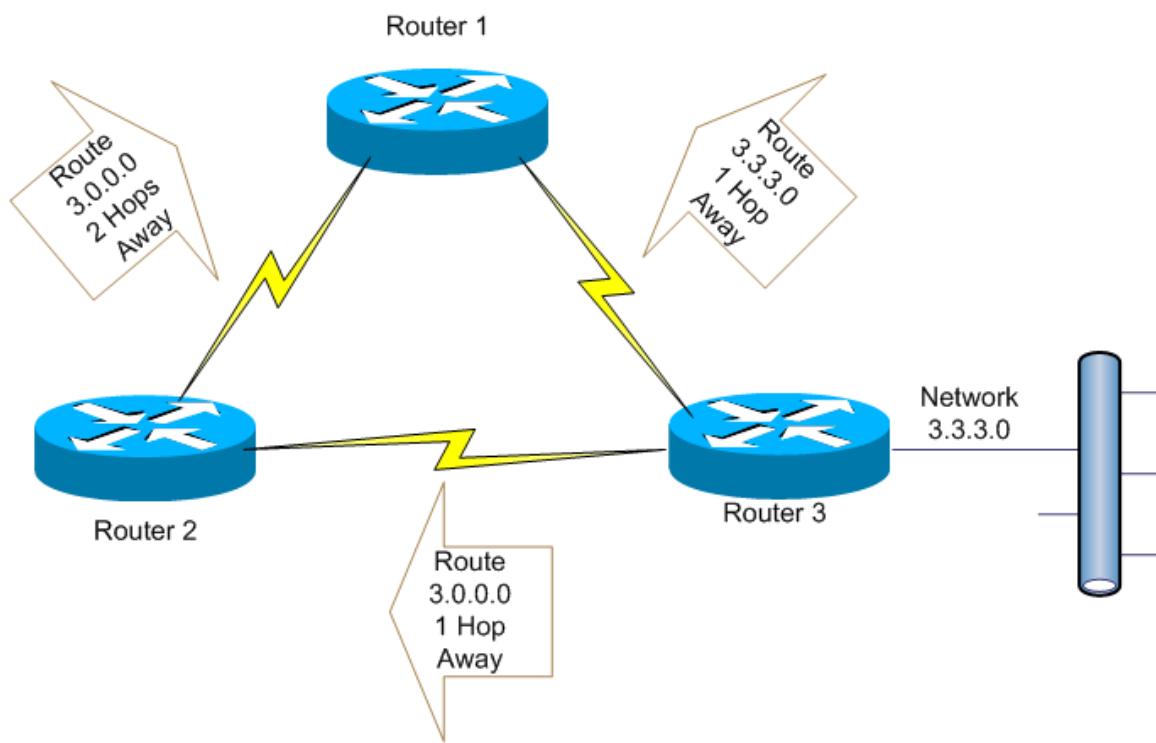
Router 2 and Router 3 are on the same Ethernet link. Router 2 is advertising its loopback address via a Distance Vector protocol. Router 3 receives the route, and wants to include it in routing updates send out on the Ethernet link. This would quickly cause a serious routing problem. Split Horizon proactively prevents the loop from

having a chance to form by preventing R3 from sending the 2.2.2.0 route back out the same physical interface on which it was received.

Route Poisoning

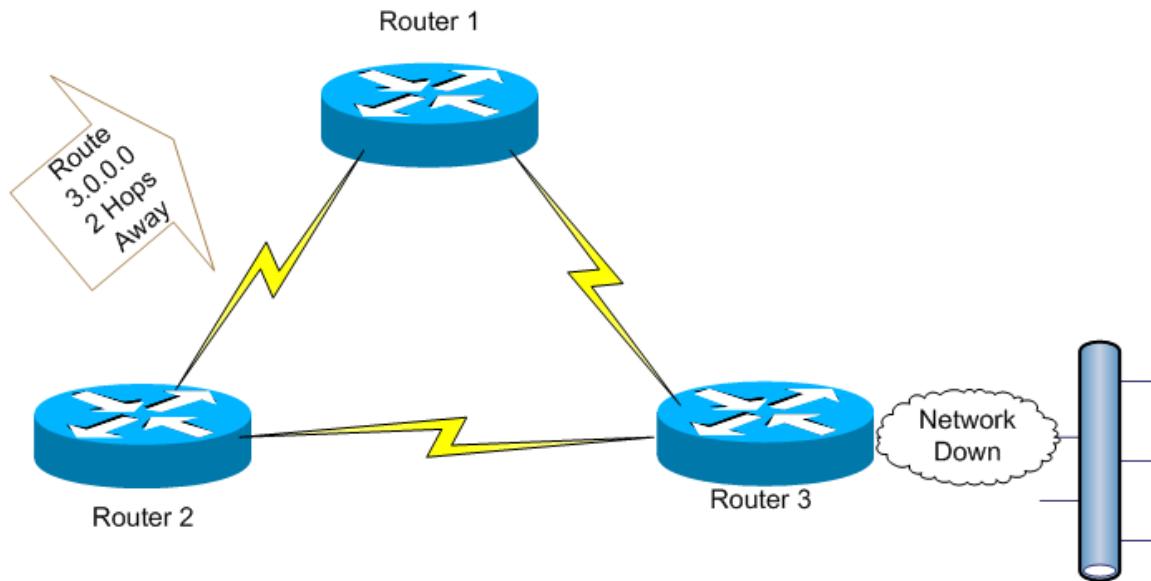
Route Poisoning occurs when a connected route becomes unavailable. If the router simply stopped sending advertisements for the connected routes, other routers in the network would continue to advertise the now invalid route.

With route poisoning, the router with the failed connected route continues to advertise the route, but with an invalid metric indicating the route is unreachable. Upon receipt of the advertisement containing the invalid metric, the downstream routers remove the network from their routing tables, and will no longer advertise that route.



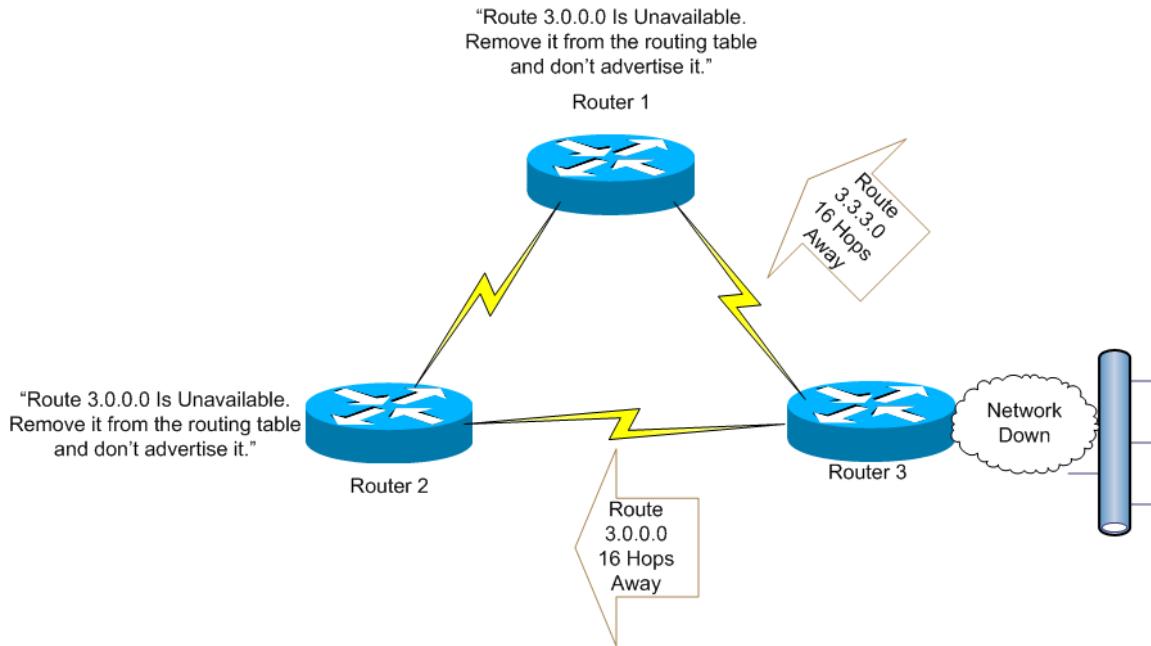
In this full-mesh network, R3 is advertising the network of its directly attached Ethernet interface to both R1 and R2. R1 is receiving updates for that network from both R2 and R3. Under normal circumstances with a distance vector protocol, the chances are great that the more direct path will be taken.

If the Ethernet network on R3 becomes unavailable, why doesn't R3 simply stop advertising the route, rather than advertising it with an unreachable metric? Consider what would happen if R2 sent a routing update to R1 while R3 simply stopped advertising the route.



Distance vector protocols do not converge immediately. (*Convergence* refers to all routers running a given protocol fully exchanging routes after a change in the network.) If R3 simply stops advertising network 3.0.0.0, R2 will continue to advertise it to R1. From R1's perspective, the more direct route will time out while it continues to get an advertisement for the route from R2. R1 may install the less-direct route into its routing table, resulting in continued transmission of data to an unavailable network. Neither R1 nor R2 has any way of knowing that 3.0.0.0 is down.

To prevent this, distance vector protocols use route poisoning. When network 3.0.0.0 becomes unavailable, R3 continues to advertise it, but with a metric indicating that it is unavailable. (In RIP, that metric is 16, shown here.) R1 and R2 will receive that route, see that the network is unavailable, and will no longer advertise it. The result is that R1 will not receive any route to the down network from either R2 or R3.



Split Horizon with Poison Reverse

Cisco routers utilize a combination of these two distance vector loop-avoidance procedures. ***Split Horizon with Poison Reverse*** operates exactly like basic split horizon until a route failure. Upon a route failure, the Cisco router begins the poison reverse procedure, advertising the route with an invalid metric. This advertisement containing the invalid metric is sent out **all** interfaces, including the interface the route the advertisement containing the invalid metric was received on.

Hold-Down Timers

Convergence occurs when the routers in a network all recognize that a change in the network has occurred, and have adjusted their routing tables according to the new route advertisements they have received.

Convergence is the goal after a change in the network topology, but if a route continually becomes available and unavailable, not staying in one state for more than a few seconds, the network can become flooded with routing updates and routing loops may occur.

A route or physical link is said to be "flapping" when the route becomes unavailable for a few seconds, then is available for few seconds, then unavailable again, and so on. If a distance-vector protocol were to remove the route from its table, then put it back in a

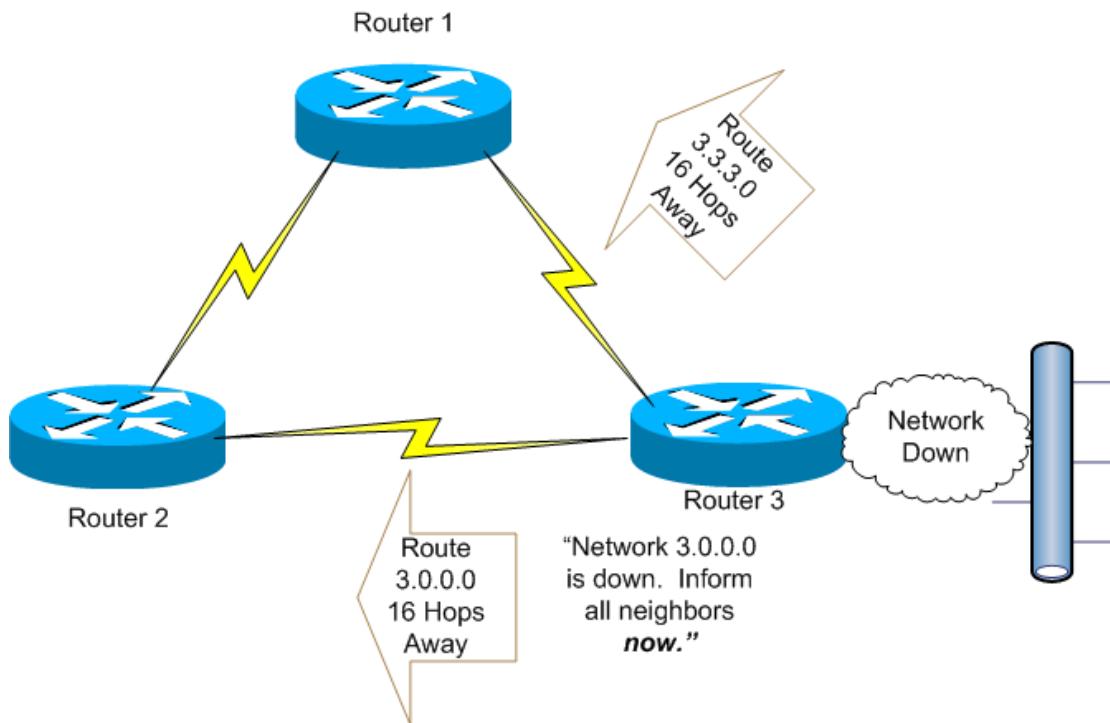
few seconds later, then remove it again, the sheer number of routing updates can cause a network slowdown. Worse, routing loops can form since the routers will be unable to converge due to the number of routing updates.

Distance vector protocols avoid this through the use of the hold-down timer. When a router running a distance vector protocol receives an advertisement that a route has failed, that router will not accept that route from any other router for the duration of the hold-down timer. This allows the routers to converge, avoiding the possibility of a routing loop.

Triggered Updates

The default behavior of a typical distance vector protocol is to send route advertisements at a given interval. With triggered updates, a new route advertisement is sent immediately upon a route failure, advising the neighbors of the failure almost immediately rather than waiting for the next scheduled update.

The network failure in the Route Poisoning section illustrates how triggered updates work. Instead of waiting until the next regularly scheduled update period to send an update regarding the downed network, R3 sends an immediate update to both R1 and R2.



Configuring RIP

The basic configuration of RIP, which runs on UDP port 520, is a simple one. In config mode, enter **router rip**, bringing you into router configuration mode. With the **network** command, enter the network number on the interface(s) that should participate in RIP routing. To view the protocol(s) running on this router, run **show ip protocols**.

Examining The Commands “router rip” and “show ip protocols”

```
R3#conf t
R3(config)#router rip      <Enables RIP on the router>
R3(config-router)#network 172.12.0.0
<All interfaces in network 172.12.0.0 will run RIP.>

R3#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 0 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface      Send   Recv   Key-chain
    Ethernet0      1      1 2 <Default Behavior : Send ver 1, receive ver. 1 and 2 >
    Serial0.31     1      1 2
    Serial1       1      1 2
  Routing for Networks:
    172.12.0.0
  Routing Information Sources:
    Gateway      Distance      Last Update
  <No information sources yet, as the other routers are not yet running a protocol. >
  Distance: (default is 120)
```

RIP runs in both Version 1 and Version 2. There are several major differences, the first being that RIP Version 1 does not recognize Variable-Length Subnet Masking (VLSM). As mentioned in the subnetting section, VLSMs allow the use of non-default, or **classless**, subnet masks in order to preserve IP addresses.

RIP version 1 does not recognize VLSMs; it only recognizes the default masks of Class A, B, and C networks. This is referred to as **classful routing**.

To configure a router to use RIP version 1 only, configure **version 1** in RIP configuration mode. To illustrate classful routing, R1 has been configured with a loopback address of 1.1.1.1 /27; R2 has a loopback address of 2.2.2.2/27; R3 has a loopback of 3.3.3.3/27. Each router is configured with RIP version 1, and each router is running RIP on its Frame Relay cloud interface , its loopback interface, and the Ethernet interfaces on R2 and R3.

Configuring RIP Version 1 On All Three Routers

```
R1#conf t
R1(config)#router rip
R1(config-router)#version 1
R1(config-router)#network 1.0.0.0
R1(config-router)#network 172.12.0.0
< The network command indicates all interfaces in that network will run RIP. >
R2#conf t
R2(config)#router rip
R2(config-router)#version 1
R2(config-router)#network 2.0.0.0
R2(config-router)#network 172.12.0.0

R3#conf t
R3(config)#router rip
R3(config-router)#version 1
R3(config-router)#network 3.0.0.0
R3(config-router)#network 172.12.0.0
```

The routers are all running RIP on their loopback interfaces, their Ethernet connections, and their Frame Relay interfaces. Since R1 and R2's BRI interface is also in the 172.12.0.0 network, those interfaces are also running RIP, which has ramifications to the network as well. R1 and R3 are also running RIP over their directly connected Serial interface.

Running **show ip protocols** displays what version of RIP is running, and on what interfaces.

Examining The Output of “show ip protocols” on R1 after configuring RIP version 1.

R1#**show ip protocols**

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 24 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

< These are the default values for RIP timers and updates. >

Outgoing update filter list for all interfaces is

Incoming update filter list for all interfaces is

< No filtering is taking place, so the filter list is blank. >

Redistributing: rip

Default version control: send version 1, receive version 1

< The default RIP behavior was changed with the “version 1” command. >

Interface	Send	Recv	Key-chain
Dialer1	1	1	
Loopback0	1	1	
Serial0	1	1	
Serial1	1	1	

< All four interfaces, including the logical interface Dialer1, are running RIP version 1.

The “Key-chain” column refers to RIP authentication, not supported by RIP version 1. >

Routing for Networks:

1.0.0.0
172.12.0.0

< The networks configured under RIP are 172.12.0.0 and 1.0.0.0. >

Routing Information Sources:

Gateway	Distance	Last Update
172.12.13.3	120	00:00:07
172.12.21.2	120	00:00:03
172.12.123.3	120	00:00:07
172.12.123.2	120	00:00:03

< R1 is receiving RIP updates from four IP addresses; 172.12.13.3 is R3’s directly connected Serial interface; 172.12.21.2 is R2’s BRI interface; 172.12.123.2 and 123.3 are the Frame Relay interfaces on R2 and R3. >

Distance: (default is 120)

< “Distance” refers to Administrative Distance, covered later in the chapter.

The RIP default Administrative Distance is 120. >

Examining The Output of “show ip route” on R1:

```
R1#show ip route
Gateway of last resort is not set
< A default route has not been set. >

  1.0.0.0/27 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
R  2.0.0.0/8 [120/1] via 172.12.21.2, 00:00:24, Dialer1
      [120/1] via 172.12.123.2, 00:00:24, Serial0
R  3.0.0.0/8 [120/1] via 172.12.123.3, 00:00:14, Serial0
      [120/1] via 172.12.13.3, 00:00:14, Serial1
R  172.23.0.0/16 [120/1] via 172.12.13.3, 00:00:28, Serial1
      [120/1] via 172.12.123.3, 00:00:28, Serial0
      [120/1] via 172.12.123.2, 00:00:25, Serial0
      [120/1] via 172.12.21.2, 00:00:25, Dialer1
< R1 has received two routes for 2.0.0.0 /8 and for 3.0.0.0 /8, and four different routes for
  172.23.0.0 /16, the Ethernet network connecting R2 and R3. >
```

The numbers contained in the brackets following the network number are the **Administrative Distance** and the **Hop Count**. The Administrative Distance is a measure of a protocol's "believability". If a router receives a route via two different protocols, the route advertised by the protocol with the **lowest** Administrative Distance is the route placed in the routing table. Protocols use **metrics** to measure the desirability of one route against another, and RIP uses Hop Count as its sole metric. RIP does not care what the speed of a link may be; RIP sees all links as being the same, and only cares about how many hops it takes to reach a destination.

Administrative Distance does not play a role in this configuration, since only one protocol is running. The hop counts of the different routes to the same destinations are equal; such routes are called "equal-cost" routes. RIP's default behavior is to allow up to four equal-cost paths to be added to the routing table for the same destination, and **load-balancing** will occur across the links. (Up to six equal-cost paths can be utilized for load balancing with RIP; to change the default value of four, use the **maximum-paths** configuration command under the RIP process.) This is true for both versions of RIP.

Since RIP version 1 is in use, the subnet masks of /27 configured on the loopback interfaces are ignored. RIP version 1 is a **classful** protocol; it will only support subnet masks of /8, /16, and /24 for Class A, B, and C networks, respectively. That is why the routes to the loopback interfaces on R1 appear with /8 masks.

By configuring the same network with RIP version 2, the same subnets are advertised, but with their correct /27 subnet masks. While routing in a small network such as this will work correctly with RIP version 1, larger networks that must conserve networks need the VLSM support of RIP version 2. RIP Version 2's default behavior of summarizing routes across classful network boundaries is removed with the command **no auto-summary**.

Configuring RIP Version 2

```
R1#conf t
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 1.0.0.0
R1(config-router)#network 172.12.0.0

R2#conf t
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#network 2.0.0.0
R2(config-router)#network 172.12.0.0
R2(config-router)#network 172.23.0.0

R3#conf t
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)#network 3.0.0.0
R3(config-router)#network 172.12.0.0
R3(config-router)#network 172.23.0.0
```

R1's routing table after configuring RIP Version 2:

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
  1.0.0.0/27 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
  2.0.0.0/27 is subnetted, 1 subnets
R    2.2.2.0 [120/1] via 172.12.21.2, 00:00:22, Dialer1
      [120/1] via 172.12.123.2, 00:00:21, Serial0
  3.0.0.0/27 is subnetted, 1 subnets
R    3.3.3.0 [120/1] via 172.12.123.3, 00:00:24, Serial0
      [120/1] via 172.12.13.3, 00:00:24, Serial1
  172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C    172.12.13.0/24 is directly connected, Serial1
C    172.12.21.0/30 is directly connected, Dialer1
C    172.12.123.0/24 is directly connected, Serial0
  172.23.0.0/27 is subnetted, 1 subnets
R    172.23.23.0 [120/1] via 172.12.21.2, 00:00:22, Dialer1
      [120/1] via 172.12.123.2, 00:00:22, Serial0
      [120/1] via 172.12.123.3, 00:00:25, Serial0
      [120/1] via 172.12.13.3, 00:00:25, Serial1
```

R1 has the same routes, but with the much more accurate subnet masks. This gives the routers a more accurate picture of the network, particularly important in larger networks.

The load balancing behavior of RIP version 2 is the same as version 1; up to four equal-cost paths can load-balance, with a maximum of six possible with the **maximum-paths** command.

Critical Differences Between RIP Version 1 and RIP Version 2:
RIP version 2 supports authentication, RIP version 1 does not.

Authentication is the process of a router running a process to verify the identity of the remote router or device attempting to contact it.

RIP version 1 has no authentication mechanism. RIP version 2 allows for two kinds of authentication. **Clear-text** authentication sends the password over the link with no encryption; if the authentication packet is intercepted during transmission, the password is easily acquired by the intruder. **Message Digest 5 (MD5)** authentication performs a hash on the password before transmitting. Even if the password is intercepted during transmission, the hash is of no use to the intruder.

Configuring RIP Version 2 Text Authentication Between R1, R2, and R3

```
R1#conf t
R1(config)#key chain RIP
< The key chain can have any name. >
R1(config-keychain)#key 1
< Key chains can have multiple keys. Number them carefully when using multiples. >
R1(config-keychain-key)#key-string CISCO
< This is the text string the key will use for authentication. >
R1(config)#int s0
R1(config-if)#ip rip authentication mode text
< The interface will use clear-text mode. >
R1(config-if)#ip rip authentication key-chain RIP
< The interface is using key chain RIP, configured earlier. >

R2#conf t
R2(config)#key chain RIP
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string CISCO
R2(config)#int s0.123
R2(config-subif)#ip rip authentication mode text
R2(config-subif)#ip rip authentication key-chain RIP

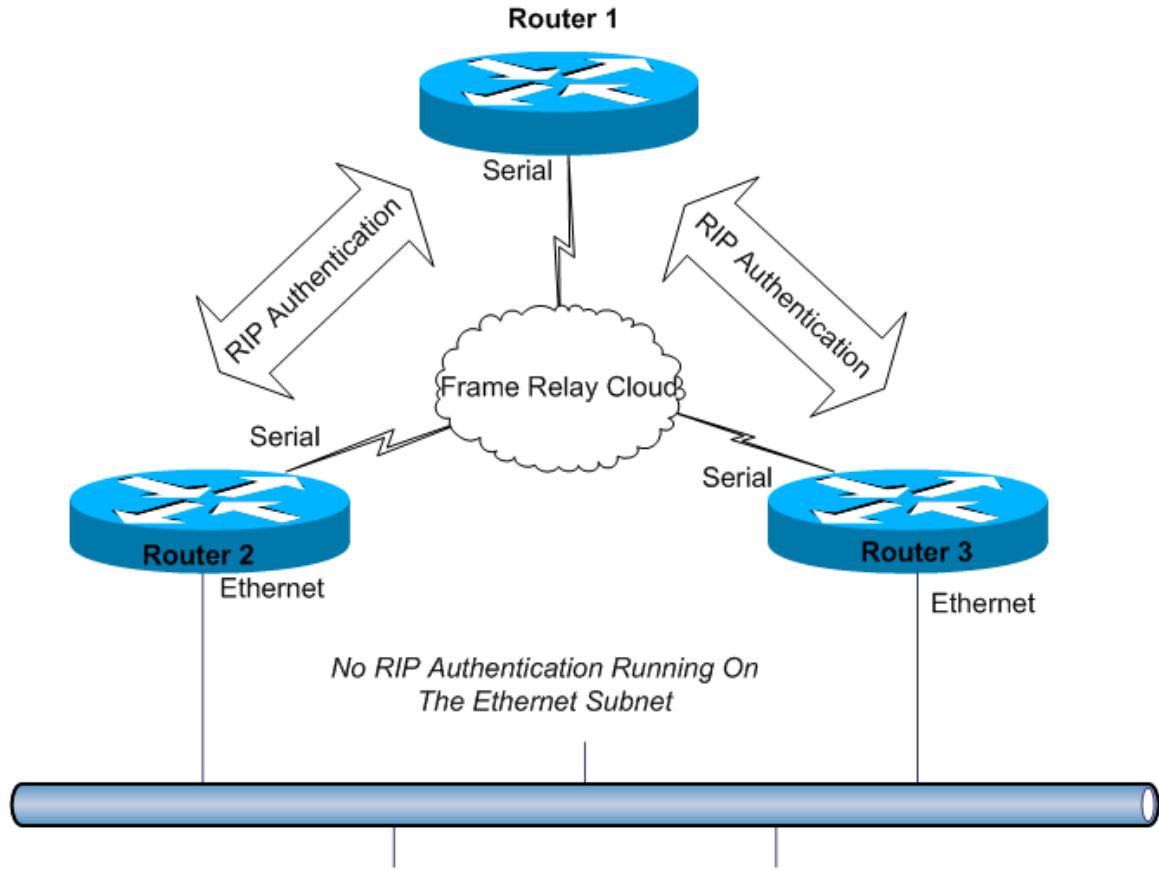
R3#conf t
R3(config)#key chain RIP
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CISCO
R3(config)#int s0.31
R3(config-subif)#ip rip authentication mode text
R3(config-subif)#ip rip authentication key-chain RIP
```

Show ip protocols displays the name of the key chain RIP, and that authentication is in use on that interface.

Examining the output of “show ip protocols” after configuring text authentication.

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
  Interface      Send  Recv  Key-chain
  Dialer1        2    2
  Loopback0      2    2
  Serial0        2    2          RIP
  Serial1        2    2
Routing for Networks:
  1.0.0.0
  172.12.0.0
Routing Information Sources:
  Gateway      Distance   Last Update
  172.12.13.3      120   00:00:07
  172.12.21.2      120   00:00:02
  172.12.123.3     120   00:00:07
  172.12.123.2     120   00:00:02
Distance: (default is 120)
```

RIP authentication can be run over a single link in a network running RIP without every router in the network having to do so. In this network, the router interfaces on the Frame Relay cloud are authenticating, but the two routers on the Ethernet segment are not.



To authenticate with MD5 encryption, replace the word "text" in the prior configuration with "md5". All other configurations are the same.

Configuring MD5 authentication between R2 and R3's Ethernet interfaces.

```
R2#conf t
R2(config)#key chain ENCRYPTED
< The key chain is called ENCRYPTED. It can have any name. >
R2(config-keychain)#key 1
< A key chain can have multiple keys. Even if using only one, it must be numbered. >
R2(config-keychain-key)#key-string CISCO
< CISCO is the password that will be hashed before transmission. >
R2(config-keychain-key)#interface e0
R2(config-if)#ip rip authentication mode md5
< MD5 authentication is specified. >
R2(config-if)#ip rip authentication key-chain ENCRYPTED
< The key chain name is used here. Again, note the hyphen in this command. >

R3#conf t
R3(config)#key chain ENCRYPTED
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CISCO
R3(config-keychain-key)#interface ethernet0
R3(config-if)#ip rip authentication mode md5
R3(config-if)#ip rip authentication key-chain ENCRYPTED
```

```
R3#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 1 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface      Send  Recv  Key-chain
    Ethernet0      2      2      ENCRYPTED
    Loopback0      2      2
    Serial0.31     2      2      RIP
    Serial1        2      2
  Routing for Networks:
    3.0.0.0
    172.12.0.0
    172.23.0.0
```

Troubleshooting RIP: Authentication Issues

A common error with RIP authentication occurs during configuration. If the engineer configuring the password hits the space bar before hitting <ENTER> to enter the data string, **a null space will be added to the password**. This null space is not visible by looking at the running configuration, but if one router has a null space after the key-string value and the other does not, authentication will not occur.

Another common error is to configure authentication on one side but not the other, or to configure text authentication on one side and md5 on the other. Even if both routers are configured with the same password, authentication will not occur if the two routers are using different encryption schemes.

Debug ip rip is a vital RIP troubleshooting command. This command displays routes being exchanged, RIP versions being used, and encryption types in use. Running this command on R1 and then clearing the routing table with **clear ip route *** yields this partial output:

```
R1#debug ip rip
RIP protocol debugging is on
R1#cle ip route *
02:42:44: RIP: sending general request on Dialer1 to 224.0.0.9
<RIP Version 2 multicasts to 224.0.0.9 for initial RIP neighbor information. >
02:42:44: RIP: sending general request on Serial0 to 224.0.0.9
02:42:44: RIP: sending general request on Serial1 to 224.0.0.9
02:42:44: RIP: received v2 update from 172.12.13.3 on Serial1
02:42:44:    172.12.13.0/24 -> 0.0.0.0 in 1 hops
02:42:44:    172.12.123.0/24 -> 0.0.0.0 in 1 hops
02:42:44:    2.2.2.0/27 -> 0.0.0.0 in 2 hops
02:42:44:    3.3.3.0/27 -> 0.0.0.0 in 1 hops
02:42:44:    172.23.23.0/27 -> 0.0.0.0 in 1 hops
02:42:51: RIP: received packet with text authentication CISCO
<The update from Serial 1 used clear-text password CISCO. >
02:42:51: RIP: received v2 update from 172.12.123.2 on Serial0
02:42:51:    172.12.21.0/30 -> 0.0.0.0 in 1 hops
02:42:51:    2.2.2.0/27 -> 0.0.0.0 in 1 hops
02:42:51:    3.3.3.0/27 -> 0.0.0.0 in 2 hops
02:42:51:    172.23.23.0/27 -> 0.0.0.0 in 1 hops
02:42:56: RIP: received packet with text authentication CISCO
<The update from Serial0 used clear-text password CISCO. >
```

That output shows the routes, subnet masks, hop counts, and working authentication. It will also show when there is a problem with authentication.

Assume R1 is still using text authentication for both RIP neighbor relationships over the Frame Relay cloud, but R2 has mistakenly been configured with MD5 authentication. This would become obvious when shortly after configuring authentication, R1 would be learning routes from R3 but not R2:

Using “show ip protocols” and “show ip route rip” to troubleshoot.

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 18 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface      Send  Recv  Key-chain
    Dialer1        2     2
    Loopback0      2     2
    Serial0        2     2          RIP
    Serial1        2     2
Routing for Networks:
  1.0.0.0
  172.12.0.0
Routing Information Sources:
  Gateway      Distance      Last Update
  172.12.13.3    120    00:00:02
  172.12.21.2    120    00:00:13
  172.12.123.3   120    00:00:02
  172.12.123.2   120    00:02:38
Distance: (default is 120)
```

```
R1#show ip route rip
2.0.0.0/27 is subnetted, 1 subnets
R  2.2.2.0 [120/1] via 172.12.21.2, 00:00:10, Dialer1
  3.0.0.0/27 is subnetted, 1 subnets
  R  3.3.3.0 [120/1] via 172.12.13.3, 00:00:24, Serial1
    [120/1] via 172.12.123.3, 00:00:24, Serial0
  172.23.0.0/27 is subnetted, 1 subnets
  R  172.23.23.0 [120/1] via 172.12.21.2, 00:00:10, Dialer1
    [120/1] via 172.12.13.3, 00:00:25, Serial1
    [120/1] via 172.12.123.3, 00:00:24, Serial0
```

After running **show ip protocols** and **show ip route rip**, two changes are evident. The “Routing Information Sources” section of **show ip protocols** indicates the last routing update that originated from 172.12.123.2 occurred over two minutes ago. The second line of that output indicates that the default time of 30 seconds between updates is still being used, so there is some sort of problem with receiving updates from 172.12.123.2.

Show ip route rip, which displays only routes learned via RIP, now shows only a single route for 2.2.2.0/27, where there were two. There are also only three routes to the Ethernet segment 172.23.23.0, where there were four. The routes that are no longer present were being sent by 172.12.123.2, another indicator of a problem with receiving routes over the Frame Relay cloud link to R2.

When troubleshooting a router configuration, the first question to ask is, **“What was the most recent change to this device?”** The answer here is that RIP authentication was added. To view possible authentication problems with RIP, run **debug ip rip** and clear the routing table with **clear ip route ***.

WARNING:

Do not practice debugs on production networks, as certain debugs can cause a router to overload and then lock up. Do not use debugs unless you are certain of the output you will get. Do not clear production network routing tables with “clear ip route *”.

Examining partial output of “debug ip rip” to detect authentication problems.

```
03:20:08: RIP: received v2 update from 172.12.21.2 on Dialer1
03:20:08: 172.12.123.0/24 -> 0.0.0.0 in 1 hops
03:20:08: 2.2.2.0/27 -> 0.0.0.0 in 1 hops
03:20:08: 3.3.3.0/27 -> 0.0.0.0 in 2 hops
03:20:08: 172.23.23.0/27 -> 0.0.0.0 in 1 hops
03:20:08: RIP: ignored v2 packet from 172.12.123.2 (invalid authentication)
```

The message "invalid authentication" indicates that either R2 is running a mismatched authentication type, or no authentication at all. **Show ip protocols** on R2 shows that no key chain is configured on interface Serial 0.123. The command **show key chain** is also used to display the different key chains that have been configured on R2. Double-checking key chains before configuring authentication can prevent these problems from occurring in the first place.

Resolving the RIP authentication issue on R2.

R2#**show ip protocols**

Routing Protocol is "rip"

 Sending updates every 30 seconds, next due in 18 seconds

 Invalid after 180 seconds, hold down 180, flushed after 240

 Outgoing update filter list for all interfaces is not set

 Incoming update filter list for all interfaces is not set

 Redistributing: rip

 Default version control: send version 2, receive version 2

Interface	Send	Recv	Key-chain
-----------	------	------	-----------

BRI0	2	2	
------	---	---	--

Ethernet0	2	2	ENCRYPTED
-----------	---	---	-----------

Loopback0	2	2	
-----------	---	---	--

Serial0.123	2	2	<No key chain is configured on Serial 0.123 >
-------------	---	---	---

R2#**show key chain**

Key-chain RIP:

 key 1 -- text "CISCO"

 accept lifetime (always valid) - (always valid) [valid now]

 send lifetime (always valid) - (always valid) [valid now]

< The text-authentication key chain exists, but needs to be configured on the interface. >

Key-chain ENCRYPTED:

 key 1 -- text "CISCO"

 accept lifetime (always valid) - (always valid) [valid now]

 send lifetime (always valid) - (always valid) [valid now]

R2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#int s0.123

R2(config-subif)#**ip rip authentication mode text**

R2(config-subif)#**ip rip authentication key-chain RIP**

< The authentication type and key chain are applied to the interface. >

Show ip route rip displays that R2's Serial0.123 interface is again sending routes to R1, and **debug ip rip** shows the routes coming in with text authentication from that interface. To turn off all working debugs, run **undebbug all**.

```
R1#show ip route rip
 2.0.0.0/27 is subnetted, 1 subnets
R    2.2.2.0 [120/1] via 172.12.21.2, 00:00:21, Dialer1
      [120/1] via 172.12.123.2, 00:00:21, Serial0
  3.0.0.0/27 is subnetted, 1 subnets
R    3.3.3.0 [120/1] via 172.12.13.3, 00:00:07, Serial1
      [120/1] via 172.12.123.3, 00:00:07, Serial0
  172.23.0.0/27 is subnetted, 1 subnets
R    172.23.23.0 [120/1] via 172.12.21.2, 00:00:21, Dialer1
      [120/1] via 172.12.13.3, 00:00:07, Serial1
      [120/1] via 172.12.123.3, 00:00:07, Serial0
      [120/1] via 172.12.123.2, 00:00:21, Serial0

R1#debug ip rip
RIP protocol debugging is on
R1#cle ip route *
03:49:56: RIP: received packet with text authentication CISCO
03:49:56: RIP: received v2 update from 172.12.123.2 on Serial0
03:49:56:   172.12.21.0/30 -> 0.0.0.0 in 1 hops
03:49:56:   2.2.2.0/27 -> 0.0.0.0 in 1 hops
03:49:56:   3.3.3.0/27 -> 0.0.0.0 in 2 hops
03:49:56:   172.23.23.0/27 -> 0.0.0.0 in 1 hops
R1#undebbug all
All possible debugging has been turned off
```

Critical Differences Between RIP Version 1 and RIP Version 2: How The Two Versions Send Updates

The final major difference between versions is how they send updates. RIP version 1 uses broadcasts to send updates, where RIP version 2 sends multicasts to address 224.0.0.9. This difference is seen when a router first starts running RIP and sends requests for information out each interface running RIP.

By running **debug ip rip** and then configuring RIP version 1 on a router, the broadcasts (destination of 255.255.255.255) can be seen be sent out the RIP-enabled interfaces. RIP version 1 also uses

broadcasts to send routing updates. RIP Version 2 uses multicasts to 224.0.0.9 to perform these tasks.

RIP Version 1 sends broadcasts to discover neighbors and send updates.

```
R1#debug ip rip
RIP protocol debugging is on
R1#conf t
R1(config)#router rip
R1(config-router)#version 1
R1(config-router)#network 172.12.0.0
04:02:59: RIP: sending general request on Dialer1 to 255.255.255.255
04:02:59: RIP: sending general request on Serial0 to 255.255.255.255
04:02:59: RIP: sending general request on Serial1 to 255.255.255.255
04:05:58: RIP: sending v1 update to 255.255.255.255 via Serial0 (172.12.123.1)
04:05:58:    subnet 172.12.13.0, metric 1
04:05:58:    subnet 172.12.123.0, metric 1
04:05:58: RIP: sending v1 update to 255.255.255.255 via Serial1 (172.12.13.1)
04:05:58:    subnet 172.12.123.0, metric 1
```

RIP Version 2 uses multicast address 224.0.0.9 to discover neighbors and send updates.

```
R1#debug ip rip
RIP protocol debugging is on
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 172.12.0.0
04:09:45: RIP: sending general request on Dialer1 to 224.0.0.9
04:09:45: RIP: sending general request on Serial0 to 224.0.0.9
04:09:45: RIP: sending general request on Serial1 to 224.0.0.9
04:09:55: RIP: sending v2 update to 224.0.0.9 via Dialer1 (172.12.21.1)
04:09:55:    172.12.13.0/24 -> 0.0.0.0, metric 1, tag 0
04:09:55:    172.12.123.0/24 -> 0.0.0.0, metric 1, tag 0
04:09:55: RIP: sending v2 update to 224.0.0.9 via Serial0 (172.12.123.1)
```

Troubleshooting RIP: Keeping The ISDN Line Down

In the ISDN section, mention is made of keeping the ISDN line down when not in use. Most organizations use their ISDN links to for backup purposes, in case the main link (generally Frame Relay) goes down. The reason for this is cost; an ISDN line is much like a phone call in that the billing for the link depends on how often the link is actually dialed and connected to the remote end.

It is vital to know what traffic is capable of bringing the ISDN line up. In the lab situations examined so far, any IP traffic has been capable of bringing the ISDN line up. Note that RIP has been running between R1 and R2 over both the Frame Relay link and the ISDN link. A look at the output of **show dialer** on R1:

Partial output of “show dialer” on R1 reveals that the line has been up for almost 20 minutes.

R1#**show dialer**

```
BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Interface bound to profile Dialer1
Time until disconnect 114 secs
Current call connected 00:19:26
```

The output indicates the call will disconnect in 114 seconds, but remember that any interesting traffic that crosses the link – in this case, any IP traffic – will reset that timer. If interesting traffic crosses the link for the next week, that line is going to stay up for a week, resulting in a huge phone bill. What traffic is keeping the link up? Two consecutive **show dialer** commands, run about 15 seconds apart, reveals what traffic is bringing the link up, and why the line has remained up for so long.

“show dialer” on R2 reveals that RIP version 2 updates, multicast to 224.0.0.9, are keeping the link up. Since the updates are sent every 30 seconds by default, the idle-timer will continue to reset, and the ISDN link will stay up indefinitely.

R2#show dialer

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=172.12.21.2, d=224.0.0.9)
< The RIP Version 2 update sent to 224.0.0.9 is the destination of the packet that brought the line up – and kept it up. The source is R2’s Serial interface. >
Time until disconnect 93 secs
< This value is reset to 120 seconds by default, but will be reset by the RIP Version 2 updates, which will go out over this interface every 30 seconds by default. >
Connected to 8358661 (R1)

R2#show dialer

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=172.12.21.2, d=224.0.0.9)
Time until disconnect 107 secs
< The idle-timer has been reset. Since this value is larger than the idle-timeout setting was a few seconds ago, interesting traffic has again reset the timer. >
Connected to 8358661 (R1)

Cisco provides a method by which an interface will not send updates, but will still receive them. By configuring the ISDN interfaces as **passive-interfaces**, these interfaces will not send updates out those interfaces, which would allow the line to go down in the absence of interesting traffic.

Configuring the ISDN interfaces as passive, preventing them from sending RIP updates.

R1#conf t

R1(config)#**router rip**

< Configuring interfaces as passive is done in router configuration mode. >

R1(config-router)#**passive-interface dialer0**

< R1 is using Dialer Profiles, so Dialer0 is the passive interface. >

R2#conf t

R2(config)#**router rip**

R2(config-router)#**passive-interface bri0**

R2#show dialer

BRI0 - dialer type = ISDN

Dial String Successes Failures Last called Last status

8358661 3 1 00:52:37 successful

0 incoming call(s) have been screened.

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is idle

< The RIP updates are no longer sent out the BRI interface. The lack of interesting traffic has allowed the ISDN link to come down. >

BRI0:2 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is idle

Configuring IGRP

When configuring IGRP, the initial configuration must include an autonomous system number, or AS number. This number is used as a link between routers running the same IGRP process; for example, any routers running IGRP AS 1 are said to be in the same autonomous system.

Configuring IGRP Autonomous System 1 on R1, R2, and R3.

```
R1#conf t
R1(config)#router igrp 1
< IGRP AS 1 is configured. >
R1(config-router)#passive-interface dialer1
< IGRP broadcast updates will not be sent out Dialer1, keeping the ISDN line down. >
R1(config-router)#network 172.12.0.0
R1(config-router)#network 1.0.0.0

R2#conf t
R2(config)#router igrp 1
R2(config-router)#passive-interface bri0
< IGRP broadcast updates will not be sent out BRI0, keeping the ISDN line down. >
R2(config-router)#network 172.12.0.0
R2(config-router)#network 2.0.0.0
R2(config-router)#network 172.23.0.0

R3#conf t
R3(config)#router igrp 1
R3(config-router)#network 172.12.0.0
R3(config-router)#network 3.0.0.0
R3(config-router)#network 172.23.0.0
```

R1's routing table shows several routes learned via IGRP. As with RIP, if multiple equal-cost routes are discovered, up to four can be placed into the routing table by default; to change that default to allow from one to six equal-cost routes, use the **maximum-paths** command under the IGRP process.

RI's routing table, containing several IGRP routes ("I").

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

1.0.0.0/27 is subnetted, 1 subnets
C 1.1.1.0 is directly connected, Loopback0
I 2.0.0.0/8 [100/8976] via 172.12.123.2, 00:00:01, Serial0
I 3.0.0.0/8 [100/8976] via 172.12.13.3, 00:00:59, Serial1
[100/8976] via 172.12.123.3, 00:00:59, Serial0
172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 172.12.13.0/24 is directly connected, Serial1
C 172.12.21.0/30 is directly connected, Dialer1
C 172.12.123.0/24 is directly connected, Serial0
I 172.23.0.0/16 [100/8576] via 172.12.123.2, 00:00:01, Serial0
[100/8576] via 172.12.13.3, 00:00:59, Serial1
[100/8576] via 172.12.123.3, 00:00:59, Serial0

Equal-cost multiple routes have been placed in the routing table, but the Administrative Distance and IGRP metrics are different. The AD is the first number in the brackets following the network number in the route entry, and for IGRP that number is 100. The number following that is the IGRP metric, which for these routes is 8976.

How IGRP Computes The Metric

IGRP considers hop count, but unlike RIP, it is not the sole determining factor. RIP sees all links as being the same; the bandwidth is not considered, only the number of hops to a destination. IGRP uses a composite metric that considers hop count, but can also consider *bandwidth, delay, load, and reliability*.

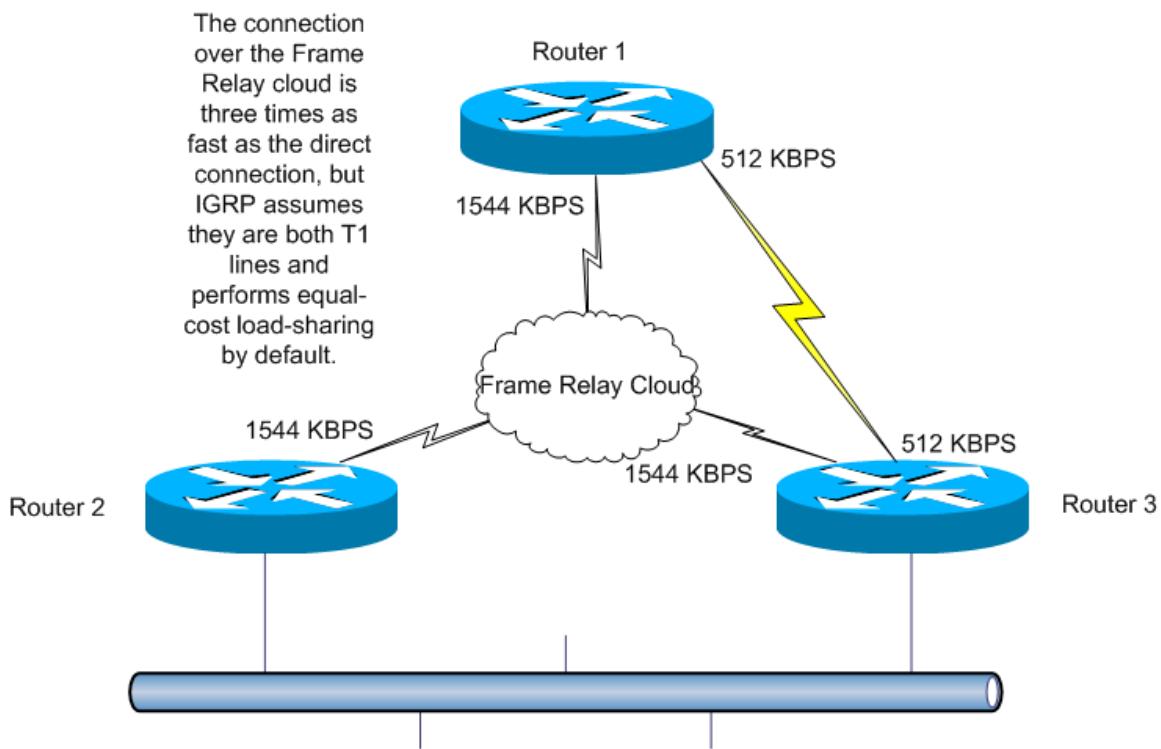
Of those four, IGRP only considers bandwidth and delay by default. Both of these values can be changed at the interface level. Generally, when it becomes necessary to alter the IGRP metric, bandwidth is the

value that is changed. By raising the bandwidth value, the metric value lowers, making that route more desirable.

Tuning IGRP's Default Bandwidth Assumptions

IGRP assumes all Serial interfaces are connected to a T1 line, which runs at 1544 kbps. If a serial line is actually configured to a slower line, IGRP has no way to detect this. The bandwidth value should be changed at the interface level to reflect the correct speed, allowing IGRP to recalculate the routes and give the routers in the IGRP AS a truer picture of the network.

Assume the direct link between R1 and R3 is only a 512 KBPS line, and the Frame Relay connection is a true T1 line. By default, IGRP calculates its metrics by considering both of these connections to be a T1 line running at 1544 kbps, which results in equal-cost load balancing for traffic leaving R1 destined for 172.23.23.0.



"show ip route 172.23.0.0" on R1 shows three equal-cost routes, and all three lines are assumed by IGRP to be running at 1544 KBPS, even though the R1-R3 direct connection is actually running at 512 KBPS.

R1#**show ip route 172.23.0.0**

Routing entry for 172.23.0.0/16

Known via "igrp 1", distance 100, metric 8576

Redistributing via igrp 1

Advertised by igrp 1 (self originated)

Last update from 172.12.123.2 on Serial0, 00:00:17 ago

Routing Descriptor Blocks:

* 172.12.123.2, from 172.12.123.2, 00:00:17 ago, via Serial0

Route metric is 8576, traffic share count is 1

 Total delay is 21000 microseconds, **minimum bandwidth is 1544 Kbit**

 Reliability 255/255, minimum MTU 1500 bytes

 Loading 1/255, Hops 0

172.12.13.3, from 172.12.13.3, 00:01:16 ago, via Serial1

Route metric is 8576, traffic share count is 1

 Total delay is 21000 microseconds, **minimum bandwidth is 1544 Kbit**

 Reliability 255/255, minimum MTU 1500 bytes

 Loading 1/255, Hops 0

172.12.123.3, from 172.12.123.3, 00:01:16 ago, via Serial0

Route metric is 8576, traffic share count is 1

 Total delay is 21000 microseconds, **minimum bandwidth is 1544 Kbit**

 Reliability 255/255, minimum MTU 1500 bytes

 Loading 1/255, Hops 0

Load balancing is occurring on the equal-cost routes, but since one of the lines is not truly running at 1544 KBPS, this load balancing may not be desired.

The interface-level command **bandwidth 512** on both ends of the R1 – R3 direct connection allows IGRP to recalculate the metric based on the actual bandwidth of the interfaces, rather than the assumed value. After configuring the true bandwidth, **show ip route 172.23.0.0** shows load balancing is only occurring over the true T1 lines.

The interface-level “bandwidth” command changes the default IGRP assumption that all serial interfaces run at 1544 kbps, preventing load balancing over slower links.

```
R1#conf t  
R1(config)#int serial1  
R1(config-if)#bandwidth 512  
< The bandwidth is adjusted at the interface level. >
```

```
R3#conf t  
R3(config)#int serial1  
R3(config-if)#bandwidth 512
```

```
R1#show ip route 172.23.0.0  
Routing entry for 172.23.0.0/16  
Known via "igrp 1", distance 100, metric 8576  
Redistributing via igrp 1  
Advertised by igrp 1 (self originated)  
Last update from 172.12.123.3 on Serial0, 00:00:07 ago  
Routing Descriptor Blocks:  
* 172.12.123.2, from 172.12.123.2, 00:00:12 ago, via Serial0  
    Route metric is 8576, traffic share count is 1  
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit  
    Reliability 255/255, minimum MTU 1500 bytes  
    Loading 1/255, Hops 0  
172.12.123.3, from 172.12.123.3, 00:00:08 ago, via Serial0  
    Route metric is 8576, traffic share count is 1  
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit  
    Reliability 255/255, minimum MTU 1500 bytes  
    Loading 1/255, Hops 0
```

```
R1#show ip route igrp  
I 2.0.0.0/8 [100/8976] via 172.12.123.2, 00:00:53, Serial0  
I 3.0.0.0/8 [100/8976] via 172.12.123.3, 00:00:50, Serial0  
I 172.23.0.0/16 [100/8576] via 172.12.123.2, 00:00:53, Serial0  
                           [100/8576] via 172.12.123.3, 00:00:50, Serial0
```

As a result of the IGRP route recalculation that occurred after changing the bandwidth values, the metric of the direct connection between R1 and R3 increased. Since it is no longer an equal-cost best route, it has been removed from the routing table. Load balancing will now only occur over the true T1 lines.

IGRP Unequal-Cost Load Balancing

As mentioned, IGRP uses a composite metric. If load balancing is desired, it can be quite difficult to get this metric to be exactly the

same. The **variance** command is used to allow a route with a metric higher than the path(s) in use to be used for load balancing. After the serial line upgrade, the metric of the directly connected serial interface link is still higher than the links running over the Frame Relay cloud. The **variance** command can be used to bring the R1-R3 link into use and utilize it for load balancing, but there are some prerequisites for using this command.

The route must be a **feasible route**. To be a feasible route, two conditions must be met:

1. The next-hop router must have a metric to the destination that is lower than the local router's metric for that destination. From R1, R3 is the next-hop router. R3 is directly attached to 172.23.0.0, so it definitely has a lower metric to the network than R1. The first condition is met.
2. When multiplied by the variance, the metric of the lowest-cost route must be greater than the metric of the route to be added.

The **variance** command is a multiplier; when the value supplied with the variance command is multiplied by the lowest-cost metric, it must exceed the higher-cost metric in order for the higher-cost route to be added.

The lowest-cost metric for network 172.23.0.0 on R1 is 8576. The **variance** command must be supplied with a value that when multiplied by 8576, is greater than the metric of the higher-cost route traversing the R1-R3 direct route.

IGRP does not have a "show" command that displays all possible routes to a destination, as does EIGRP. Using the variance command with IGRP requires a bit of experimentation. The bandwidth of a T1 line (the Frame Relay cloud links) is roughly three times that of a 512 Kbps line, so **variance 3** is a good place to start. (The command **debug ip igrp transactions** can show the current metric of the routes coming off the 512 Kbps route. This command will be explored in the lab section.)

Using the “variance” command to allow load balancing over the direct Serial link.

```
R1(config)#router igrp 1  
R1(config-router)#variance 3
```

```
R1#show ip route 172.23.0.0  
Routing entry for 172.23.0.0/16  
Known via "igrp 1", distance 100, metric 8576  
Redistributing via igrp 1  
Advertised by igrp 1 (self originated)  
Last update from 172.12.123.2 on Serial0, 00:00:01 ago  
Routing Descriptor Blocks:  
* 172.12.13.3, from 172.12.13.3, 00:00:20 ago, via Serial1  
  Route metric is 21631, traffic share count is 1  
    Total delay is 21000 microseconds, minimum bandwidth is 512 Kbit  
    Reliability 255/255, minimum MTU 1500 bytes  
    Loading 1/255, Hops 0  
  172.12.123.3, from 172.12.123.3, 00:00:20 ago, via Serial0  
    Route metric is 8576, traffic share count is 3  
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit  
    Reliability 255/255, minimum MTU 1500 bytes  
    Loading 1/255, Hops 0  
  172.12.123.2, from 172.12.123.2, 00:00:01 ago, via Serial0  
    Route metric is 8576, traffic share count is 3  
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit  
    Reliability 255/255, minimum MTU 1500 bytes  
    Loading 1/255, Hops 0
```

The estimate of **variance 3** was correct, as the metric for 172.23.0.0 through the direct connection is 21631. A variance of 3 means that any route with a metric less than the best metric multiplied by the variance (in this case, $8576 \times 3 = 25728$) will be entered into the routing table. R1 now has three unequal-cost paths to 172.23.0.0 in its routing table, and load balancing will take place.

How IGRP Unequal-Cost Load Balancing Takes Place

The default behavior of IGRP unequal-cost load balancing is for the better routes to carry proportionally more traffic. If two routes are performing unequal-cost load balancing, and one has half the metric of another, the router with the lower metric will carry twice the data of the other.

IGRP allows two changes to this default. Even though the routes have unequal metrics, they can be configured to carry approximately the

same amount of traffic with the **traffic-share balanced** router configuration command.

Splitting the traffic load in half, disregarding proportional differences in bandwidth.

```
R1#conf t  
R1(config)#router igrp 1  
R1(config-router)#traffic-share balanced
```

This command does not change the IGRP metrics in any way. It simply instructs the IGRP process to balance the load over any links performing load balancing.

IGRP can also be configured to use only the lowest-cost route. This would seem to defeat the purpose of load balancing in the first place, but does give the advantage of almost instantaneously transmitting data over the second-best route in case the best route goes down, thus avoiding convergence delays. The lesser routes will remain in the routing table, but transmit no data.

Using only the lowest-cost route for data transmission.

```
R1#conf t  
R1(config)#router igrp 1  
R1(config-router)#traffic-share minimum
```

Debugging IGRP

To view a router broadcasting requests for information, and to view where routing updates are sent and received, run **debug ip igrp events**.

Partial output of “debug ip igrp events”.

```
R1#debug ip igrp events  
IGRP event debugging is on  
19:14:42: IGRP: broadcasting request on Loopback0  
19:14:42: IGRP: broadcasting request on Serial0  
19:14:42: IGRP: broadcasting request on Serial1  
19:14:42: IGRP: received update from 172.12.13.3 on Serial1
```

Debug ip igrp events displays the initial broadcasts for information and the interfaces on which updates are sent and received, but not the actual contents of these updates. To view the routes contained in these updates, run **debug ip igrp transactions**. This command is also helpful when configuring unequal-cost load balancing with IGRP, since it shows the metrics of all possible routes.

Partial output of “debug ip igrp transactions”, displaying the routes contained in incoming and outgoing IGRP updates.

R1#**debug ip igrp transactions**

IGRP protocol debugging is on

```
19:17:51: IGRP: broadcasting request on Loopback0
19:17:51: IGRP: broadcasting request on Serial0
19:17:51: IGRP: broadcasting request on Serial1
19:17:51: IGRP: received update from 172.12.13.3 on Serial1
19:17:51:    subnet 172.12.13.0, metric 23531 (neighbor 21531)
19:17:51:    subnet 172.12.123.0, metric 23531 (neighbor 8476)
19:17:51:    network 1.0.0.0, metric 24031 (neighbor 8976)
19:17:51:    network 2.0.0.0, metric 22131 (neighbor 1600)
19:17:51:    network 3.0.0.0, metric 22031 (neighbor 501)
19:17:51:    network 172.23.0.0, metric 21631 (neighbor 1100)
```

How The Router Chooses Between Multiple Paths

If a router has only one path to a destination, obviously that will be the path taken. When multiple paths exist, the router will consider each route and then make a decision based on the *longest match* or on *administrative distance*.

Consider this routing table:

D	191.168.32.0/26 [90/25789217] via 10.1.1.1	(D = EIGRP route)
R	191.168.32.0/24 [120/4] via 10.1.1.2	(R = RIP)
O	191.168.32.0/19 [110/229840] via 10.1.1.3	(O = OSPF)

If a packet arrives on a router interface destined for 191.168.32.1, which route would the router choose? It depends on the prefix length, or the number of bits set in the subnet mask. Longer prefixes are always preferred over shorter ones when forwarding a packet.

In this case, a packet destined to 191.168.32.1 is directed toward 10.1.1.1, because 191.168.32.1 falls within the 191.168.32.0/26 network (191.168.32.0 to 191.168.32.63). It also falls within the other two routes available, but the 192.168.32.0/26 has the longest prefix within the routing table (26 bits vs. 24 or 19 bits).

What if the routes for a destination are all the same? Consider the same table, but with subnet masks of the same length for each protocol.

- D 191.168.32.0/24 [**90**/25789217] via 10.1.1.1
- R 191.168.32.0/24 [**120**/4] via 10.1.1.2
- O 191.168.32.0/24 [**110**/229840] via 10.1.1.3

The longest-match rule will not suffice in this situation. When routes are received via different protocols, and the longest-match rule does not result in selection of a route, *administrative distance* is then considered.

Administrative Distance

Administrative Distance was mentioned briefly in the RIP and IGRP sections of this chapter, but did not come into play since RIP and IGRP were not configured on the routers at the same time.

Administrative Distance is a measure of a route's believability when a router receives the same route **via two different protocols**. AD describes the order in which routing protocols as a whole are believed.

In the above routing table, the administrative distances are in bold. EIGRP has an AD of 90, OSPF has an AD of 110, and RIP has an AD of 120. The EIGRP route will be preferred.

Route Source	Administrative Distance
Directly Connected	0
Static	1
EIGRP Summary	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
ISIS	115
RIP	120

External EIGRP	170
iBGP	200

RIP version 1 – RIP version 2 – IGRP Comparison:

	RIP V 1	RIP v 2	IGRP
VLSM Support	No	Yes	No
Administrative Distance	110	110	100
Authentication Support	No	Yes, MD5 and Text	No
Equal-Cost Load Balancing	Yes	Yes	Yes
Unequal-Cost Load Balancing	No	No	Yes, with variance
Updates Sent To What Address	Broadcast 255.255.255.255	Multicast Address 224.0.0.9	Broadcast 255.255.255.255
Metric	Hop Count	Hop Count	Composite Metric involving Hop Count, Bandwidth, and Delay by default. Can also include Load and Reliability.
Default Paths Used In Load Balancing	4	4	4

Static Routing, IGRP, and RIP Q&A

1. The letter “C” in an IP routing table indicates what?
 - A. The route is static.
 - B. The route is directly connected.
 - C. The route is an IGRP route.
 - D. The route is a RIP v2 route.
 - E. The route is a RIP v1 route.

ANSWER: B. The “C” indicates that the network in the route statement is directly connected.

2. You wish to configure a static route to the network 172.12.12.0 /24. The exit interface will be Ethernet0.
Which of the following commands will perform this task?
 - A. ip route 172.12.12.0 255.255.255.0 ethernet0
 - B. ip route 172.12.12.0 0.255.255.255 ethernet0
 - C. ip route 172.12.12.0 255.255.0.0 ethernet0
 - D. ip route ethernet0 172.12.12.0 255.255.255.0
 - E. ip route ethernet0 172.12.12.0 255.255.0.0
 - F. ip route ethernet0 172.12.12.0 255.255.255.252

ANSWER: A. When you’re passing the CCNA exam, make sure to note the details. Be very careful when looking at subnet masks in multiple choice questions. There is no “ip static” or “route ip” command.

3. You wish to configure a default static route. The exit interface for this static route will be interface ethernet0, with an IP address of 172.12.12.13. Which commands below are correct configurations for this requirement?

- A. ip route 0.0.0.0 255.255.255.255 172.12.12.13
- B. ip route 0.0.0.0 0.0.0.0 172.12.12.13
- C. ip route all 172.12.12.13
- D. ip route default 172.12.12.13
- E. ip route 0.0.0.0 0.0.0.0 ethernet0
- F. ip route 0.0.0.0 255.255.255.255 ethernet0

ANSWER: E. The first part of the command to configure a static default route is “ip route 0.0.0.0 0.0.0.0”. If you use an IP address for a static route, the IP address used has to be the next-hop address, not the local IP address.

4. What metric do RIPv1 and RIPv2 use when making routing decisions?

- A. Bandwidth and delay.
- B. Hop count.
- C. Hop count, bandwidth, and delay.
- D. Advertised distance.
- E. Administrative distance.

ANSWER: B. RIPv1 and RIPv2 do have differences, but this isn't one of them. Hop count is the only metric RIP cares about.

5. The concept of “split horizon” states what?

- A. A loopback address can only be advertised by a routing protocol.
- B. A route advertised on a router interface must not be received on that same interface.
- C. A route received on a router interface cannot be advertised back out that same interface.
- D. A route will have a “split metric” if advertised out the same interface it was received on.

ANSWER: C. Split horizon prevents routes from being advertised out the same interface upon which it was received.

6. The concept of “route poisoning” states what?

- A. A route’s metric will be set to 32 if the advertising router determined the route can no longer be reached.
- B. A route’s metric will be set to 16 if the advertising router determined the route can no longer be reached.
- C. A route is automatically removed from all routing tables in the network if the hub router determines it cannot be reached.
- D. A route is no longer advertised if the advertising router determines it can no longer be reached.

ANSWER: B. The metric is set to 16 in this case, making it “unreachable” for distance-vector networks. Remember that the route is still advertised. Review the distance-vector behavior section of the Course Guide for a fully illustrated explanation.

7. In the Cisco concept of “split horizon with poison reverse”, what happens when a router realizes there has been a route failure?

- A. In accordance with split horizon, the route is advertised with a metric of 16, and is advertised out all interfaces except the one the advertisement was received on.
- B. The route will be advertised with a metric of 16, but will be advertised out all interfaces, including the one on which it was received.
- C. The route advertisements are immediately filtered. The defunct route will not be advertised.
- D. The router will send poison reverse packets to all routers, except the one that advertised the route in the first place.

ANSWER: B. Though split horizon prevents a route from being advertised out the same interface it was received on, split horizon with poison reverse sends the poisoned route out all interfaces.

8. What feature of distance vector protocols allows a router to send immediate routing updates upon a change in the network topology, rather than waiting for the regularly scheduled update?

- A. Triggered Updates
- B. Split Horizon
- C. Dynamic Updates
- D. Holddown Time

ANSWER: A. Triggered updates send immediate notification of network changes.

9, You wish to enable RIP on all interfaces of a router that are on the 10.0.0.0 network. Which of the following configurations serve that purpose?

- A. router rip
network 10.0.0.0 255.0.0.0
- B. router rip
network 10.0.0.0 all
- C. router rip
network 10.0.0.0
- D. router rip
10.0.0.0 network

ANSWER: C. “A” is incorrect, because subnet masks are not included in the network command, and “B” is wrong because there is no “all” keyword for the network command. “D” has the network number and the network command in the wrong positions.

10. Which of the following are true of RIP version 1?

- A. RIP v1 is classless.
- B. RIP v1 is classful.
- C. RIP v1 understands variable-length subnet masking.
- D. RIP v1 does not understand variable-length subnet masking.
- E. RIP v1 allows authentication.
- F. RIP v1 does not allow authentication.

ANSWER: B, D, F. There is no authentication with RIP v1, no use of VLSMs, and RIPv1 is classful.

11. Consider the following table:

R1#show ip route

Gateway of last resort is not set

< A default route has not been set. >

```
1.0.0.0/27 is subnetted, 1 subnets
C 1.1.1.0 is directly connected, Loopback0
R 2.0.0.0/8 [120/1] via 172.12.21.2, 00:00:24, Dialer1
      [120/1] via 172.12.123.2, 00:00:24, Serial0
R 3.0.0.0/8 [120/1] via 172.12.123.3, 00:00:14, Serial0
      [120/1] via 172.12.13.3, 00:00:14, Serial1
R 172.23.0.0/16 [120/1] via 172.12.13.3, 00:00:28, Serial1
      [120/1] via 172.12.123.3, 00:00:28, Serial0
      [120/1] via 172.12.123.2, 00:00:25, Serial0
      [120/1] via 172.12.21.2, 00:00:25, Dialer1
```

Which of the following statements are true? (Select all that apply.)

- A. The ISDN connection is up.
- B. Equal-cost load sharing is in effect.
- C. The number “120” in the routing statements refers to RIP’s administrative distance.
- D. The number “120” in the routing statements refers to RIP’s metric to the destination.
- E. The number “1” refers to the number of hops to the destination.

- F. The number of hops to the destination cannot be seen in the routing table.

ANSWER: A, B, C, E. Routes were received less than 30 seconds ago over interface Dialer1, so it's a safe assumption the ISDN link is up. Equal-cost load sharing is in effect by default, since multiple routes to the same destination are in the table. The "120" in the table is RIP's administrative distance. The number "1" refers to the metric, which in RIP's case, is hop count.

12. What is the maximum number of paths that can be utilized by RIP in equal-cost load balancing?

- A. Three
- B. Four
- C. Five
- D. Six
- E. Seven

ANSWER: D. Careful – four is the default number of paths for load balancing; six is the maximum.

13. What types of authentication does RIPv1 provide?

- A. MD5 only.
- B. MD5 and clear-text.
- C. Clear-text only.
- D. None.

ANSWER: D. RIPv1 does not support authentication.

14. Consider this router output:

```
R1# < command removed >
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface      Send  Recv  Key-chain
    Dialer1        2     2
    Loopback0      2     2
    Serial0        2     2      RIP
    Serial1        2     2
  Routing for Networks:
    1.0.0.0
    172.12.0.0
  Routing Information Sources:
    Gateway      Distance   Last Update
    172.12.13.3      120      00:00:07
    172.12.21.2      120      00:00:02
```

Which of the following statements are true?

- A. The command used here was “show ip rip”.
- B. The command used here was “show ip protocols”.
- C. The RIP defaults for timers are being used.
- D. The RIP defaults for versions sent and received are in use.
- E. The RIP default for administrative distance is in use.
- F. The RIP default for administrative distance has been changed.

ANSWER: B, C, E. The command is “show ip protocols”; the default timer values are being used (and the version defaults are not in use; only version 2 is being sent and received), and the RIP default administrative distance of 120 is in use (under “routing information sources”).

15. Which of the following statements about RIP version 2 authentication are FALSE?

- A. If authentication is run on one link in a RIP network, it must be run on all of them.
- B. RIP version 2 uses key-chains and interface-level commands to perform authentication; no commands are added under “router rip”.
- C. RIP version 2 allows for clear-text and MD authentication.
- D. If authentication is configured on a RIP link, both routers must agree on the authentication type as well as the password.

ANSWER: A. RIP authentication can be run on a single link in the network without running on all of them. The other three statements are true.

16. To what address does RIP version 1 send updates?

- A. The updates are unicast to all known RIP neighbors.
- B. The updates are multicast to 224.0.0.9.
- C. The updates are broadcast to 255.255.255.255.
- D. The updates are broadcast to 0.0.0.0.

ANSWER: C. RIP version 1 broadcasts routing updates.

17. To what address does RIP version 2 send updates?

- A. The updates are unicast to all known RIP neighbors.
- B. The updates are multicast to 224.0.0.9.
- C. The updates are multicast to 224.0.0.10.
- D. The updates are broadcast to 255.255.255.255
- E. The updates are broadcast to 0.0.0.0.

ANSWER: B. RIP version 2 updates are multicast to 224.0.0.9.

18. When configuring RIP, what effect does the command “passive interface” have?

- A. The router will neither receive nor accept routes; it becomes a “RIP placeholder” router.
- B. The passive interface will not receive routes or send them, but other interfaces will function normally.
- C. The passive interface will not receive routes, but will send them, and other interfaces will function normally.
- D. The passive interface will not send routes, but will receive them, and other interfaces will function normally.

ANSWER: D. Passive interfaces will not send route updates, but will receive them. This command affects only the interface named in the command. (Note that this is not an interface-level command; it's actually configured under “router rip”.)

19. When configuring IGRP, what is meant by the term “autonomous system”?

- A. A group of routers, connected physically, all running IGRP.
- B. A logical group of routers running IGRP.
- C. A group of IGRP routers exchanging routing updates with non-IGRP routers.
- D. A group of IGRP routers that under no circumstances will exchange routes with other routers.

ANSWER: B. An autonomous system is simply a logical group of routers running IGRP. They can exchange routes with non-IGRP speaking routers under certain circumstances, such as with route redistribution.

20. What is the Administrative Distance of an IGRP route?

- A. 100
- B. 105
- C. 110
- D. 120
- E. 130
- F. 140

ANSWER: A. An IGRP route has an Administrative Distance of 100.

21. What letter in a routing table indicates an IGRP route?

- A. E
- B. I
- C. G
- D. D
- E. F

ANSWER: B. The letter "I" indicates an IGRP route.

22. Which statement is true of default IGRP metric computation?

- A. Like RIP, IGRP only takes hop count into consideration.
- B. IGRP considers bandwidth and delay as well as hop count.
- C. IGRP considers bandwidth and delay, but not hop count.
- D. IGRP considers bandwidth, delay, load, and reliability.

ANSWER: B. IGRP does consider hop count, but also takes bandwidth and delay into consideration when calculating the route metric. Load and reliability can be used in the metric computation, but that is not a default behavior.

23. What assumption does IGRP make of a Serial interface?

- A. IGRP assumes the Serial interface is running at 56 kbps.
- B. IGRP assumes the Serial interface is running at 256 kbps.
- C. IGRP assumes the Serial interface is running at T1 speed.
- D. IGRP assumes the Serial interface is running at E1 speed.

ANSWER: C. IGRP assumes all Serial interfaces are connected to a T1 line, which runs at 1544 kbps.

24. What can be done about IGRP's default assumption of bandwidth?

- A. Nothing. This default behavior helps prevent routing loops.
- B. The "bandwidth" command can be used under the IGRP routing process.
- C. The "bandwidth" command can be used at the interface level.
- D. The "variance" command can be used under the IGRP routing process.

ANSWER: C. The interface-level command "bandwidth" is used to change IGRP's assumption of a Serial interface's bandwidth.

25. You are performing IGRP unequal-cost load balancing over three links. Two of the links are running at 1544 kbps; the other is running at 512 kbps, and you have configured "bandwidth 512" on that link. What is the default behavior of the load balancing?

- A. The load will be split equally over the three links.
- B. The faster link will carry approximately three times as much data as the slower link.
- C. IGRP does not support unequal-cost load balancing.
- D. The slower link will actually carry more data than the faster links. This issue was fixed with EIGRP (Enhanced IGRP).

ANSWER: B. By default, IGRP load balancing will send proportionally more data over the faster links. Since the faster links

are three times as fast as the slow link, that proportion will carry over to the data load.

26. You are performing IGRP unequal-cost load balancing over three links. Two of the links are running at 1544 kbps; the other is running at 512 kbps, and you have configured “bandwidth 512” on that link. You want the slower link to carry approximately the same amount of data as either of the faster links. What is your next step?

- A. Tell your boss this can't be done, because it can't be done.
- B. Use the “traffic-share multiple” command under the IGRP process.
- C. Use the “traffic-share balanced” command under the IGRP process.
- D. Use the “traffic share all” command on the appropriate interfaces.

ANSWER: B. The “traffic-share balanced” command will place an approximately equal load on all three links.

Static Routing Lab

Create a static route on R3 and one on R1 that will allow R3 to successfully ping R2's loopback interface, 2.2.2.2. The route should only consider traffic destined for 2.2.2.2. Use **show ip route** to display the static routes.

Configuring static routes with “ip route”.

```
R3#conf t
R3(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.1
R3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
2.0.0.0/32 is subnetted, 1 subnets
S  2.2.2.2 [1/0] via 172.12.123.1
3.0.0.0/27 is subnetted, 1 subnets
C    3.3.3.0 is directly connected, Loopback0
     172.12.0.0/24 is subnetted, 2 subnets
C      172.12.13.0 is directly connected, Serial1
C      172.12.123.0 is directly connected, Serial0.31
     172.23.0.0/27 is subnetted, 1 subnets
C      172.23.23.0 is directly connected, Ethernet0
```

```
R1#conf t
R1(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.2
```

```
R1#show ip route
< codes deleted for clarity >
```

Gateway of last resort is not set

```
1.0.0.0/27 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
2.0.0.0/32 is subnetted, 1 subnets
S    2.2.2.2 [1/0] via 172.12.123.2
     172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C      172.12.13.0/24 is directly connected, Serial1
C      172.12.21.0/30 is directly connected, Dialer1
C      172.12.123.0/24 is directly connected, Serial0
```

Examining the syntax of the “ip route” commands used in this lab:

R3(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.1

“ip route”: The command.

2.2.2.2 : The destination address.

255.255.255.255: The wildcard mask. This particular mask means that only traffic destined for 2.2.2.2 will use this static route.

172.12.123.1: The next-hop IP address used to reach the destination.

R1(config)#ip route 2.2.2.2 255.255.255.255 172.12.123.2

“ip route”: The command.

2.2.2.2: The destination address.

255.255.255.255: The wildcard mask. Again, only traffic destined for 2.2.2.2 will use this static route.

172.12.123.2: The next-hop IP address used to reach this destination.

On R3, run **debug ip packet**, then ping 2.2.2.2. The pings will return successfully, and the packets can be seen leaving and entering the router. Turn all debugs off with **undebbug all**.

```
R3#debug ip packet
IP packet debugging is on
R3#ping 2.2.2.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 132/136/144 m

R3#

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

IP: s=172.12.123.3 (local), d=2.2.2.2 (Serial0.31), len 100, sending

IP: s=2.2.2.2 (Serial0.31), d=172.12.123.3 (Serial0.31), len 100, rcvd 3

R3#**undebbug all**

All possible debugging has been turned off

Remove the static routes with the command **no ip route**. Replace them with a static route with a destination and wildcard mask of 0.0.0.0. This route will serve as a default route; to verify this, run **show ip route** after configuring these default static routes.

Removing the previously configured static routes, and replacing them with default static routes.

```
R3#conf t
R3(config)#no ip route 2.2.2.2 255.255.255.255 172.12.123.1
R3(config)#ip route 0.0.0.0 0.0.0.0 172.12.123.1

R1#conf t
R1(config)#no ip route 2.2.2.2 255.255.255.255 172.12.123.2
R1(config)#ip route 0.0.0.0 0.0.0.0 172.12.123.2
```

A static route configured with a destination and subnet mask of 0.0.0.0 will serve as a default route.

Examining the routing table of R3 after configuring the default static route.

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 172.12.123.1 to network 0.0.0.0

- 3.0.0.0/27 is subnetted, 1 subnets
 - C 3.3.3.0 is directly connected, Loopback0
 - 172.12.0.0/24 is subnetted, 2 subnets
 - C 172.12.13.0 is directly connected, Serial1
 - C 172.12.123.0 is directly connected, Serial0.31
 - 172.23.0.0/27 is subnetted, 1 subnets
 - C 172.23.23.0 is directly connected, Ethernet0
- S* 0.0.0.0/0 [1/0] via 172.12.123.1**

The static route appears on R3 as a candidate default route, and is then used as the default route. The “gateway of last resort” is now set as well to 172.12.123.1, the next-hop address of the static default route.

Examining R1's routing table after configuring the static default route.

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 172.12.123.2 to network 0.0.0.0

1.0.0.0/27 is subnetted, 1 subnets
C 1.1.1.0 is directly connected, Loopback0
172.12.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 172.12.13.0/24 is directly connected, Serial1
C 172.12.21.0/30 is directly connected, Dialer1
C 172.12.123.0/24 is directly connected, Serial0
S* 0.0.0.0/0 [1/0] via 172.12.123.2

R1 is also using the static route as a default route. The gateway of last resort is set to 172.12.123.2, the next-hop address set in the static default route.

RIP Lab: Configuring RIP Version 1; using show and debug commands.

Remove any existing routing protocol configuration from your network, including interface and subinterface commands.

Configure RIP version 1 on all three routers. Run RIP over all interfaces interconnecting the routers, and the loopback interfaces. On R1 and R2, configure the appropriate dialer interfaces as passive with the **passive-interface** command to prevent the ISDN line from staying up due to RIP routing updates.

Configuring RIP Version 1 on R1, R2, and R3, with passive-interface on the dial interfaces.

```
R1#conf t
R1(config)#router rip
R1(config-router)#version 1
R1(config-router)#passive-interface dialer1
R1(config-router)#network 172.12.0.0
R1(config-router)#network 1.0.0.0
```

```
R2#conf t
R2(config)#router rip
R2(config-router)#version 1
R2(config-router)#passive-interface bri0
R2(config-router)#network 172.12.0.0
R2(config-router)#network 172.23.0.0
R2(config-router)#network 2.0.0.0
```

```
R3#conf t
R3(config)#router rip
R3(config-router)#version 1
R3(config-router)#network 172.12.0.0
R3(config-router)#network 3.0.0.0
R3(config-router)#network 172.23.0.0
```

Run **show dialer** on R1 to ensure the ISDN line is not up due to RIP version 1 broadcast updates.

```
R1#show dialer
```

BRI0 - dialer type = ISDN

Dial String Successes Failures Last called Last status
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

BRI0:2 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

Dialer1 - dialer type = DIALER PROFILE

Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

The line is down, so **passive-interface** was correctly configured. If the dial interfaces had not been configured as passive, the RIP version 1 updates sent to 255.255.255.255 would have brought the line up (due to all IP traffic being configured as interesting in an earlier lab), and the broadcasts would have kept the line up.

Run **show ip route rip** on R1 to view the RIP routing table.

```
R1#show ip route rip
```

```
R  2.0.0.0/8 [120/1] via 172.12.123.2, 00:00:09, Serial0
R  3.0.0.0/8 [120/1] via 172.12.13.3, 00:00:04, Serial1
               [120/1] via 172.12.123.3, 00:00:04, Serial0
R  172.23.0.0/16 [120/1] via 172.12.123.2, 00:00:09, Serial0
               [120/1] via 172.12.13.3, 00:00:04, Serial1
               [120/1] via 172.12.123.3, 00:00:04, Serial0
```

RIP version 1 does not support VLSM. The loopbacks were configured with /27 subnet masks, but the classful /8 mask is shown for the remote loopbacks.

Load balancing is also taking place. There are three equal-cost routes from R1 to 172.23.0.0 (remember, no updates are being received on the dial interface). RIP version 1 supports equal-cost load balancing. Up to four equal-cost routes can be installed by default; change this to the Cisco maximum of six on all routers with the router configuration command **maximum-paths**.

Configuring the routers to install a maximum of six equal-cost routes for a single destination into their routing tables.

```
R1#conf t  
R1(config)#router rip  
R1(config-router)#maximum-paths 6
```

```
R2#conf t  
R2(config)#router rip  
R2(config-router)#maximum-paths 6
```

```
R3#conf t  
R3(config)#router rip  
R3(config-router)#maximum-paths 6
```

Display the type of update sent by RIP version 1, and the routes being advertised by and to R1, with **debug ip rip** and **clear ip route ***.

*Partial output of “debug ip rip” on R1 after clearing the routing table with “clear ip route *”*

```
R1#debug ip rip  
RIP protocol debugging is on  
R1#clear ip route *  
22:01:04: RIP: sending v1 update to 255.255.255.255 via Serial0 (172.12.123.1)  
22:01:04:   subnet 172.12.13.0, metric 1  
22:01:04:   subnet 172.12.123.0, metric 1  
22:01:04:   network 1.0.0.0, metric 1  
22:01:04:   network 2.0.0.0, metric 2  
22:01:04:   network 3.0.0.0, metric 2  
22:01:04:   network 172.23.0.0, metric 2  
22:01:04: RIP: sending v1 update to 255.255.255.255 via Serial1 (172.12.13.1)  
22:01:04:   subnet 172.12.123.0, metric 1  
22:01:04:   network 1.0.0.0, metric 1  
22:01:04:   network 2.0.0.0, metric 2  
22:01:06: RIP: sending general request on Loopback0 to 255.255.255.255  
22:01:06: RIP: sending general request on Serial0 to 255.255.255.255  
22:01:06: RIP: sending general request on Serial1 to 255.255.255.255  
22:01:07: RIP: received v1 update from 172.12.123.3 on Serial0
```

RIP version 1 uses broadcasts (255.255.255.255) to send and receive updates.

On R3, use **show ip protocols** to view the RIP update timers.

```
R3#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 9 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
```

On all three routers, double the default timers as shown with the **timers basic** command. Note that when changing RIP timers with this command, the timers must be changed on every router in the RIP domain. Failure to do so will result in unpredictable routing behavior.

The syntax of this command:

```
timers basic update invalid holddown flush
```

Each timer value is expressed in seconds. Defining each timer:

Update: Rate at which RIP updates are sent. Default is 30 seconds.

Invalid: If no update is received in this period of time, the route is marked as inaccessible and advertised as unreachable. Default is 180 seconds.

Holddown: Period of time during which routing information informing the router of better paths are suppressed. A route enters holddown when an update packet is received that indicates the route is unreachable. Default is 180 seconds.

Flush: Period of time that must elapse before the route is removed from the routing table. This value must be larger than the update and holddown values. Default is 240 seconds.

*Cisco IOS Help is used to show the four values that can be changed with **timers basic**.*

```
R3#conf t
R3(config)#router rip
R3(config-router)#timers basic ?
<0-4294967295> Interval between updates
R3(config-router)#timers basic 60 ?
<1-4294967295> Invalid
R3(config-router)#timers basic 60 360 ?
<0-4294967295> Holddown
R3(config-router)#timers basic 60 360 360 ?
<1-4294967295> Flush
R3(config-router)#timers basic 60 360 360 480
```

Notes

Lab: Configuring RIP Version 2.

Disabling auto-summarization; using text and MD5 authentication; Troubleshooting RIP with show and debug commands.

Configure RIP version 2 on all three routers. Disable RIP's auto-summarization feature with **no auto-summary**. Enable RIP on all interfaces of each router, including the loopbacks. Prevent the dialer interfaces from sending RIP version 2 multicasts with the **passive-interface** command.

```
R1#conf t
R1(config)#router rip
R1(config-router)#version 2
< The RIP-enabled interfaces will receive and send version 2 only. >
R1(config-router)#no auto-summary
R1(config-router)#network 172.12.0.0
R1(config-router)#network 1.0.0.0
R1(config-router)#passive-interface dialer1

R2#conf t
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#passive-int bri0
R2(config-router)#network 172.12.0.0
R2(config-router)#network 172.23.0.0
R2(config-router)#network 2.0.0.0

R3#conf t
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)#network 172.12.0.0
R3(config-router)#network 172.23.0.0
R3(config-router)#network 3.0.0.0
```

To verify VLSM support and equal-cost load-balancing, run **show ip route rip** on R1.

```
R1#show ip route rip
2.0.0.0/27 is subnetted, 1 subnets
R    2.2.2.0 [120/1] via 172.12.123.2, 00:00:15, Serial0
3.0.0.0/27 is subnetted, 1 subnets
R    3.3.3.0 [120/1] via 172.12.13.3, 00:00:14, Serial1
                  [120/1] via 172.12.123.3, 00:00:14, Serial0
    172.23.0.0/27 is subnetted, 1 subnets
R    172.23.23.0 [120/1] via 172.12.123.2, 00:00:15, Serial0
                  [120/1] via 172.12.13.3, 00:00:14, Serial1
                  [120/1] via 172.12.123.3, 00:00:15, Serial0
```

VLSM support is evident from the non-classful subnet masks for networks 2.0.0.0 and 3.0.0.0. Equal-cost load balancing is taking place as well, with three routes sharing the load from R1 to network 172.23.23.0.

From each router, ping the remote loopback addresses. All pings should succeed.

After the routers pass that connectivity test, configure R1, R2, and R3 for RIP text authentication over the Frame Relay cloud. Configure a key chain called FRAME with key number 1 and key-string CISCO.

Configuring the key chain, key number, and key-string to be used in RIP authentication.

```
R1#conf t
R1(config)#key chain FRAME
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string CISCO
```

```
R2#conf t
R2(config)#key chain FRAME
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string CISCO
```

```
R3#conf t
R3(config)#key chain FRAME
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CISCO
```

On R1, configure the Serial0 interface for text authentication, using the key chain FRAME. Do the same on R2's multipoint serial interface and R3's point-to-point serial interface.

```
R1#conf t
R1(config)#int s0
R1(config-if)#ip rip authentication mode text
R1(config-if)#ip rip authentication key-chain FRAME

R2#conf t
R2(config)#int s0.123
R2(config-subif)#ip rip authentication mode text
R2(config-subif)#ip rip authentication key-chain FRAME

R3#conf t
R3(config)#int s0.31
R3(config-subif)#ip rip authentication mode text
R3(config-subif)#ip rip authentication key-chain FRAME
```

On R1, run **show ip protocols** to view what version of RIP is running on the router interfaces, the name of any key-chains, and any passive-interfaces.

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 14 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
Default version control: send version 2, receive version 2
  Interface      Send  Recv  Key-chain
  Loopback0      2    2
  Serial0        2    2    FRAME
  Serial1        2    2
Routing for Networks:
  1.0.0.0
  172.12.0.0
Passive Interface(s):
Dialer1
Routing Information Sources:
  Gateway      Distance   Last Update
  172.12.13.3    120     00:00:02
  172.12.123.3   120     00:00:02
  172.12.123.2   120     00:00:22
Distance: (default is 120)
```

On R1, run **debug ip rip** and clear the routing table with **clear ip route *** to examine the contents of the routing updates and the authentication type, if any.

```
R1#debug ip rip
RIP protocol debugging is on
R1#clear ip route *
22:51:38: RIP: sending general request on Loopback0 to 224.0.0.9
22:51:38: RIP: sending general request on Serial0 to 224.0.0.9
22:51:38: RIP: sending general request on Serial1 to 224.0.0.9
22:51:38: RIP: received v2 update from 172.12.13.3 on Serial1
22:51:38:    172.12.13.0/24 -> 0.0.0.0 in 1 hops
22:51:38:    172.12.123.0/24 -> 0.0.0.0 in 1 hops
22:51:38:    2.2.2.0/27 -> 0.0.0.0 in 2 hops
22:51:38:    3.3.3.0/27 -> 0.0.0.0 in 1 hops
22:51:38:    172.23.23.0/27 -> 0.0.0.0 in 1 hops
22:51:38: RIP: sending v2 update to 224.0.0.9 via Loopback0 (1.1.1.1)
22:51:38:    2.2.2.0/27 -> 0.0.0.0, metric 3, tag 0
22:51:38:    3.3.3.0/27 -> 0.0.0.0, metric 2, tag 0
22:51:38:    172.12.13.0/24 -> 0.0.0.0, metric 1, tag 0
22:51:38:    172.12.21.0/30 -> 0.0.0.0, metric 1, tag 0
22:51:39:    172.12.123.0/24 -> 0.0.0.0, metric 1, tag 0
22:51:39:    172.23.23.0/27 -> 0.0.0.0, metric 2, tag 0
22:51:39: RIP: sending v2 update to 224.0.0.9 via Serial0 (172.12.123.1)
22:51:39:    172.12.13.0/24 -> 0.0.0.0, metric 1, tag 0
22:51:39:    172.12.21.0/30 -> 0.0.0.0, metric 1, tag 0
22:51:39:    172.12.123.0/24 -> 0.0.0.0, metric 1, tag 0
22:51:39:    1.1.1.0/27 -> 0.0.0.0, metric 1, tag 0
22:51:39:    2.2.2.0/27 -> 0.0.0.0, metric 3, tag 0
22:51:39:    3.3.3.0/27 -> 0.0.0.0, metric 2, tag 0
22:51:39:    172.23.23.0/27 -> 0.0.0.0, metric 2, tag 0
22:51:41: RIP: received v2 update from 172.12.13.3 on Serial1
22:51:41:    172.12.123.0/24 -> 0.0.0.0 in 1 hops
22:51:41:    2.2.2.0/27 -> 0.0.0.0 in 2 hops
22:51:41:    3.3.3.0/27 -> 0.0.0.0 in 1 hops
22:51:41:    172.23.23.0/27 -> 0.0.0.0 in 1 hops
22:51:41: RIP: received packet with text authentication CISCO
22:51:41: RIP: received v2 update from 172.12.123.3 on Serial0
22:51:41:    172.12.13.0/24 -> 0.0.0.0 in 1 hops
22:53:12: RIP: received packet with text authentication CISCO
22:53:12: RIP: received v2 update from 172.12.123.2 on Serial0
22:53:12:    172.12.21.0/30 -> 0.0.0.0 in 1 hops
22:53:12:    2.2.2.0/27 -> 0.0.0.0 in 1 hops
22:53:12:    3.3.3.0/27 -> 0.0.0.0 in 2 hops
22:53:12:    172.23.23.0/27 -> 0.0.0.0 in 1 hops
```

Updates are being sent to 224.0.0.9, the RIP version 2 multicast address. Routes from 172.12.123.3 and 182.12.123.2 are received with text authentication with a key-string of CISCO. (Note that this is the key-string contained in the key, not the name of the key chain.) All other RIP updates are sent and received with no authentication.

Create a separate key chain on R1 and R3 to authenticate updates sent over the directly connected interfaces. This key chain will be called DIRECT, use key number 1, and a key-string of CCNA.

```
R1#conf t
R1(config)#key chain DIRECT
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string CCNA

R3#conf t
R3(config)#key chain DIRECT
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CCNA
```

View the key chains configured on R1 and R3 with **show key chain**.

```
R1#show key chain
Key-chain FRAME:
  key 1 -- text "CISCO"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]

Key-chain DIRECT:
  key 1 -- text "CCNA"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]

R3#show key chain
Key-chain FRAME:
  key 1 -- text "CISCO"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]

Key-chain DIRECT:
  key 1 -- text "CCNA"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]
```

The key chains and keys are correctly configured. Keys can be configured with a valid time range; outside that time range, they cannot be used for authentication.

Troubleshooting With “show key chain”

The RIP section of this book discussed a common error in RIP authentication that is very difficult to spot. Hitting the space bar after entering the key-string, but before hitting <ENTER>, will result in a null space being placed at the end of the key-string. This extra space cannot be spotted by looking at the running configuration, but can be quickly spotted with **show key chain**.

An additional key chain has been added to R3, and the key-string has such a null value at the end. Running **show key chain** and looking closely at the end of the displayed key-string can reveal an issue with authentication:

```
R3#show key chain
```

Key-chain FRAME:

```
key 1 -- text "CISCO"
```

```
    accept lifetime (always valid) - (always valid) [valid now]
```

```
    send lifetime (always valid) - (always valid) [valid now]
```

Key-chain DIRECT:

```
key 1 -- text "CCNA"
```

```
    accept lifetime (always valid) - (always valid) [valid now]
```

```
    send lifetime (always valid) - (always valid) [valid now]
```

Key-chain TROUBLESHOOTING:

```
key 1 -- text "CISCO "
```

```
    accept lifetime (always valid) - (always valid) [valid now]
```

```
    send lifetime (always valid) - (always valid) [valid now]
```

Take a close look at the text strings. There is a visible space at the end of the CISCO key-string in the TROUBLESHOOTING key chain. There is none at the end of either of the other two. This command should be run immediately upon realizing there is a problem with RIP authentication, as this space is the most common problem.

On R1 and R3, configure the directly connected interfaces for RIP authentication using MD5 and the key chain DIRECT.

```
R1#conf t
R1(config)#int s1
R1(config-if)#ip rip authentication mode md5
R1(config-if)#ip rip authentication key-chain DIRECT

R3#conf t
R3(config)#int s1
R3(config-if)#ip rip authentication mode md5
R3(config-if)#ip rip authentication key-chain DIRECT
```

On R1, check what interfaces are running the different key chains with **show ip protocols**.

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 6 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface      Send   Recv  Key-chain
    Loopback0      2      2      FRAME
    Serial0        2      2      FRAME
    Serial1        2      2      DIRECT
  Routing for Networks:
    1.0.0.0
    172.12.0.0
  Passive Interface(s):
    BRI0
    Dialer1
  Routing Information Sources:
    Gateway      Distance   Last Update
    172.12.13.3      120      00:00:18
    172.12.123.3     120      00:00:18
    172.12.123.2     120      00:00:18
  Distance: (default is 120)
```

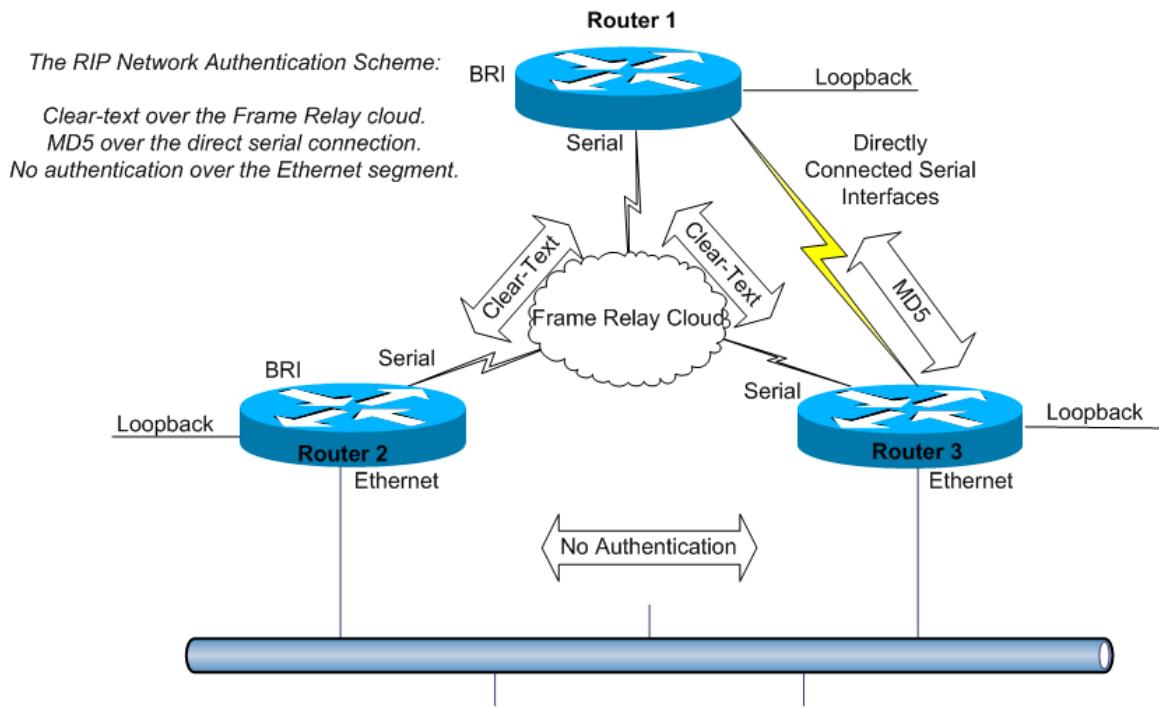
On R1, run **debug ip rip** and clear the routing table with **clear ip route ***.

Partial output of “debug ip rip” on R1 after clearing the routing table.

```
R1#debug ip rip
RIP protocol debugging is on
R1#clear ip route *
23:22:05: RIP: sending general request on Loopback0 to 224.0.0.9
23:22:05: RIP: sending general request on Serial0 to 224.0.0.9
23:22:05: RIP: sending general request on Serial1 to 224.0.0.9
23:22:12: RIP: sending v2 update to 224.0.0.9 via Serial0 (172.12.123.1)
23:22:12:   172.12.13.0/24 -> 0.0.0.0, metric 1, tag 0
23:22:12:   172.12.21.0/30 -> 0.0.0.0, metric 1, tag 0
23:22:12:   172.12.123.0/24 -> 0.0.0.0, metric 1, tag 0
23:22:12:   1.1.1.0/27 -> 0.0.0.0, metric 1, tag 0
23:22:12: RIP: sending v2 update to 224.0.0.9 via Serial1 (172.12.13.1)
23:22:12:   172.12.21.0/30 -> 0.0.0.0, metric 1, tag 0
23:22:12:   172.12.123.0/24 -> 0.0.0.0, metric 1, tag 0
23:22:12:   1.1.1.0/27 -> 0.0.0.0, metric 1, tag 0
23:22:14: RIP: received packet with text authentication CISCO
23:22:14: RIP: received v2 update from 172.12.123.2 on Serial0
23:22:14:   172.12.21.0/30 -> 0.0.0.0 in 1 hops
23:22:14:   2.2.2.0/27 -> 0.0.0.0 in 1 hops
23:22:14:   3.3.3.0/27 -> 0.0.0.0 in 2 hops
23:22:14:   172.23.23.0/27 -> 0.0.0.0 in 1 hops
23:22:16: RIP: received packet with MD5 authentication
23:22:16: RIP: received v2 update from 172.12.13.3 on Serial1
23:22:16:   172.12.123.0/24 -> 0.0.0.0 in 1 hops
23:22:16:   2.2.2.0/27 -> 0.0.0.0 in 2 hops
23:22:16:   3.3.3.0/27 -> 0.0.0.0 in 1 hops
23:22:16:   172.23.23.0/27 -> 0.0.0.0 in 1 hops
23:22:16: RIP: received packet with text authentication CISCO
23:22:16: RIP: received v2 update from 172.12.123.3 on Serial0
23:22:16:   172.12.13.0/24 -> 0.0.0.0 in 1 hops
23:22:16:   2.2.2.0/27 -> 0.0.0.0 in 2 hops
23:22:16:   3.3.3.0/27 -> 0.0.0.0 in 1 hops
23:22:16:   172.23.23.0/27 -> 0.0.0.0 in 1 hops
```

R1 is now receiving three RIP authenticated updates. The router continues to receive text-authenticated updates from 172.12.123.3 and 172.12.123.2, and now receives MD5-authenticated RIP updates from 172.12.13.3.

Note that within a single RIP network, two different authentication types are in use, and one segment (the Ethernet segment) is not running authentication at all. This network will function correctly as long as both ends of a network segment agree on the type of authentication to run, or to not run it at all.



IGRP Lab

Remove any previous routing protocol configurations before proceeding, including interface and subinterface commands.

Configure IGRP on R1, R2, and R3 with the **router igrp 1** command. IGRP will run on all interfaces in the 172.12.0.0 network, the 172.23.0.0 network, and all loopback interfaces. IGRP updates should not bring the ISDN line up; configure **passive-interface** under the IGRP process.

```
R1#conf t
R1(config)#router igrp 1
R1(config-router)#network 172.12.0.0
R1(config-router)#network 1.0.0.0
R1(config-router)#passive-interface dialer1
```

*The “1” in the **router igrp** command refers to the Autonomous System (AS). IGRP is a classful routing protocol, so wildcard masks are not used in the network statements. **Passive-interface** prevents the named interface from sending routing updates out for this protocol, but the interface could still receive them.*

```
R2#conf t
R2(config-if)#router igrp 1
R2(config-router)#network 172.12.0.0
R2(config-router)#network 2.0.0.0
R2(config-router)#passive-interface bri0

R3#conf t
R3(config-if)#router igrp 1
R3(config-router)#network 172.12.0.0
R3(config-router)#network 3.0.0.0
```

Run **show ip route** on R1. R1 will see three equal-cost paths to the Ethernet network. IGRP supports load-sharing over up to four equal-cost paths by default, so all three paths appear in the routing table. R1 will also see a route to the loopback address on R2 and two routes to the loopback address on R3.

From each router, ping the loopback addresses of the other two routers. From R1, ping both R2’s and R3’s Ethernet interfaces.

Examining R1's routing table.

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

136.1.0.0/24 is subnetted, 1 subnets
C 136.1.136.0 is directly connected, Ethernet0/0
1.0.0.0/24 is subnetted, 1 subnets
C 1.1.1.0 is directly connected, Loopback0
I 2.0.0.0/8 [100/8976] via 172.12.123.2, 00:00:29, Serial0/0
I 3.0.0.0/8 [100/8976] via 172.12.13.3, 00:00:16, Serial1/0
[100/8976] via 172.12.123.3, 00:00:16, Serial0/0
172.12.0.0/24 is subnetted, 2 subnets
C 172.12.13.0 is directly connected, Serial1/0
C 172.12.123.0 is directly connected, Serial0/0
I 172.23.0.0/16 [100/8576] via 172.12.13.3, 00:00:17, Serial1/0
[100/8576] via 172.12.123.3, 00:00:17, Serial0/0
[100/8576] via 172.12.123.2, 00:00:30, Serial0/0

Remember that the numbers in the brackets following the network number in the routes are the Administrative Distance and the IGRP metric, in that order.

There are two serial connections between R1 and R3. IGRP is assuming that both lines are T1 lines, running at 1544 KBPS. If the direct connection between the routers was actually a 512 KBPS line, equal-cost load sharing would be occurring because of IGRP's bandwidth assumption, not because of the actual bandwidth.

To give IGRP a better idea of the network's actual topology, the **bandwidth** command is used. On the directly connected serial interface, configure the interface-level command **bandwidth 512**. Clear the routing tables after doing so and note the differences in the revised table and the original table.

*Examining the effect of the **bandwidth** command on IGRP metric and path availability.*

```
R1#conf t  
R1(config)#interface serial1/0  
R1(config-if)#bandwidth 512
```

```
R3#conf t  
R3(config)#interface serial 1/0  
R3(config-if)#bandwidth 512
```

*IGRP's assumption that all serial lines run at 1544 KBPS is overridden by the **bandwidth 512** command. IGRP now believes this line runs at 512 KBPS.*

```
R1#clear ip route *  
R1#show ip route igrp  
I 2.0.0.0/8 [100/8976] via 172.12.123.2, 00:00:17, Serial0/0  
I 3.0.0.0/8 [100/8976] via 172.12.123.3, 00:00:24, Serial0/0  
I 172.23.0.0/16 [100/8576] via 172.12.123.3, 00:00:24, Serial0/0  
[100/8576] via 172.12.123.2, 00:00:17, Serial0/0
```

*The routing table is cleared with **clear ip route ***. To see only the routes received in IGRP updates instead of the entire table, run **show ip route igrp**.*

One of the paths to 3.0.0.0 is gone from the table, as is one of the routes to 172.23.0.0. Both routes now gone from the table went through the 172.12.13.0 network. Now that IGRP sees that link as slower than the others, equal-cost load-balancing will not occur over the 172.12.13.0 network, and those two routes are removed from the IGRP routing table.

IGRP does support unequal-cost load-sharing, but it must be manually configured. The metric of the best route, the one installed in the routing table, is 8576. To find out what the metrics are for the other routes to 172.23.0.0 and 3.0.0.0, run **debug ip igrp transaction** on R1, then clear the routing table with **clear ip route ***.

```
R1#debug ip igrp transactions  
IGRP protocol debugging is on  
R1#clear ip route *  
  
05:40:07: IGRP: received update from 172.12.13.3 on Serial1/0  
05:40:07: subnet 172.12.123.0, metric 23531 (neighbor 8476)  
05:40:07: network 1.0.0.0, metric 24031 (neighbor 8976)  
05:40:07: network 2.0.0.0, metric 22131 (neighbor 1600)  
05:40:07: network 3.0.0.0, metric 22031 (neighbor 501)  
05:40:07: network 172.23.0.0, metric 21631 (neighbor 1100)
```

*This partial output of **debug ip igrp transactions** shows the IGRP update coming in from 172.12.13.3, the directly connected serial interface on R3. The metric for network 3.0.0.0 is 22031, and the metric for 172.23.0.0 is 21631.*

The variance command is used to configure unequal-cost load balancing with both IGRP and EIGRP. The variance value is a multiplier; multiplied by the metric of the best route, it must be larger than the metric of any feasible successor.

The concept is much clearer when actual metrics are used. The metric of the best route for both those routes is 8576. What number, multiplied by 8576, will be greater than 21631?

Three times 8576 is 25728. Configure variance 3 under the IGRP routing process on R1, clear the routing table, and display the IGRP routing table.

Examining the effect of variance 3 on the IGRP routing table.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router igrp 1
R1(config-router)#variance 3

R1#clear ip route *
R1#show ip route igrp
I 2.0.0.0/8 [100/22131] via 172.12.13.3, 00:00:04, Serial1/0
    [100/9076] via 172.12.123.3, 00:00:04, Serial0/0
    [100/8976] via 172.12.123.2, 00:00:04, Serial0/0
I 3.0.0.0/8 [100/22031] via 172.12.13.3, 00:00:04, Serial1/0
    [100/8976] via 172.12.123.3, 00:00:04, Serial0/0
    [100/9076] via 172.12.123.2, 00:00:04, Serial0/0
I 172.23.0.0/16 [100/21631] via 172.12.13.3, 00:00:04, Serial1/0
    [100/8576] via 172.12.123.3, 00:00:04, Serial0/0
    [100/8576] via 172.12.123.2, 00:00:04, Serial0/0
```

The variance command has two effects, one intended and one unintended. The routes to 172.23.0.0 and 3.0.0.0 through 172.12.13.3 are back in the routing table and will participate in unequal-cost load sharing. Note that the metrics themselves do not change.

There are now three routes to R2's loopback as well. There was only one, but the variance 3 command means that any feasible route to R2 with a metric of 26928 (8976 x 3) results in the installation of the other two routes, both with a metric lower than 26928.

Section Eight: Link-State Protocols and OSPF

The other major type of routing protocol is a link-state protocol. What you've got to do now is set aside everything you learned about distance-vector protocols, because little of it applies here. You've got to know the differences between link-state and distance vector protocols, and which is which.

OSPF and ISIS are the two big link-state protocols in use today; the CCNA exams concern themselves only with OSPF.

Commands Introduced In This Section and Labs:

router ospf <process number> : Enters router configuration mode for an OSPF process.

ip ospf priority <priority value> : Interface-level command setting the OSPF priority for the interface. Used during election of Designated Router and Backup Designated Router.

neighbor < neighbor IP address >: Used to statically identify OSPF neighbors when using the non-broadcast OSPF network type.

ip ospf hello-interval <interval value> : Interface-level command specifying how often OSPF Hellos should be sent out that interface.

ip ospf network <network type> : Interface-level command used to change the default OSPF network type of that interface.

show ip ospf neighbor : Displays OSPF neighbor relationships and their states.

debug ip ospf adj : Used to diagnose OSPF adjacency issues. Displays hello times of adjacency packets received from neighboring routers, a frequent cause of adjacency problems.

area < area number > virtual-link: Used to create a virtual link, allowing an area with no physical connection to Area 0 to create a logical connections.

The Concepts Behind Link-State Protocols

Distance vector protocols such as RIP and IGRP exchange routing information in broadcasts and multicasts. The routing information includes what routes the sending router knows, and what the metric is for getting there. Commands such as **debug ip rip** and **debug ip igrp transactions** show these routes and metrics as they enter and leave the distance vector protocol-enabled interfaces of a router.

Link-state protocols do not exchange routes and metrics. Link-state protocols exchange just that – the state of the links they know about, and the **cost** associated with those links. As a router running a link-state protocol receives these Link State Advertisements (LSA) from routers it has formed a **neighbor relationship** with, the router performs a series of computations on these LSAs, giving the router a complete picture of the link-state network. This series of computations is known as the **Shortest Path First (SPF) algorithm**, also referred to as the **Dijkstra algorithm**.

Before any link exchange can begin, a router with a newly configured link-state routing process such as Open Shortest Path First (OSPF) must discover its neighbors. The method of neighbor discovery varies with the different types of OSPF networks that exist; suffice to say that neighbors must be discovered and form an **adjacency**, after which LSAs will begin to be exchanged.

Link State Advertisements are actually “packaged” by the sending router into an Link State Update (LSU). After an LSU is received, the LSA topology information is placed into an OSPF database. The routers will run the Dijkstra algorithm to calculate the best routes, and then place those routes into the routing table.

Link-State Protocols Compared To Distance-Vector Protocols

	Distance-Vector Protocols	Link-State Protocols
How Are Routes Learned?	Via routing updates received from neighbors, which include the route and the metric.	The router receives Link State Advertisements (LSA) from neighbors, and then runs the Dijkstra SPF Algorithm to calculate the best routes.
How Do Routers Handle Updates?	Routers send out regularly scheduled updates, regardless of whether changes have actually occurred.	Once the routers in a Link State domain, or area , have calculated their routes, LSAs are sent only upon a change in the network topology.
What Address Are Routes or Updates Sent To?	RIP v 1 and IGRP broadcast routes to 255.255.255.255. RIP v 2 multicasts routes to 224.0.0.9.	OSPF will flood changes to 224.0.0.5.

Hello Packets: The “Heartbeat” Of OSPF

Once neighboring routers form an adjacency, some kind of keepalive is needed for each router to know the other one is still there, particularly since OSPF routers do not regularly exchange routes or send the other an LSU. OSPF routers send neighbors **hello packets** at regularly scheduled intervals. This default intervals are 10 seconds on an Ethernet segment and 30 seconds for non-broadcast links such as Serial links.

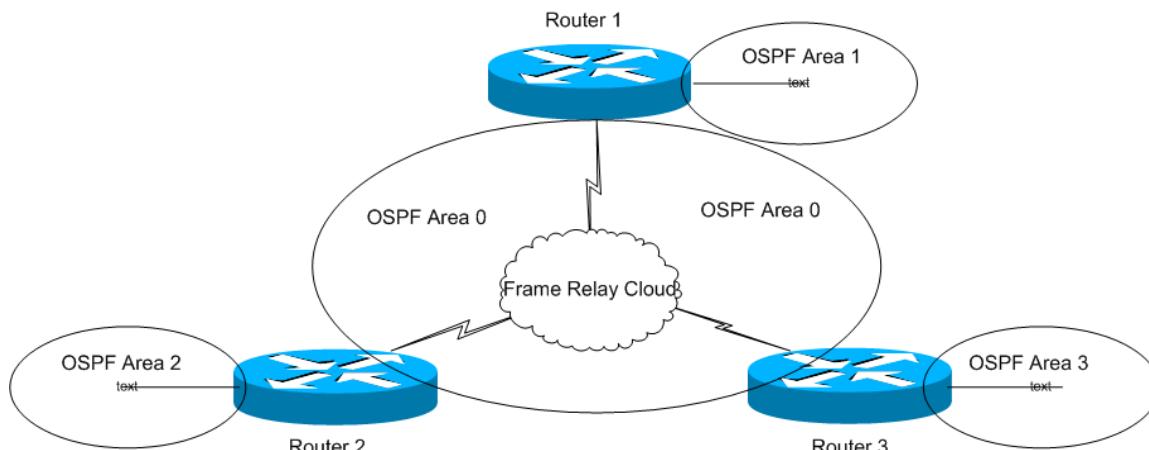
When a router goes a certain amount of time without receiving a Hello from a neighbor, the router declares the adjacency as dead. The SPF algorithm will recalculate due to the change in network topology. The default **dead time** for OSPF is four times the hello-interval, which makes it 40 seconds for Ethernet links and 120 seconds for non-broadcast links.

How The Dijkstra Algorithm Assists With Loop Prevention

Link-state protocols do not rely on distance-vector loop prevention methods such as split horizon or poison reverse. (These would not work with link-state protocols anyway; remember that link-state routers are not actually exchanging routes in the first place.) Instead, the Dijkstra Algorithm recalculates network changes so quickly that routing loops have no time to form.

Upon detection of a change in the network topology, such as a down link, the detecting router **floods** news of this change. The routers receiving this new information then flood the change to all its neighbors, and the process continues until all routers are informed of the change. The routers are running a new SPF Algorithm immediately after learning of the change, and new routes are determined almost immediately. This default link-state behavior is what makes OSPF convergence much faster than distance-vector convergence.

Configuring OSPF



Where the number in the initial IGRP indicates what Autonomous System the router is part of, the number in the **router ospf** command indicates the **process number**. OSPF can run multiple processes on one router, and the links are not advertised by default between processes.

OSPF uses the concept of **areas** to logically segment the network. Each router has its interface in the 172.12.123.0 /24 network in Area

- O. R1 has its loopback in Area 1; R2 has its loopback in Area 2 ; R3 has its loopback in Area 3.

This design is known as **hub-and-spoke**, which provides certain challenges in the configuration. The router in the hub must be the **designated router**, which is the router that will flood changes to all other routers. If one of the spokes becomes the DR, the configuration becomes invalid. By using the interface-level command **ip ospf priority 0** (zero) on both R2 and R3, these routers cannot become the DR even if R1 goes down. (The default priority for a OSPF interface is one.)

Examining The Initial OSPF Configuration

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#**router ospf 1**

< The number following “router ospf” refers to the OSPF process number. >

R1(config-router)#**network 172.12.123.0 0.0.0.255 area 0**

< OSPF uses wildcard masks in the network statements. This command indicates that any interface in the 172.12.123.0 /24 network will be placed into Area 0, the backbone. >

R1(config-router)#**network 1.1.1.0 0.0.0.15 area 1**

< Any interface in the 1.1.1.0 /28 network is placed into Area 1. The loopback interface falls in that network, so it will run in Area 1. >

R1(config-router)#**neighbor 172.12.123.2**

R1(config-router)#**neighbor 172.12.123.3**

*< By default, OSPF will run in “nonbroadcast” mode on a physical Serial interface. With a nonbroadcast network, manual **neighbor** statements must be configured in router configuration mode. >*

R2#conf t

R2(config)#int s0.123

R2(config-subif)#**ip ospf priority 0**

< The “ip ospf priority 0” interface-level command will prevent this router from ever becoming the Designated Router. Spoke routers in a hub-and-spoke network cannot be allowed to become the Designated Router. >

R2(config-subif)#**router ospf 1**

R2(config-router)#**network 172.12.123.0 0.0.0.255 area 0**

R2(config-router)#**network 2.2.2.0 0.0.0.15 area 2**

R3#conf t

R3(config)#**int s0.31 point**

R3(config-subif)#**ip ospf network non-broadcast**

< By default, a point-to-point interface will run as an OSPF point-to-point network type. This would be a network mismatch with R1’s serial interface and an adjacency would not form. The command “ip ospf network non-broadcast” changes the OSPF network type and allows an adjacency to form. >

R3(config-subif)#**ip ospf priority 0**

< “ip ospf priority 0” prevents this router from becoming a Designated Router.

R3(config)#**router ospf 1**

R3(config-router)#**network 172.12.123.0 0.0.0.255 area 0**

R3(config-router)#**network 3.3.3.0 0.0.0.15 area 3**

There were several concepts in that initial configuration that may not be required for the CCNA exam itself, but are important to truly understand how OSPF works:

The Designated Router

In a production OSPF network, if all directly connected routers had to form adjacencies with every other router, and continue to exchange LSAs with each, a large amount of bandwidth would be used any time a router flooded a network topology change.

The **designated router** is the router that will receive the LSAs from the other routers in the area. Then the DR will flood the LSA indicating the network change. Instead of having every router flooding the network with LSAs after a network change, the change notification is sent straight to the DR, and the DR then floods the network with the change.

If the DR fails, the **backup designated router (BDR)** takes its place.

The value used to elect the DR and BDR is the **OSPF interface priority**. By default, this value is one on all OSPF-enabled routers. To influence the election, the interface-level command **ip ospf priority** is used. Setting an interface's priority to zero prevents it from becoming the DR or BDR.

In a hub-and-spoke configuration, only the router directly connected to the others (the "hub") should become the DR. The other routers should not be allowed to even participate in the election for DR or BDR. The interface-level command **ip ospf priority 0** prevents an interface's participation in either election.

OSPF Network Types

To view the OSPF network type of an interface, run **show ip ospf interface <type.number>**:

```
R1#show ip ospf interface s0
Serial0 is up, line protocol is up
  Internet Address 172.12.123.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 172.12.123.1
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:10
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 3.3.3.3
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
```

The OSPF network type should be the same on interfaces attempting to form adjacencies. Use **show ip ospf interface** to see the OSPF network type, and change it as needed with the interface-level command **ip ospf network**. The command **show ip ospf neighbor** will show if the adjacencies have formed.

Examining R3's OSPF network type on interface serial 0.31.

```
R3#conf t
R3#show ip ospf int s0.31
Serial0.31 is up, line protocol is up
  Internet Address 172.12.123.3/24, Area 0
  Process ID 1, Router ID 3.3.3.3, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
< Note the timer values for point-to-point networks. >
  Hello due in 00:00:04
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

R3#show ip ospf neighbor

< There is no output from this command, so no neighbors exist. >

R3(config)#int s0.31

R3(config-subif)#ip ospf network non-broadcast

< "ip ospf network non-broadcast" changes the interface's OSPF network type. >

R3#show ip ospf int s0.31

Serial0.31 is up, line protocol is up

Internet Address 172.12.123.3/24, Area 0

Process ID 1, Router ID 3.3.3.3, Network Type NON_BROADCAST, Cost: 64

Transmit Delay is 1 sec, State DROTHER, Priority 0

Designated Router (ID) 1.1.1.1, Interface address 172.12.123.1

No backup designated router on this network

Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5

< By changing the network type, the timer values have changed as well.

Hello due in 00:00:21

Neighbor Count is 1, Adjacent neighbor count is 0

R3#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/DR	00:01:55	172.12.123.1	Serial0.31

After changing the network type on R3's point-to-point interface, the timers change and an adjacency with R1 successfully forms. The HELLO timer is actually the value that allows the adjacency to form; when the HELLO times match, the adjacency can form.

Get in the habit of running "show ip ospf interface" to check network types. This is the #1 reason that expected OSPF adjacencies do not form.

In the initial configuration, R1 was configured to become the Designated Router and will show two neighbors:

```
R1#show ip ospf neighbor
```

<i>Neighbor ID</i>	<i>Pri</i>	<i>State</i>	<i>Dead Time</i>	<i>Address</i>	<i>Interface</i>
3.3.3.3	0	FULL/DROTHER	00:01:57	172.12.123.3	Serial0
2.2.2.2	0	FULL/DROTHER	00:01:44	172.12.123.2	Serial0

Examining the output of “show ip ospf neighbor”:

Neighbor ID: By default, a router's OSPF ID is the highest IP address configured on a LOOPBACK interface. This can also be manually configured with the command “router-id” in router configuration mode, and in the real world, is usually set with that command instead of leaving the router-id selection up to the router.

Pri: Short for “Priority”, this is the OSPF priority of the interface on the remote end of the adjacency. Both were manually set to 0 in the initial configuration to prevent them from becoming the DR or BDR.

State: FULL refers to the state of the adjacency. DROTHER means that this router is neither the DR or BDR.

Dead Time: A decrementing timer that resets when a HELLO packet is received from the neighbor.

Address: The IP address of the neighbor.

Interface: The adjacency was created via this interface.

OSPF Router Types

OSPF categorizes routers into four different categories:

Internal Routers are routers whose interfaces are all in the same area.

Area Border Routers (ABR) will have at least one interface in Area 0, and connect other areas to Area 0.

Backbone Routers are routers with at least one interface in Area 0. All ABRs are backbone routers, but not all backbone routers are ABRs.

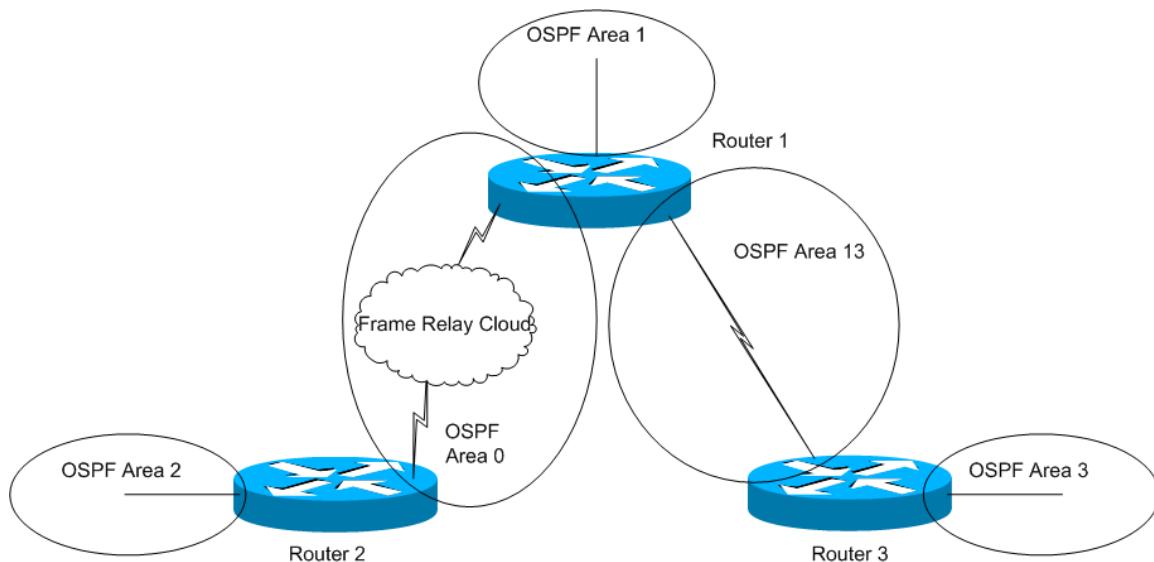
Autonomous System Border Routers take routes from other protocols and place them into the OSPF domain. This process is called **route redistribution**.

To view a router's OSPF router type, router ID, and the number of interfaces running in each area, run **show ip ospf**.

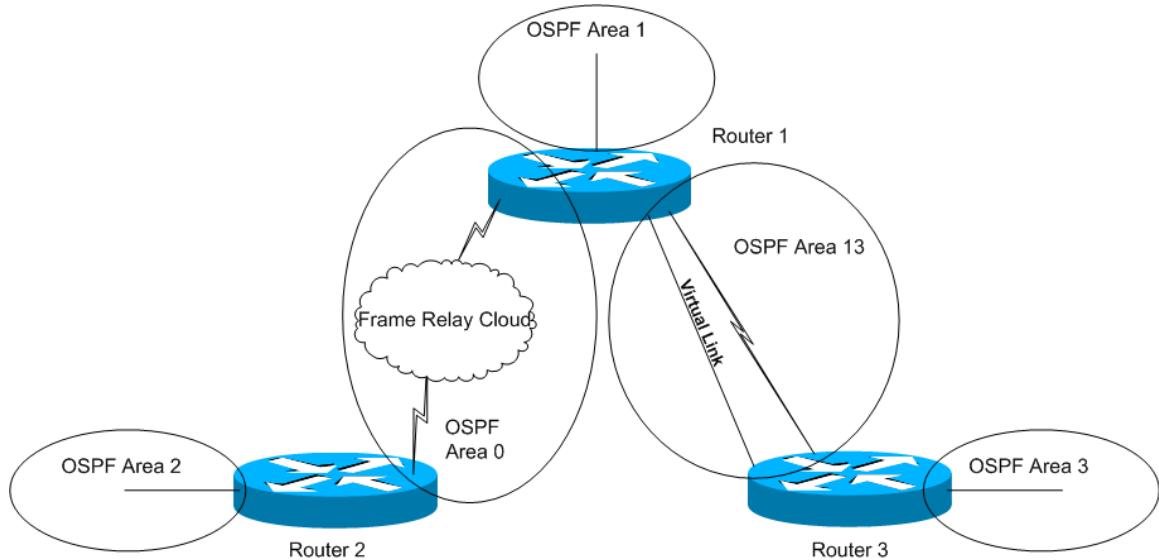
```
R1#show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Supports only single TOS(TOS0) routes
It is an area border router
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 3. 3 normal 0 stub 0 nssa
Area BACKBONE(0)
    Number of interfaces in this area is 2
    Area has no authentication
    SPF algorithm executed 10 times
Area 1
    Number of interfaces in this area is 1
    Area has no authentication
    SPF algorithm executed 2 times
Area 13
    Number of interfaces in this area is 1
    Area has no authentication
    SPF algorithm executed 6 times
```

Virtual Links

Every router in the OSPF domain must have a physical or logical connection to Area 0 to be a valid configuration. In the previous example, each router had a physical connection to Area 0. Consider the following example where R3 has no physical interface in Area 0:



This configuration is currently invalid, since R3 has no interface adjacent to Area 0. R3's neighbor R1 does have an interface in Area 0, so creating a **virtual link** over the transit area between R1 and R3 (Area 13) will allow R3 to have a logical connection to Area 0, making the configuration valid.



Configuring a virtual link across Area 13, the transit area between R1 and R3.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#area 13 virtual-link 3.3.3.3
< The virtual link is created using the remote router's OSPF ID. >
```

```
R3#conf t
R3(config)#router ospf 1
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone a
st be virtual-link but not found from 172.12.13.1, Serial1
```

< When configuring the second router for a virtual link, this message is common. It is not a configuration error; it is the first router looking for the virtual link on the second router. Once the virtual link command is completed on the second router, the message should no longer be received. If it is, there is an error in the virtual link configuration. >

```
R3(config-router)#area 13 virtual-link 1.1.1.1
```

```
R3#show ip ospf virtual-link
Virtual Link OSPF_VL0 to router 1.1.1.1 is up
```

< The virtual link is up, but an adjacency must also form over the link before it can be used. >

Run as demand circuit

DoNotAge LSA allowed.

Transit area 13, via interface Serial1, Cost of using 195

Transmit Delay is 1 sec, State POINT_TO_POINT,

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:05

Adjacency State FULL (Hello suppressed)

< An adjacency has been formed over the virtual link and the configuration is now valid. >

.

OSPF Stub Area Types

OSPF has several different kinds of **stub areas**. These areas may prohibit learning routes from other protocols, from other OSPF area, or an area may have only a default route used to reach all destinations. The different kinds of OSPF areas are out of the scope of the CCNA exam, but it is important to know of their existence. **Virtual links cannot be created over stub areas.**

The “basic” type of stub area does not allow routes learned via redistribution to be placed into the routing tables. The Area Border Router of a stub area will send an LSA advertising a default route that should be used by the stub routers to reach the destinations learned by redistribution. Routes injected into OSPF via redistribution are called **external routes**.

To configure an OSPF area as stub, use the command “area stub”, as shown.

```
R1#conf t  
R1(config)#router ospf 1  
R1(config-router)#network 172.12.123.0 0.0.0.255 area 0  
R1(config-router)#network 1.1.1.0 0.0.0.15 area 1  
R1(config-router)#area 1 stub
```

Totally stubby areas take the concept of stub areas one step farther. Not only is a default route used to reach external routes, but is also used to reach destinations outside the router’s OSPF area.

The addition of “no-summary” to the “area stub” command makes the area a totally stubby area.

```
R1#conf t  
R1(config)#router ospf 1  
R1(config-router)#area 1 stub no-summary
```

OSPF Authentication

An OSPF area can be authenticated with either clear-text or MD5 encryption. OSPF clear-text authentication is commonly referred to as "simple" authentication.

For clear-text authentication, the command **area x authentication** is run in router configuration mode, and the interface-level command **ip ospf authentication-key** followed by the password. The password and authentication type must match on all interfaces in the area, or adjacencies will not form.

Configuring OSPF “simple” authentication on Area 0.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#area 0 authentication
R1(config)#interface s0
R1(config-if)#ip ospf authentication-key CISCO

R2#conf t
R2(config)#router ospf 1
R2(config-router)#area 0 authentication
R2(config-router)#exit
R2(config-if)#interface serial0.123
R2(config-subif)#ip ospf authentication-key CISCO

R3#conf t
R3(config)#router ospf 1
R3(config-router)#area 0 authentication
R3(config)#interface s0.31
R3(config-subif)#ip ospf authentication-key CISCO
```

< The output of “show ip ospf” confirms that Area 0 is using simple password authentication. >

```
R1#show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Supports only single TOS(TOS0) routes
It is an area border router
Area BACKBONE(0)
    Number of interfaces in this area is 1
Area has simple password authentication
```

To configure MD5 authentication for the area, add **message-digest** to the **area authentication** command, and configure the interface with the **ip ospf message-digest-key** command.

Configuring MD5 authentication over Area 0.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#network 172.12.123.0 0.0.0.255 area 0
R1(config-router)#area 0 authentication message-digest
R1(config)#interface serial0
R1(config-if)#ip ospf message-digest-key 1 md5 CISCO
< The "1" following the message-digest-key command is the key number; this is followed by "md5", and then the password, in this case CISCO. >
```

```
R2#conf t
R2(config-subif)#router ospf 1
R2(config-router)#network 172.12.123.0 0.0.0.255 area 0
R2(config-router)#area 0 authentication message-digest
R2(config)#interface s0.123
R2(config-subif)#ip ospf message-digest-key 1 md5 CISCO
```

```
R3#conf t
R3(config-subif)#router ospf 1
R3(config-router)#network 172.12.123.0 0.0.0.255 area 0
R3(config-router)#area 0 authentication message-digest
R3(config-router)#interface s0.31
R3(config-subif)#ip ospf message-digest-key 1 md5 CISCO
```

```
R1#show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Area BACKBONE(0)
    Number of interfaces in this area is 1
    Area has message digest authentication
```

The partial output of "show ip ospf" verifies that Area 0, the backbone area, is running message digest authentication.

Regardless of which authentication type is selected, always run **show ip ospf neighbors** after configuring authentication to verify that all neighbor relationships are intact.

Configuring the OSPF Router ID

By default, the OSPF Router ID (RID) will be the numerically highest IP address of all loopback interfaces configured on the router. If no loopbacks exist, the numerically highest IP address of all physical interfaces will be used as the RID.

It's good network design to determine the RID manually, rather than leave the selection up to the router. To configure the RID, use the router configuration mode command **router-id**.

Changing R1's OSPF RID to 11.11.11.11.

```
R1#conf t  
R1(config)#router ospf 1  
R1(config-router)#router-id 11.11.11.11
```

R1#clear ip ospf processes

< If a change is made to an OSPF RID and a RID has been previously set, the RID will only change upon a router reboot or by clearing the OSPF processes. WARNING: Clearing the OSPF process will cause all existing adjacencies to be lost. They will be renegotiated with the new RID. >

```
R1#show ip ospf  
Routing Process "ospf 1" with ID 11.11.11.11
```

Troubleshooting OSPF: The ISDN link stays up after the BRI interfaces are added to an OSPF area.

If an ISDN link is in an OSPF area, the Hello packets will keep the link up.

Configuring OSPF Area 12 over the ISDN link brings the line up, and the Hello packets multicast to 224.0.0.5 will keep the line up.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#network 172.12.21.0 0.0.0.3 area 12
1d22h: %DIALER-6-BIND: Interface BRI0:1 bound to profile Dialer1
1d22h: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 8358662
< Upon adding the ISDN link to Area 12 on R1, the ISDN line dials R2. >
```

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#network 172.12.21.0 0.0.0.3 area 12
< The ISDN link is added to Area 12 on R2. >
```

```
R1#show dialer
```

```
BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=172.12.21.1, d=224.0.0.5)
Interface bound to profile Dialer1
Time until disconnect 111 secs
Current call connected 00:07:18
Connected to 8358662 (R2)
```

Hello packets in OSPF are multicast to 224.0.0.5. The current ISDN configuration sees all IP traffic as interesting traffic; each time an OSPF Hello leaves R1, the idle-timer is reset. The call will never time out.

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/ -	00:00:38	172.12.21.2	Dialer1
3.3.3.3	0	FULL/DROTHER	00:01:42	172.12.123.3	Serial0
2.2.2.2	0	FULL/DROTHER	00:01:53	172.12.123.2	Serial0

The adjacency has formed over the ISDN link. Note that the adjacency has a dead time listed.

OSPF will prevent Hellos from keeping up the ISDN link while keeping the adjacency through the use of an **OSPF demand circuit**. The link will come up when added to an OSPF area; however, once the adjacency has formed, Hellos will no longer be sent across the line, allowing the line to go down. The adjacency will still exist, as OSPF makes an assumption that the line will be available when needed – that is, when OSPF demands the circuit.

Configuring an OSPF demand circuit on R1's Dialer1 interface.

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#interface dialer1  
R1(config-if)#ip ospf demand-circuit  
< The "ip ospf demand-circuit" command is only needed on one side of the ISDN link. >
```

R1#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/		172.12.21.2	Dialer1
3.3.3.3	0	FULL/DROTHER	00:01:58	172.12.123.3	Serial0
2.2.2.2	0	FULL/DROTHER	00:01:38	172.12.123.2	Serial0

< The adjacency is still there, but a dead time is no longer listed. Hellos are no longer being sent over the ISDN link due to the **ip ospf demand-circuit** command. A lack of Hellos would usually bring an adjacency down, but here OSPF is assuming the link will be available when needed. >

R1#show dialer

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

Dialer1 - dialer type = DIALER PROFILE
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

< "show dialer" verifies that the line is down. An OSPF adjacency exists over the ISDN link, but the line is down, keeping costs down. >

Troubleshooting OSPF: Adjacencies will not form.

Different interface types default to different OSPF network types. A physical or multipoint Serial interface will default to an OSPF non-broadcast network type; a point-to-point interface will default to an OSPF point-to-point network type. If the network types are not the same on each end of an attempted adjacency, the adjacency may not form.

The value that matters most when OSPF adjacencies form or don't form **is actually the Hello timer**. The Hello timers vary between network types:

	Hello Time	Dead Time
Non-broadcast	30	120
Point-to-multipoint	30	120
Broadcast	10	40
Point-to-point	10	40

The different OSPF network types are covered thoroughly in the CCNP curriculum and exams. What is important for a true CCNA to know is that adjacencies **can** form between mismatched network types as long as the Hello timers are the same.

In The REAL World...

It's not good design to have adjacencies forming over mismatched OSPF network types. Unpredictable issues with OSPF routing have been seen in such a network in a lab environment. It **is** important to know why it **can** be done – because the Hello timers match.

Consider the original network topology in this chapter. R1 is running OSPF on its Serial0 physical interface; this interface will default to the **non-broadcast** OSPF network type, verified by **show ip ospf interface serial1**:

```
R1#show ip ospf interface serial0
Serial0 is up, line protocol is up
  Internet Address 172.12.123.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
```

R3 is connected to the Frame Relay cloud on interface Serial0.31, a point-to-point interface. Point-to-point interfaces default to the OSPF network type point-to-point, verified by **show ip ospf interface serial0.31**:

```
R3#show ip ospf interface serial0.31
Serial0.31 is up, line protocol is up
  Internet Address 172.12.123.3/24, Area 0
  Process ID 1, Router ID 3.3.3.3, Network Type POINT_TO_POINT, Cost: 64
```

Running **show ip ospf neighbor** shows that no adjacency exists over the R1 – R3 link:

```
R1#show ip ospf neighbor

Neighbor ID Pri State          Dead Time Address      Interface
N/A           0 ATTEMPT/DROTHER -   172.12.123.3 Serial0
2.2.2.2       0 FULL/DROTHER  00:01:53  172.12.123.2 Serial0

R3#show ip ospf neighbor
R3# <no output>
```

The most efficient manner to spot the problem with an adjacency is to run **debug ip ospf adj** on R1. Almost immediately, the issue is spotted:

Examining the output of “debug ip ospf adj” to diagnose an adjacency problem.

```
R1#debug ip ospf adj
OSPF adjacency events debugging is on
2d03h: OSPF: Rcv hello from 3.3.3.3 area 0 from Serial0 172.12.123.3
2d03h: OSPF: Mismatched hello parameters from 172.12.123.3
2d03h: Dead R 40 C 120, Hello R 10 C 30 Mask R 255.255.255.0 C 255.255.255.0
```

The OSPF Hello was received from R3 (with the Router ID 3.3.3.3), but then the problem is seen. Note that the debug output does not say anything about a network mismatch; the issue is with “Mismatched hello parameters”; in other words, Hello time. The R3 OSPF interface has a Hello time of 10 seconds, but the receiving interface on R1, Serial0, has a Hello time of 40 seconds. Until this issue is resolved, the adjacency cannot take place.

There are actually two ways to resolve this issue. One is to change R3's Serial0.31 OSPF Hello time to 30 seconds with the **ip ospf hello** command:

```
R3#conf t
R3(config)#int s0.31
R3(config-subif)#ip ospf hello 30

R1#debug ip ospf adj
OSPF adjacency events debugging is on
2d03h: OSPF: Rcv hello from 2.2.2.2 area 0 from Serial0 172.12.123.2
2d03h: OSPF: End of hello processing
2d03h: OSPF: Rcv hello from 3.3.3.3 area 0 from Serial0 172.12.123.3
2d03h: OSPF: 2 Way Communication to 3.3.3.3 on Serial0, state 2WAY
2d03h: OSPF: Neighbor change Event on interface Serial0
```

A Hello packet is again received from R3 (Router ID 3.3.3.3), but since the Hello times now match, the state of the connection goes almost immediately to 2-way.

```
R1#show ip ospf neighbor
Neighbor ID Pri      State        Dead Time   Address      Interface
3.3.3.3      0 FULL/DROTHER 00:01:57  172.12.123.3  Serial0
2.2.2.2      0 FULL/DROTHER 00:01:51  172.12.123.2  Serial0
```

The adjacency is now full.

The conventional method is to change the OSPF interface network type to match the remote end:

```
R3#conf t
R3(config)#int s0.31
R3(config-subif)#ip ospf network non-broadcast
```

Regardless of the method used to resolve the issue, the key is to know how to diagnose the problem with **show ip ospf interface** and **debug ip ospf adj**.

Link-State and OSPF Q&A

1. What is the major difference in the way link-state and distance-vector protocols build their routing tables?
 - A. **Link state protocols build a table from Link State advertisements, while distance-vector protocols exchange routing updates.**
 - B. **Link state protocols and distance-vector protocols exchange routing updates, but link state protocols understand VLSMs and distance-vector protocols don't.**
 - C. **Link state protocols send routing updates only to neighbors, where distance-vector protocols multicast or broadcast them.**
 - D. **Link state protocols use the Dijkstra algorithm, where distance-vector protocols use DUAL.**

ANSWER: A. Link-state routers do not exchange routing updates, as we saw RIP and IGRP do. Link-state protocols such as OSPF build their routing tables via the exchange of LSAs.

2. By default, what is OSPF's interval between Hello packets on an Ethernet network?

- A. **5 seconds.**
- B. **10 seconds.**
- C. **20 seconds.**
- D. **30 seconds.**

ANSWER: B.

3. By default, what is OSPF's interval between Hello packets on a non-broadcast link, such as a Serial interface?

- A. **5 seconds.**
- B. **10 seconds.**
- C. **20 seconds.**
- D. **30 seconds.**

ANSWER: D.

4. By default, the OSPF “dead time” is how many times the “hello time”?

- A. Two.
- B. Three.
- C. Four.
- D. Five.

ANSWER: C. The OSPF dead-time is four times the hello interval.

5. OSPF uses what algorithm to prevent routing loops?

- A. DUAL.
- B. CDP.
- C. Dijkstra Algorithm.
- D. Hamilton Algorithm.

ANSWER: C. The Dijkstra Algorithm recalculates network changes so quickly after network changes that routing loops don't have the opportunity to form.

6. When configuring OSPF, how does Split Horizon help to prevent loops?

- A. It doesn't.
- B. Routes will not be advertised out the same interface on which they were received.
- C. Routers will be marked with an unreachable metric if the advertising router determines it can no longer be reached.
- D. The loopback interface will receive updates, but will not receive them.

ANSWER: A. Split horizon is used with distance-vector protocols. Remember that route updates are not exchanged in OSPF, so you don't have to worry about them going out the same interface they were received on – because they're not being received, period.

7. What term is used for the logical segments of an OSPF network?

- A. Autonomous System.
- B. Area.
- C. Loopback Segment.

D. Partitions

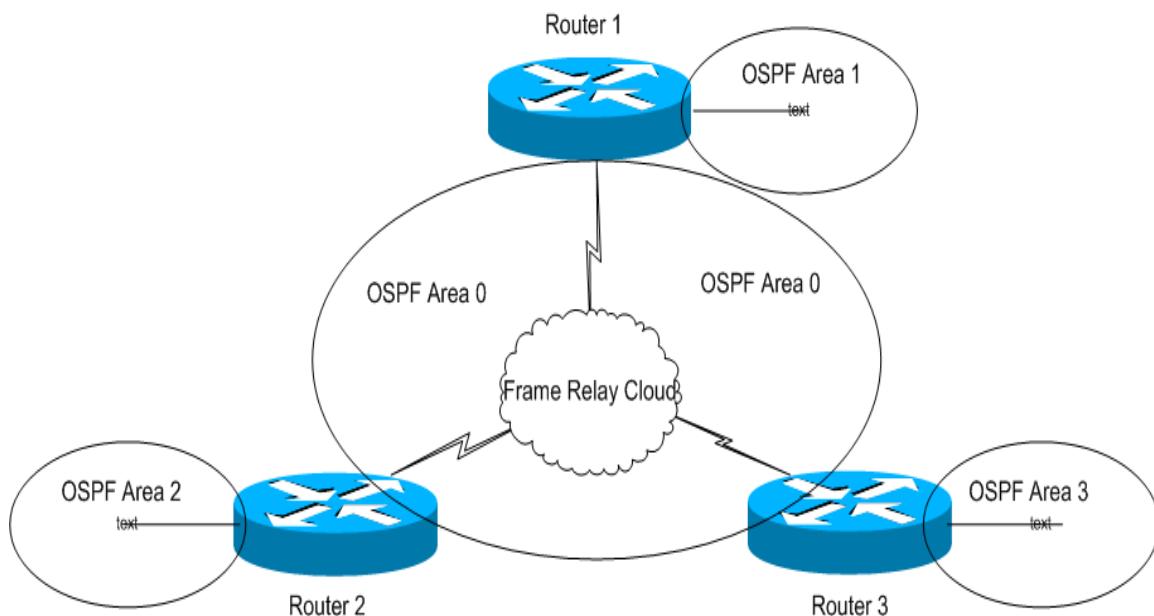
ANSWER: B. An OSPF domain can be segmented into logical sections called "areas".

8. What number is associated with the "backbone" of an OSPF network?

- A. Zero.
- B. 1
- C. 2
- D. 3
- E. 4

ANSWER: A. Area 0 is the backbone area of an OSPF network.

Consider the following diagram:



9. Which of the following statements is true? (Choose two.)

- A. There is a need for a virtual link in this network.
- B. Router 1 must be the Designated Router for this network.
- C. Router 2 or 3 can be the Backup Designated Router for this network.
- D. Neither Router 2 nor Router 3 can be the Designated Router or Backup Designated Router for this network.

ANSWER: B, D. Care must be taken when running OSPF on a hub-and-spoke network. The hub, Router 1, must be the Designated Router, and there can be no Backup Designated Router. If one of the two spoke routers is the BDR, and the DR goes down, the BDR will then become the DR. OSPF will not function correctly over a hub-and-spoke network if a spoke becomes the Designated Router.

10. By default, what is a router's OSPF Router ID?

- A. **The numerically highest loopback address.**
- B. **The numerically highest IP address on the router, regardless of interface.**
- C. **The numerically lowest loopback address.**
- D. **The numerically lowest IP address on the router, regardless of interface.**

ANSWER: A. The numerically highest loopback address is the OSPF Router ID, often referred to as the "RID". If there is no loopback address on the router, the highest IP address on a physical interface is then used.

11. On what OSPF router type are routes from other protocols redistributed into the OSPF network?

- A. **Internal router**
- B. **External router**
- C. **ABR**
- D. **ASBR**
- E. **Stub router**

Answer: D. An ASBR, or "Autonomous System Border Router", performs redistribution between another protocol and OSPF.

12. What defines an OSPF router as being an ABR?

- A. **Having at least one interface in Area 0.**
- B. **Having at least one interface in every area in the OSPF network.**
- C. **Having all interfaces in a single area, Area 0 or not.**
- D. **Having all interfaces in Area 0.**

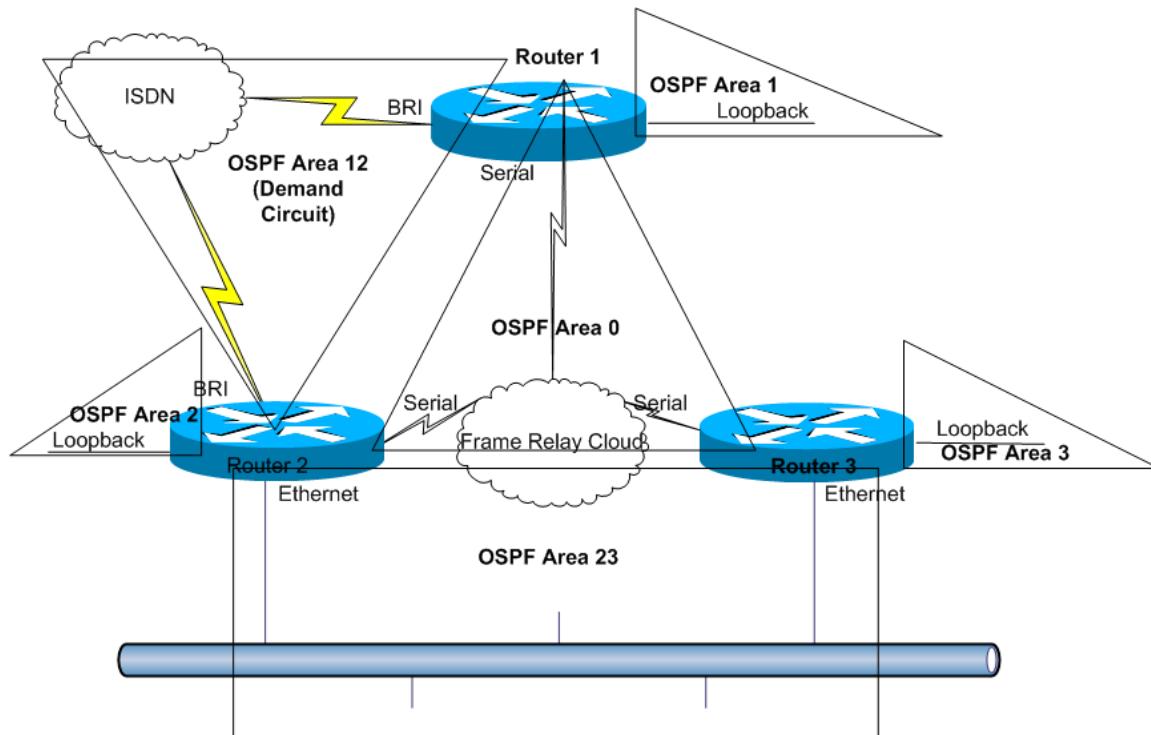
ANSWER: A. An ABR, or Area Border Router, has at least one interface in Area 0.

Notes

OSPF Lab: Configuring OSPF areas, stub areas, ISDN demand circuit, and clear-text and MD5 authentication.

Remove any existing routing protocol configuration, including interface and subinterface commands.

This is the OSPF network you will build:



Configure OSPF Area 0 on each router interface connected to the Frame Relay cloud with the **router ospf 1** and **network** commands. Run **show ip ospf interface** on each router to see what OSPF network type the interfaces are running.

Configuring OSPF on the Frame Relay cloud interfaces on R1, R2, and R3.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#network 172.12.123.0 0.0.0.255 area 0

R2#conf t
R2(config)#router ospf 1
R2(config-router)#network 172.12.123.0 0.0.0.255 area 0

R3#conf t
R3(config)#router ospf 1
R3(config-router)#network 172.12.123.0 0.0.0.255 area 0
```

Examining the partial output of “show ip ospf interface”

R1#**show ip ospf interface serial0**

Serial0 is up, line protocol is up

Internet Address 172.12.123.1/24, Area 0

Process ID 1, Router ID 1.1.1.1, **Network Type NON_BROADCAST**, Cost: 64

R2#**show ip ospf interface serial0.123**

Serial0.123 is up, line protocol is up

Internet Address 172.12.123.2/24, Area 0

Process ID 1, Router ID 2.2.2.2, **Network Type NON_BROADCAST**, Cost: 64

R3#**show ip ospf interface serial0.31**

Serial0.31 is up, line protocol is up

Internet Address 172.12.123.3/24, Area 0

Process ID 1, Router ID 3.3.3.3, **Network Type POINT_TO_POINT**, Cost: 64

R3’s point-to-point interface is defaulting to OSPF network type point-to-point. The timers will be different between R3 and R1, requiring that the network type be changed before an adjacency can occur.

This is a hub-and-spoke OSPF network, requiring that the hub router, R1, be the Designated Router. Additionally, since all three interfaces will be OSPF network type non-broadcast after changing R3, “neighbor” statements will need to be configured on the hub router.

Change R3’s serial 0.31 interface to OSPF network type non-broadcast with the **ip ospf network** interface-level command. Prevent R2 and R3 from possibly becoming the Designated Router by configuring **ip ospf priority 0** on the interfaces connected to the Frame Relay cloud.

```
R3#conf t  
R3(config)#int s0.31  
R3(config-subif)#ip ospf network non-broadcast  
R3(config-subif)#ip ospf priority 0
```

```
R2#conf t  
R2(config)#int s0.123  
R2(config-subif)#ip ospf priority 0
```

Allow R1 to discover its OSPF neighbors over the OSPF nonbroadcast network with two **neighbor** commands, naming the remote Frame Relay cloud neighbors. Run **show ip ospf neighbor** on R1 to verify adjacencies.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#neighbor 172.12.123.2
R1(config-router)#neighbor 172.12.123.3

R1#show ip ospf neighbor

Neighbor ID      Pri  State        Dead Time   Address          Interface
3.3.3.3           0    FULL/DROTHER  00:01:57   172.12.123.3  Serial0
2.2.2.2           0    FULL/DROTHER  00:01:57   172.12.123.2  Serial0
```

Prevent the Router ID (RID) from changing in the future by hard-coding it on each router with the **router-id** command.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#router-id 1.1.1.1

R2#conf t
R2(config)#router ospf 1
R2(config-router)#router-id 2.2.2.2

R3#conf t
R3(config)#router ospf 1
R3(config-router)#router-id 3.3.3.3
```

Add R1's loopback address to Area 1, R2's loopback to Area 2, and R3's loopback to Area 3. Use a wildcard mask of **0.0.0.0** so that only the loopback interface will be part of the respective area.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#network 1.1.1.1 0.0.0.0 area 1

R2#conf t
R2(config)#router ospf 1
R2(config-router)#network 2.2.2.2 0.0.0.0 area 2

C R3#conf t
W R3(config)#router ospf 1
C R3(config-router)#network 3.3.3.3 0.0.0.0 area 3
```

On R1, run **show ip route ospf**. A route to both R2's and R3's loopback should be present. Ping both interfaces to verify connectivity.

Examining R1's OSPF routing table.

```
R1#show ip route ospf
 2.0.0.0/32 is subnetted, 1 subnets
O IA  2.2.2.2 [110/65] via 172.12.123.2, 00:02:53, Serial0
  3.0.0.0/32 is subnetted, 1 subnets
O IA  3.3.3.3 [110/65] via 172.12.123.3, 00:02:53, Serial0
```

Note the “O IA” on the far left-hand side of the command output. The “O” indicates that this is an OSPF route; the “IA” means it is an InterArea route, or a route to a destination in another area.

```
R1#ping 2.2.2.2
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/68 ms
```

```
R1#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/68 ms
```

Run **show ip route ospf** on R2. Routes to the loopbacks of R1 and R3 should be present. Ping both loopbacks to verify connectivity.

```
R2#show ip route ospf
 1.0.0.0/32 is subnetted, 1 subnets
O IA  1.1.1.1 [110/65] via 172.12.123.1, 00:10:35, Serial0.123
  3.0.0.0/32 is subnetted, 1 subnets
O IA  3.3.3.3 [110/65] via 172.12.123.3, 00:10:35, Serial0.123
R2#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/68/68 ms
R2#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/133/144 ms
```

Run **show ip route ospf** on R3. Routes to the loopbacks of R1 and R2 should be present. Ping both loopbacks to verify connectivity.

```
R3#show ip route ospf
  1.0.0.0/32 is subnetted, 1 subnets
O IA  1.1.1.1 [110/65] via 172.12.123.1, 00:14:52, Serial0.31
    2.0.0.0/32 is subnetted, 1 subnets
O IA  2.2.2.2 [110/65] via 172.12.123.2, 00:14:52, Serial0.31
R3#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms
R3#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/133/144 ms
```

Configure the Ethernet segment connecting R2 and R3 as Area 23. Area 23 will be made a stub area. Use the **area stub** command on R3, but not R2. Run **show ip ospf neighbor** to verify the adjacency.

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#network 172.23.23.0 0.0.0.31 area 23

R3#conf t
R3(config)#router ospf 1
R3(config-router)#network 172.23.23.0 0.0.0.31 area 23
R3(config-router)#area 23 stub

R3#show ip ospf neighbor

Neighbor ID      Pri  State            Dead Time   Address      Interface
  1.1.1.1          1    FULL/DR        00:01:32   172.12.123.1  Serial0.31
```

Show ip ospf neighbor indicates that an adjacency has not started to form between R2 and R3. Diagnose the problem by running **debug ip ospf adj** on R3.

```
R3#debug ip ospf adj
OSPF adjacency events debugging is on
OSPF: Hello from 172.23.23.2 with mismatched Stub/Transit area option bit
```

The issue is quickly spotted with the debug command. The Hello packet is coming in from 172.23.23.2, but the Stub option bit is mismatched. **For a stub area to form, all routers must agree that the area is a stub. The command “area stub” must be configured on all routers with an interface in that area.**

On R2, configure **area 23 stub** in router configuration mode. On R3, run **debug ip ospf adj** and **show ip ospf neighbor** to verify the adjacency.

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#area 23 stub

R3# debug ip ospf adj
OSPF: 2 Way Communication to 2.2.2.2 on Ethernet0, state 2WAY
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 0.0.0.0
OSPF: Elect DR 3.3.3.3
    DR: 3.3.3.3 (Id) BDR: none
OSPF: Build router LSA for area 23, router ID 3.3.3.3
OSPF: Rcv DBD from 2.2.2.2 on Ethernet0 seq 0xC36 opt 0x0 flag 0x7 len 32 state EXSTART
OSPF: Rcv DBD from 2.2.2.2 on Ethernet0 seq 0x324 opt 0x0 flag 0x2 len 292 state EXSTART
< some content removed for clarity >
OSPF: Exchange Done with 2.2.2.2 on Ethernet0
OSPF: Synchronized with 2.2.2.2 on Ethernet0, state FULL
OSPF: Build router LSA for area 23, router ID 3.3.3.3
OSPF: Build network LSA for Ethernet0, router ID 3.3.3.3
R3#undbg all
All possible debugging has been turned off

R3#show ip ospf nei
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/BDR	00:00:35	172.23.23.2	Ethernet0
1.1.1.1	1	FULL/DR	00:01:59	172.12.123.1	Serial0/0.31

After configuring R2 with the **area stub** command, the adjacency begins to form. In the few seconds it takes to go back to R3, the adjacency is already in 2-way. The default network type for Ethernet interfaces is **broadcast**, which requires the election of a DR and BDR. The election is seen in the highlighted portion of the debug.

Also highlighted are the ExStart and Exchange state. The Exchange state then finishes, synchronization occurs with 2.2.2.2, the adjacency state moves to Full, and 2.2.2.2 shows as a neighbor of R3.

The OSPF adjacency states:

DOWN: Initial state of the adjacency. The router has not yet received any information from the potential neighbor.

ATTEMPT: Remote router has not sent the local router any information, but the local router is trying to contact it via unicast Hellos. (Unicast Hellos are sent with the **neighbor** command.)

INIT: A packet has been received from the remote router. The packet does not include the local router's Router ID.

2-WAY: Bidirectional communication is occurring, and the packets include the other router's Router ID.

EXSTART: The actual beginning of the adjacency. One router becomes the Master and the other the Slave. The Master/Slave relationship does not relate to the DR / BDR election.

EXCHANGE: The routers begin exchanging Database Description Packets, which describe its Link State Database.

LOADING: The routers explicitly request Link State Advertisements from its neighbor that may have been corrupted or missed during Exchange.

FULL: The adjacency is complete.

On R3, run **show ip ospf interface** to compare the characteristics of the Serial and Ethernet interfaces running OSPF.

```
R3#show ip ospf interface
Ethernet0 is up, line protocol is up
Internet Address 172.23.23.3/27, Area 23
Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 3.3.3.3, Interface address 172.23.23.3
Backup Designated router (ID) 2.2.2.2, Interface address 172.23.23.2
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2 (Backup Designated Router)
Suppress hello for 0 neighbor(s)
Serial0.31 is up, line protocol is up
Internet Address 172.12.123.3/24, Area 0
Process ID 1, Router ID 3.3.3.3, Network Type NON_BROADCAST, Cost: 64
Transmit Delay is 1 sec, State DROTHER, Priority 0
Designated Router (ID) 1.1.1.1, Interface address 172.12.123.1
No backup designated router on this network
Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
Hello due in 00:00:17
Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1 (Designated Router)
Suppress hello for 0 neighbor(s)
```

The default OSPF network type of an Ethernet interface is Broadcast, as shown in this output. Note the timer differences between the Broadcast and non-broadcast network types.

On R3, run **show ip ospf**. Area 23 will be shown as a stub area.

Examining partial output of “show ip ospf”.

R3#**show ip ospf**

Routing Process "ospf 1" with ID 3.3.3.3

Supports only single TOS(TOS0) routes

It is an area border router

Number of areas in this router is 3. 2 normal 1 stub 0 nssa

Area BACKBONE(0)

Number of interfaces in this area is 1

Area has no authentication

Area 3

Number of interfaces in this area is 1

Area has no authentication

Area 23

Number of interfaces in this area is 1

It is a stub area

generates stub default route with cost 1

Area has no authentication

From R1, ping R2's and R3's Ethernet interfaces.

R1#**ping 172.23.23.2**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.23.23.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms

R1#**ping 172.23.23.3**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.23.23.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms

Place the ISDN link into Area 12. Run **show ip ospf neighbor** to verify adjacency.

```
R1(config)#router ospf 1
R1(config-router)#network 172.12.21.0 0.0.0.3 area 12
2d21h: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
2d21h: %DIALER-6-BIND: Interface BRI0:1 bound to profile Dialer1
2d21h: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 8358662
2d21h: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1,
changed state to up
```

< Note that the ISDN link comes up immediately after enabling OSPF on it.
Recall that in an earlier lab, we made all IP traffic interesting traffic.
What is the destination address of the packets that brought the line up when
the ISDN link was placed into OSPF Area 12? >

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#network 172.12.21.0 0.0.0.3 area 12
```

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/ -	00:00:01	172.12.21.2	Dialer1
3.3.3.3	0	FULL/DROTHER	00:01:40	172.12.123.3	Serial0
2.2.2.2	0	FULL/DROTHER	00:01:31	172.12.123.2	Serial0

From R3, ping both ISDN interfaces to verify connectivity. Both pings should be successful.

Run **show dialer** on R1. Wait three minutes, and run it a second time.

Examining partial output of “show dialer” on R1, taken three minutes apart.

R1#**show dialer**

BRI0 - dialer type = ISDN

Dial String	Successes	Failures	Last called	Last status
0 incoming call(s)	have been screened.			
0 incoming call(s)		rejected for callback.		

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is data link layer up

Dial reason: ip (s=172.12.21.1, d=224.0.0.5)

Interface bound to profile Dialer1

Time until disconnect 116 secs

Current call connected 00:01:03

Connected to 8358662 (R2)

R1#**show dialer**

BRI0 - dialer type = ISDN

Dial String	Successes	Failures	Last called	Last status
0 incoming call(s)	have been screened.			
0 incoming call(s)		rejected for callback.		

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is data link layer up

Dial reason: ip (s=172.12.21.1, d=224.0.0.5)

Interface bound to profile Dialer1

Time until disconnect 115 secs

Current call connected 00:04:04

Connected to 8358662 (R2)

The “dial reason” has a destination of 224.0.0.5, the OSPF multicast address used to send Hellos. Three minutes later, the line is still up, and the idle-timer is resetting time after time, since the “time until disconnect” has not ticked down to zero. The OSPF Hellos are keeping the line up, and will continue to do so indefinitely.

Configure R1 to suppress the sending of Hellos over the ISDN link, but to keep the adjacency, with the **ip ospf demand-circuit** command. The line will go down after configuring this command. Run **show ip ospf neighbor** to verify the adjacency is up, and **show dialer** to verify the line is not in use.

Examining the behavior of the “ip ospf demand-circuit” command.

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#int dialer1

R1(config-if)#ip ospf demand-circuit

< The “ip ospf demand-circuit” command is configured. The link will not come down immediately, but OSPF Hello packets will no longer reset the idle-timeout. >

2d22h: %DIALER-6-UNBIND: Interface BRI0:1 unbound from profile Dialer1

2d22h: %ISDN-6-DISCONNECT: Interface BRI0:1 disconnected from 8358662 R2, calllasted 799 seconds

R1#

2d22h: %LINK-3-UPDOWN: Interface BRI0:1, changed state to down

2d22h: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to down

< OSPF Hellos are no longer keeping the line up, and the connection times out. >

R1#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/ -	-	172.12.21.2	Dialer1
3.3.3.3	0	FULL/DROTHER	00:01:36	172.12.123.3	Serial0
2.2.2.2	0	FULL/DROTHER	00:01:56	172.12.123.2	Serial0

< OSPF Hellos are no longer sent across the link , but the adjacency stays in Full mode. OSPF is assuming the circuit will be available when demanded – “demand circuit”. >

R1#show dialer

BRI0 - dialer type = ISDN

Dial String Successes Failures Last called Last status

0 incoming call(s) have been screened.

0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN

Idle timer (120 secs), Fast idle timer (20 secs)

Wait for carrier (30 secs), Re-enable (15 secs)

Dialer state is idle

“Show dialer” verifies the line is down. The demand circuit has been configured correctly. Note that “ip ospf demand-circuit” only needs to be configured on one side of the ISDN connection.

OSPF clear-text authentication will now be configured on Area 23. On both R2 and R3, configure **area 23 authentication** under the OSPF process. On the Ethernet interfaces, configure **ip ospf authentication-key CCNA**.

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#area 23 authentication
R2(config)#interface ethernet0
R2(config-if)#ip ospf authentication-key CCNA

R3#conf t
R3(config)#router ospf 1
R3(config-router)#area 23 authentication
R3(config)#interface ethernet0
R3(config-if)#ip ospf authentication-key CCNA
```

On R3, run **show ip ospf neighbor** and **show ip ospf** to verify the neighbor relationship still exists and that Area 23 shows as running simple authentication.

Examining partial output of “show ip ospf”.

```
R3#show ip ospf
Routing Process "ospf 1" with ID 3.3.3.3
Supports only single TOS(TOS0) routes
It is an area border router

Area 23
Number of interfaces in this area is 1
It is a stub area
generates stub default route with cost 1
Area has simple password authentication
SPF algorithm executed 8 times
Area ranges are
Link State Update Interval is 00:30:00 and due in 00:20:33
Link State Age Interval is 00:20:00 and due in 00:08:33
Number of DCbitless LSA 0
```

```
R3#show ip ospf neighbor
Neighbor ID      Pri  State          Dead Time    Address        Interface
2.2.2.2          1    FULL/BDR      00:00:38    172.23.23.2  Ethernet0
1.1.1.1          1    FULL/DR       00:01:49    172.12.123.1  Serial0.31
```

The neighbor relationship with R2 is intact, and simple password authentication is taking place.

MD5 authentication will now be configured on Area 0. Under the OSPF process on all R1, R2, and R3, configure **area 0 authentication message-digest**, and on the interface connecting to the Frame Relay cloud, configure **ip ospf message-digest-key 1 md5 CISCO**.

```
R1#conf t
R1(config)#router ospf 1
R1(config-router)#area 0 authentication message-digest
R1(config-router)#interface serial0
R1(config-if)#ip ospf authentication message-digest-key 1 md5 CISCO

R2#conf t
R2(config)#router ospf 1
R2(config-router)#area 0 authentication message-digest
R2(config-router)#interface serial0.123
R2(config-subif)#ip ospf authentication message-digest-key 1 md5 CISCO

R3#conf t
R3(config)#router ospf 1
R3(config-router)#area 0 authentication message-digest
R3(config-router)#interface serial 0.31
R3(config-subif)#ip ospf authentication message-digest-key 1 md5 CISCO
```

On R1, run **show ip ospf neighbor** and **show ip ospf** to verify the neighbor relationships and that Area 0 is running MD5 authentication.

```
R1#show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Supports only single TOS(TOS0) routes
It is an area border router
Area BACKBONE(0)
Number of interfaces in this area is 1
Area has message digest authentication

R1#show ip ospf neighbor

Neighbor ID  Pri  State      Dead Time   Address       Interface
2.2.2.2      1    FULL/ -     -          172.12.21.2  Dialer1
3.3.3.3      0    FULL/DROTHER 00:01:48  172.12.123.3  Serial0
2.2.2.2      0    FULL/DROTHER 00:01:39  172.12.123.2  Serial0
```

Section Nine: EIGRP

Your knowledge of distance vector and link-state protocols will now come in handy, because EIGRP has characteristics of both. You must know how to properly use the variance command, how to see the values of all possible EIGRP routes, and how EIGRP handles equal-cost and unequal-cost load balancing.

*Chris Bryant
CCIE #12933*

Commands Introduced In This Chapter And Labs:

Debug ip eigrp neighbor – Detects issues preventing adjacency with a neighbor.

Show ip eigrp neighbor – Displays EIGRP neighbors, how long the adjacency has been present, and what interface the adjacency was formed on.

Show ip eigrp topology – Displays the EIGRP topology table, including Successors (primary routes) and Feasible Successors (backup routes).

Variance – Used to allow EIGRP to perform unequal-cost load sharing.

Why EIGRP Is Considered A “Hybrid” Protocol

Link-state protocols (OSPF) and distance-vector protocols (RIPv1, RIPv2, IGRP) have clear-cut differences in the way the best routes are determined and what is actually exchanged between routers. There is a third category of Cisco routing protocol, the **hybrid**. Much as a hybrid plant has characteristics of more than one plant, a hybrid routing protocol has characteristics of both link-state and distance-vector protocols. The hybrid protocol is Enhanced Interior Gateway Routing Protocol – **EIGRP**.

EIGRP acts like a distance-vector protocol in that they initially exchange full routing tables. EIGRP also uses a calculation involving

bandwidth and delay to arrive at the metric for a route, just like distance-vector protocol IGRP.

However, EIGRP uses Hello packets (sent to multicast address 224.0.0.10) to keep neighbor relationships alive, much like a link-state protocol. The Reliable Transport Protocol (RTP) is used to handle the transport of messages between EIGRP-enabled routers. EIGRP also acts like a link-state protocol in that when network topology changes occur, updates containing only the change are sent, rather than another full routing table.

Hello Packets and RTP: The Heartbeat Of The EIGRP Network

Like IGRP, EIGRP uses *autonomous systems* to identify routers that will belong to the same logical group. EIGRP routers that exist in separate autonomous systems will not exchange routes by default.

The exchange of routes between different protocols, *route redistribution*, almost always has to be explicitly configured. There is one notable exception: IGRP and EIGRP routers **will** automatically exchange routes if the IGRP AS and EIGRP AS numbers are the same.

For an EIGRP neighbor relationship to be established, the routers must receive *Hello packets* from the neighbor, the Autonomous System number must match, and the *metric weights* must match. (Do not confuse the k metrics with the route metrics; they are two totally separate metrics.) The metric weights refer to the weight, or importance, EIGRP will give to the bandwidth, delay, load, and reliability metrics. Changing the metric weights is covered in the CCNP curriculum; for now, know that these metric weights must be the same on each router or the neighbor relationship will not be established.

As with OSPF, once the neighbor relationship is present, it is the Hello packets that keep it alive. If the Hellos are no longer received by a router, the neighbor relationship will eventually be terminated.

Loop Avoidance , the Successor, and the Feasible Successor

EIGRP actually keep three tables; the *route table*, where the best route to each destination is kept; the *topology table*, where all feasible routes are kept; and the *neighbor table*, where the EIGRP neighbors and information about them are kept.

As an EIGRP-enabled router learns about the network, the router will put the best route to a given destination in its routing table, as any distance-vector protocol would. EIGRP will keep any other routes that could be used without looping in the topology table, and will use one of those routes if the best route fails. EIGRP actually calculates backup routes to all destinations before a failure occurs, making convergence after a failure much quicker.

With EIGRP, the best route is referred to as the **Successor**. Any feasible, or possible, alternate route is referred to as the **Feasible Successor**. The decision process for whether a route can become a Feasible Successor can be summed up in one question:

The EIGRP Feasible Successor Question:

The router asks itself, “Is the neighboring router’s metric for this route lower than my metric?”

If so, no loop is present, and that route is a Feasible Successor.

If not, a loop may be present, and that route cannot be a Feasible Successor.

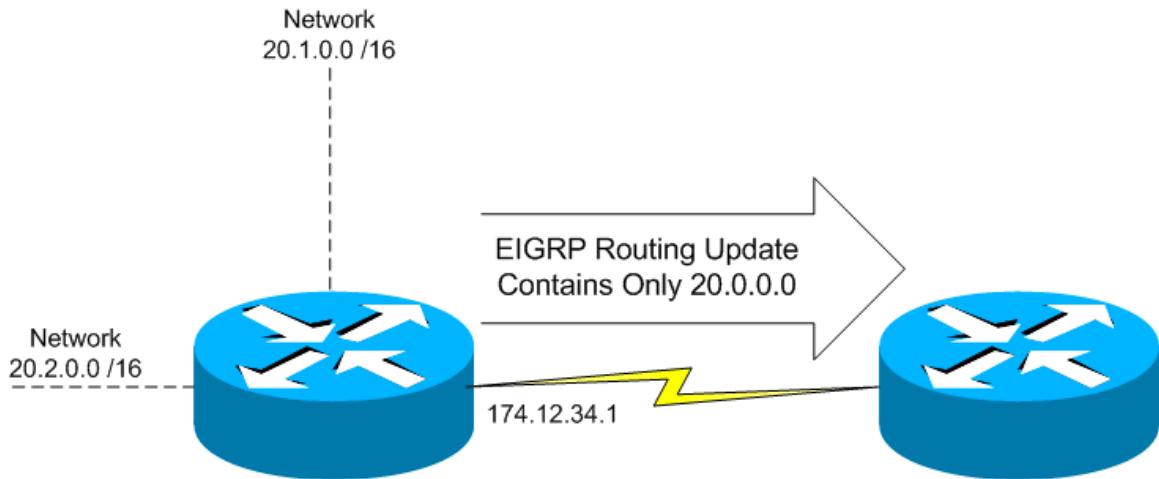
What if there is no Feasible Successor?

EIGRP uses the Diffusing Update Algorithm (DUAL) to issue queries to neighbors for a loop-free route to the destination. If the routers receiving the DUAL queries do not have a route, those routers will also send DUAL queries. This process continues until a route is found and the original router is informed of the route.

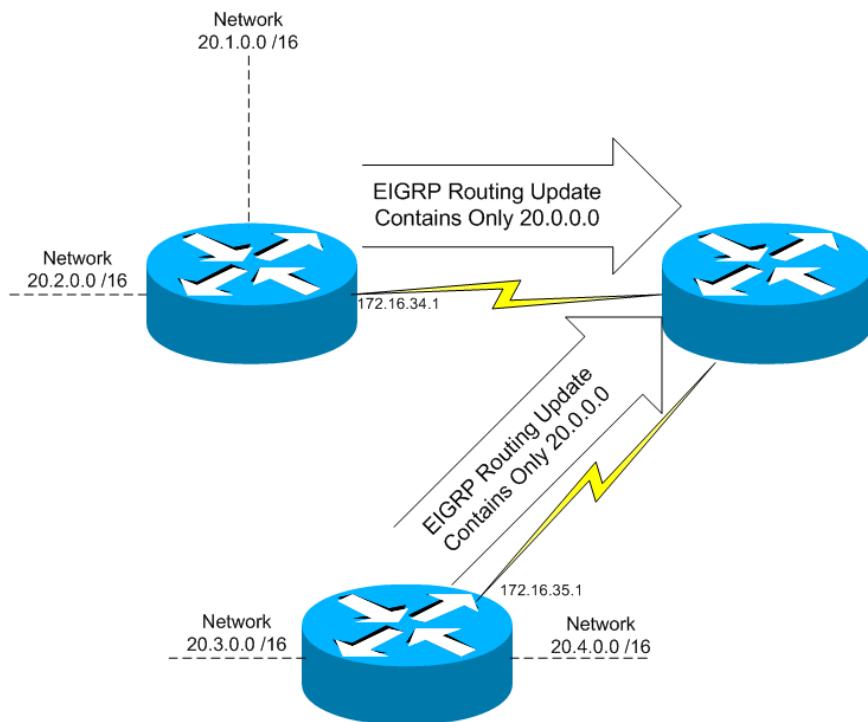
EIGRP’s Default Network Summarization

EIGRP performs *autosummarization*, which is the act of summarizing network routes when those routes are sent across a network boundary; that is, when they are advertised via an interface that is not part of the network being summarized.

Consider this router with interfaces in networks 20.1.0.0 /16, 20.2.0.0 /16, and 172.16.34.0. When the router advertises these routes via EIGRP, the default behavior is that networks 20.1.0.0 and 20.2.0.0 will be autosummarized to 20.0.0.0/8 when that route crosses the 172.16.34.0 boundary.



Seems like a good idea, right? Sometimes it is. In the above example, there are only two subnets of network 20.0.0.0, so it's a good thing to send an automatic summary route to the neighboring router. The smaller the route table, the better we like it... *as long as it's an accurate table*. In this small network, both subnets of 20.0.0.0 are reached by the receiving router by the same path – the subnets are *contiguous*. Consider the next example.

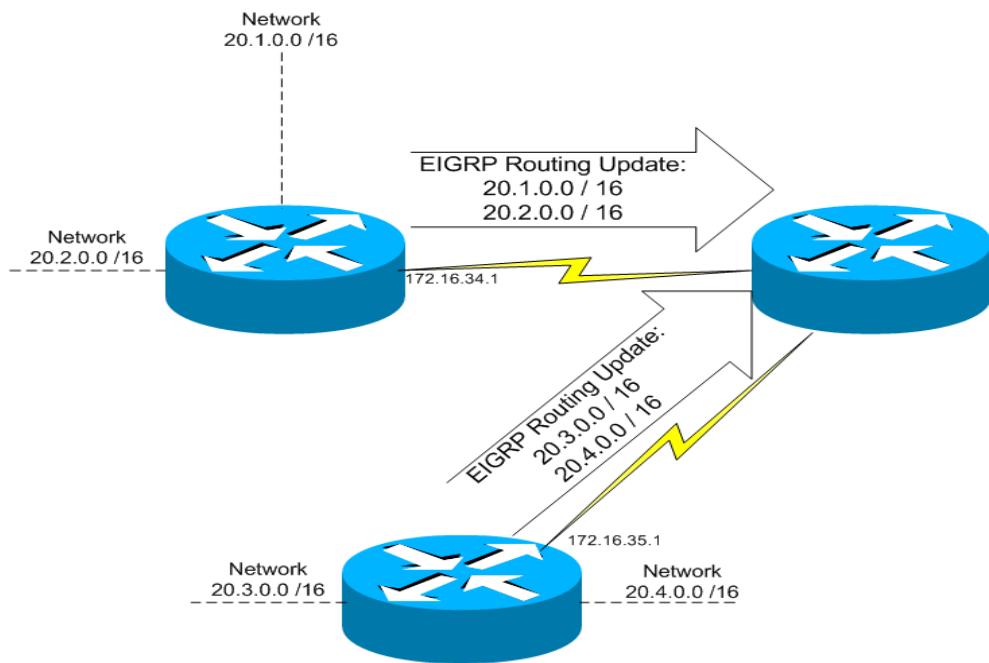


Network 20.0.0.0 is now considered *discontiguous* – there is no single path to all subnets of the major network number. This causes problems for routing protocols such as RIPv1 and IGRP, since subnet

mask information is not carried with those routing protocol updates. EIGRP carries subnet mask information, but the default autosummarization causes trouble with this network. The receiving router is getting a route for the classful network 20.0.0.0 from two different routers.

One of two things is going to happen, and both of them are bad. If the metric for the routes to 20.0.0.0 happen to be equal, equal-cost load balancing for the classful network 20.0.0.0 will be performed, ensuring that at least half of the packets destined for any particular subnet of 20.0.0.0 will be going to the wrong router. If the metric is unequal, a single route for the classful network 20.0.0.0 will be placed into the routing table. All packets for the four subnets will go to the same router, and two of the four subnets will never receive any packets.

This default behavior is easily removed with the **no auto-summary** command. When both of the sending routers add this command to their EIGRP configuration, the routes will no longer be summarized at the network boundary. The receiving router will now receive more detailed routes from both its EIGRP neighbors, and will place a route for each subnet into its table.



How EIGRP handles multiple equal-cost and unequal-cost routes to the same destination.

EIGRP and IGRP handle load-balancing in the same fashion. Both will enter four equal-cost routes to the same destination into the routing table by default, and this value can be changed with the **maximum-paths** command to a minimum of one and a maximum of six. (Setting **maximum-paths** to 1 will prevent load-balancing.)

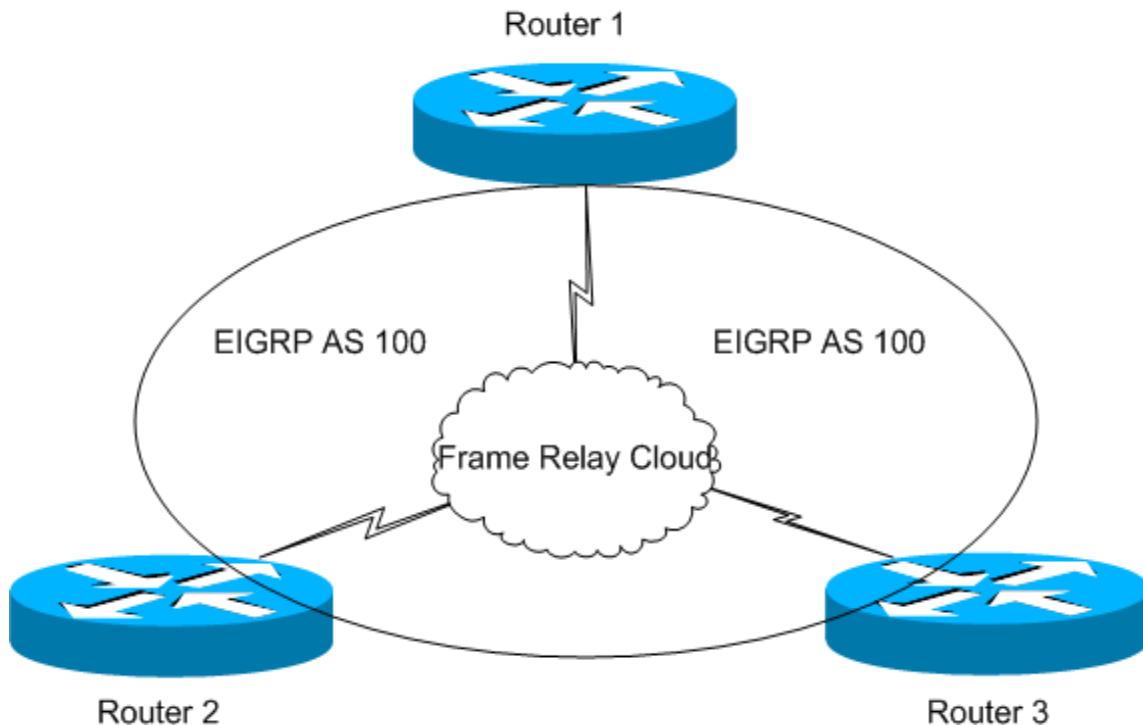
Unequal-cost load-balancing can be configured with the **variance** command with both EIGRP and IGRP. EIGRP contains a table showing all feasible successors, allowing for easier configuration of unequal-cost load-balancing. (Remember that IGRP requires a debug to see the feasible successors.)

EIGRP / IGRP Differences

	VLSM Support?	Load Balancing?	Classful/Classless	Authentication?
IGRP	No	Yes, equal and unequal-cost.	Classful	No
EIGRP	Yes	Yes, equal and unequal-cost.	Classless; performs autosummarization across network boundaries by default.	Yes, MD5 and text.

Configuring EIGRP

EIGRP uses Autonomous System numbers in its initial configuration, just like IGRP. The **network** command is slightly different; instead of naming the classful network number like IGRP, a wildcard mask is used to more specifically indicate the network number.



Configuring EIGRP AS 100 on the Frame Relay network.

```
R1#conf t
R1(config)#router eigrp 100
R1(config-router)#no auto-summary
R1(config-router)#network 172.12.123.0 0.0.0.255

R2#conf t
R2(config)#router eigrp 100
R2(config-router)#no auto-summary
R2(config-router)#network 172.12.123.0 0.0.0.255

R3#conf t
R3(config)#router eigrp 100
R3(config-router)#no auto-summary
R3(config-router)#network 172.12.123.0 0.0.0.255
```

Wildcard masks are used when configuring network numbers in EIGRP. Wildcard masks allow the configuration to be more specific in what interfaces will be running EIGRP. With the above wildcard masks, any interfaces in the network 172.12.123.0 /24 will run EIGRP.

A few seconds after configuring the three routers with EIGRP, this console message appears on R1:

```
R1#  
04:09:16: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.2  
(Serial0/0) is  
up: new adjacency  
04:09:19: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.3  
(Serial0/0) is  
up: new adjacency
```

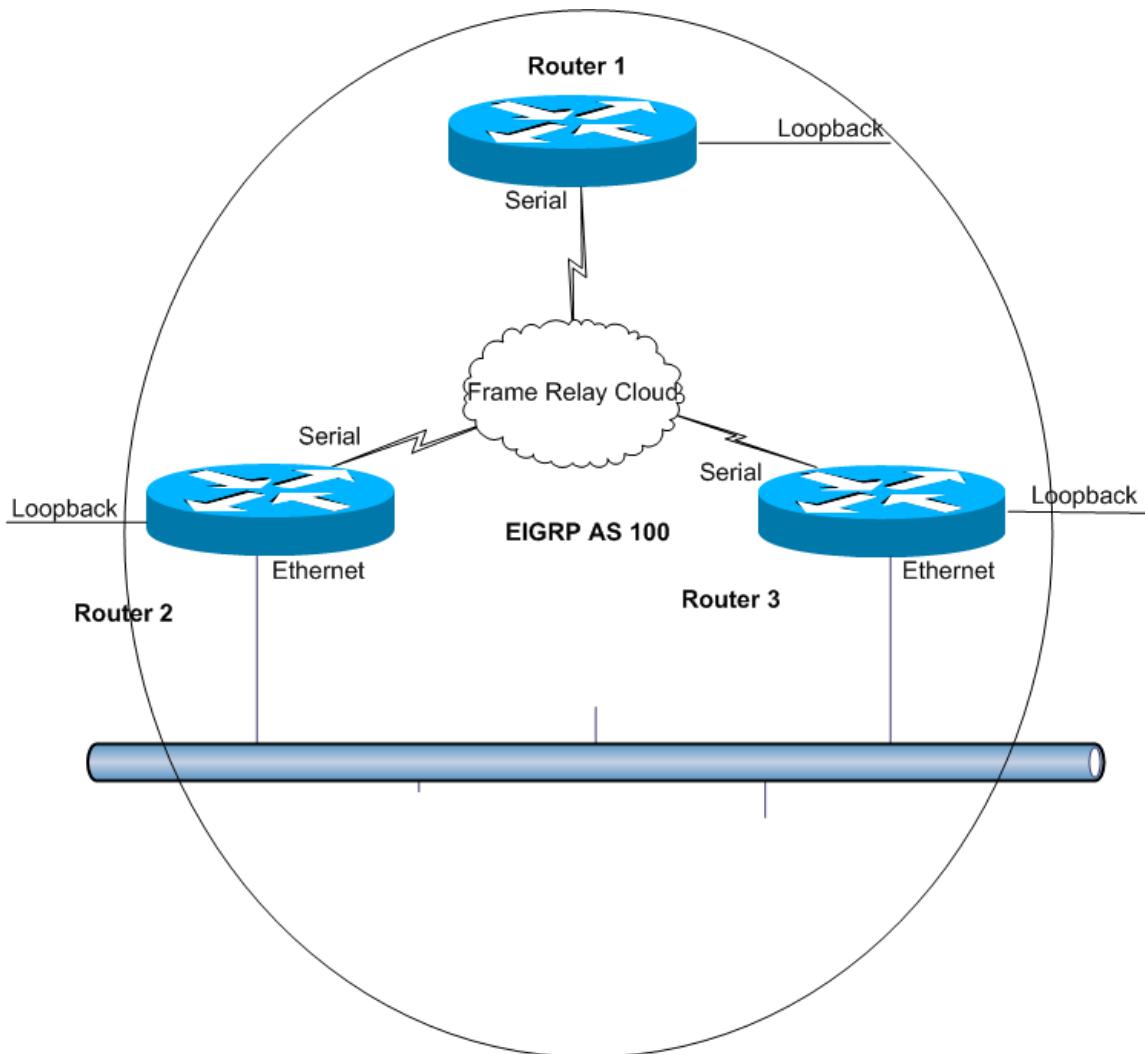
The Diffusing Update Algorithm (DUAL) has run and two new neighbors, 172.12.123.2 and 172.12.123.3, have formed adjacencies with R1.

Show ip eigrp neighbors gives the details:

```
R1#show ip eigrp neighbors  
IP-EIGRP neighbors for process 100  
H Address           Interface Hold Uptime SRTT  RTO  Q Seq Type  
          (sec)      (ms)        Cnt Num  
1 172.12.123.3     Se0/0    12 00:03:26  1    5000 0 3  
0 172.12.123.2     Se0/0    131 00:03:28 1    5000 0 3
```

The key values are the IP addresses of the EIGRP AS 100 neighbors, the interface on which they were discovered, and the Uptime, indicating how long the neighbor relationship has existed.

The loopbacks on each router will now be added to EIGRP 100, as well as the Ethernet subnet between R2 and R3. **show ip route eigrp 100** is then run at each router to ensure each router is seeing the other routers' loopbacks, and that R1 is seeing the Ethernet segment via EIGRP.



R1#show ip route eigrp 100

2.0.0.0/24 is subnetted, 1 subnets

D 2.2.2.0 [90/2297856] via 172.12.123.2, 00:00:22, Serial0/0

3.0.0.0/24 is subnetted, 1 subnets

D 3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:22, Serial0/0

172.23.0.0/27 is subnetted, 1 subnets

D 172.23.23.0 [90/2195456] via 172.12.123.3, 00:00:22, Serial0/0

[90/2195456] via 172.12.123.2, 00:00:22, Serial0/0

R2#show ip route eigrp 100

1.0.0.0/24 is subnetted, 1 subnets

D 1.1.1.0 [90/2297856] via 172.12.123.1, 00:02:36, Serial0/0.123

3.0.0.0/24 is subnetted, 1 subnets

D 3.3.3.0 [90/409600] via 172.23.23.3, 00:02:36, Ethernet0/0

R3#show ip route eigrp 100

1.0.0.0/24 is subnetted, 1 subnets

D 1.1.1.0 [90/2297856] via 172.12.123.1, 00:02:43, Serial0/0.31

2.0.0.0/24 is subnetted, 1 subnets

D 2.2.2.0 [90/409600] via 172.23.23.2, 00:02:43, Ethernet0/0

Note that the letter “D” indicates an EIGRP route.

Each router sees the other routers' loopbacks, and can ping them (ping results not shown). R1 can not only ping the Ethernet interfaces of R2 and R3, but has two routes to that subnet in its routing table. EIGRP is performing equal-cost load balancing. The metric for the route is 2195456 for both routes, so the load of any packets going from R1 to the 172.23.23.0 network will be balanced over the two Frame Relay cloud links.

To see the Successor and Feasible Successor routes in EIGRP, run **show ip eigrp topology**. On R1, two feasible successors for the route 172.23.23.0/27 exist, so both are placed into the routing table as seen previously. There are also two routes for destinations 2.2.2.0/24 and 3.3.3.0/24, but those have not been placed into the EIGRP routing table. Why?

R1#show ip eigrp topology

IP-EIGRP Topology Table for AS(100)/ID(150.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status

P 1.1.1.0/24, 1 successors, FD is 128256
via Connected, Loopback1

P 2.2.2.0/24, 1 successors, FD is 2297856

via 172.12.123.2 (2297856/128256), Serial0/0

via 172.12.123.3 (2323456/409600), Serial0/0

The first number in the parenthesis is the actual metric to the destination from this router; the second number is the metric advertised by the peer router. It is this value that is subject to the Feasibility Condition. If this value is less than the FD (Feasible Distance), the route is a Feasible Successor. If not, it cannot be a Feasible Successor.

P 3.3.3.0/24, 1 successors, FD is 2297856

via 172.12.123.3 (2297856/128256), Serial0/0

via 172.12.123.2 (2323456/409600), Serial0/0

P 172.23.23.0/27, 2 successors, FD is 2195456

via 172.12.123.2 (2195456/281600), Serial0/0

via 172.12.123.3 (2195456/281600), Serial0/0

P 172.12.123.0/24, 1 successors, FD is 2169856

via Connected, Serial0/0

R1#show ip route eigrp 100

2.0.0.0/24 is subnetted, 1 subnets

D 2.2.2.0 [90/2297856] via 172.12.123.2, 00:00:04, Serial0/0

3.0.0.0/24 is subnetted, 1 subnets

D 3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:04, Serial0/0

172.23.0.0/27 is subnetted, 1 subnets

D 172.23.23.0 [90/2195456] via 172.12.123.2, 00:00:04, Serial0/0

[90/2195456] via 172.12.123.3, 00:00:04, Serial0/0

R1 has two feasible successors to both 2.2.2.0/24 and 3.3.3.0/24 in its topology table, but is only using one route to each due to the unequal metrics.

R1 has two Feasible Successors for the Ethernet network; since they have an equal cost, load balancing is taking place.

R1 also has two routes for R2's loopback address, as well as R3's. The metrics for the second path are just a little higher, which prevents them from being entered into the EIGRP routing table. The **variance** command will be used to allow load sharing over unequal-cost links.

The **variance** command is a multiplier; the router will multiply the Feasible Distance by this value. Any feasible successor with a metric less than that new value will be entered into the routing table.

Consider the path from R1 to R2's loopback in the previous tables. The primary route has a metric of 2297856; the other route has a metric of 2323456. By default, the second route will serve only as a backup and will not carry packets unless the primary goes down.

By configuring **variance 2** on R1's EIGRP process, the process multiplies the metric of the best route by the **variance** value, resulting in 4595712. Any feasible successor with a metric less than this will now participate in load sharing. The other route does have a metric less than that, so it will now load share. After changing the variance value to 2 (by default, it's 1) and clearing the routing table, **show ip route eigrp 100** verifies that two valid routes to both R2's and R3's loopbacks appear in the EIGRP routing table.

```
R1#conf t
R1(config)#router eigrp 100
R1(config-router)#variance 2

R1#show ip route eigrp 100
 2.0.0.0/24 is subnetted, 1 subnets
 D  2.2.2.0 [90/2297856] via 172.12.123.2, 00:00:23, Serial0/0
               [90/2323456] via 172.12.123.3, 00:00:23, Serial0/0
 3.0.0.0/24 is subnetted, 1 subnets
 D  3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:23, Serial0/0
               [90/2323456] via 172.12.123.2, 00:00:23, Serial0/0
 172.23.0.0/27 is subnetted, 1 subnets
 D  172.23.23.0 [90/2195456] via 172.12.123.3, 00:00:23, Serial0/0
               [90/2195456] via 172.12.123.2, 00:00:23, Serial0/0
```

The **variance** command does not actually change the metrics; it makes a higher metric acceptable for load sharing.

EIGRP Authentication

EIGRP MD5 authentication is much like RIPv2. A key chain is created, and two interface-level commands are needed to begin authenticating packets.

```
R1#conf t
R1(config)#key chain AUTHENTICATE
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string CCNA
R1(config-keychain-key)#interface s0/0
R1(config-if)#ip authentication mode eigrp 100 md5
06:03:46: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.3 (Serial0/0) is
down: authentication mode changed
06:03:46: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.2 (Serial0/0) is
down: authentication mode changed
< The neighbor relationships have been lost since R1 is now running MD5 authentication and
the neighbors are not. >
R1(config-if)#ip authentication key-chain eigrp 100 AUTHENTICATE

R2#conf t
R2(config)#key chain AUTHENTICATE
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string CCNA
R2(config-keychain-key)#interface s0/0.123
R2(config-subif)#ip authentication mode eigrp 100 md5
R2(config-subif)#ip authentication key-chain eigrp 100 AUTHENTICATE

R3#conf t
R3(config)#key chain AUTHENTICATE
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string CCNA
R3(config-keychain-key)#interface s0/0.31
R3(config-subif)#ip authentication mode eigrp 100 md5
R3(config-subif)#ip authentication key-chain eigrp 100 AUTHENTICATE
```

```
R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
          H   Address      Interface  Hold  Uptime    SRTT    RTO  Q  Seq Type
                           (sec)      (ms)      Cnt Num
 1  172.12.123.3      Se0/0       12  00:00:12  1 5000 1  0
 0  172.12.123.2      Se0/0       60  00:00:20  522 3132 0  38
```

“show ip eigrp neighbor” shows the neighbor relationships have reestablished.

When debugging EIGRP, it's common to receive a message from the router to enable another debug first. Consider a situation where authentication was just configured on the EIGRP network running on the Frame Relay cloud, and the neighbor relationship between R1 and R2 does not reestablish. If **debug ip eigrp neighbor 100 172.12.123.2** is run, this message is received:

Examining the process of debugging EIGRP packets.

R1#**debug ip eigrp neighbor 100 172.12.123.2**

First enable IP-EIGRP Route Events or EIGRP packet debug

R1#**debug eigrp packet**

EIGRP Packets debugging is on

(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY, IAREPLY)

EIGRP packet debug has to be enabled before running most more-specific EIGRP debugs.

06:18:27: EIGRP: Sending UPDATE on Serial0/0 nbr 172.12.123.3, retry 7, RTO 500

06:18:27: AS 100, Flags 0x1, Seq 33/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rel

0/1 serno 6-35

06:18:28: EIGRP: Sending HELLO on Serial0/0

06:18:28: AS 100, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0

06:18:28: EIGRP: received packet with MD5 authentication, key id = 1

06:18:28: EIGRP: Received UPDATE on Serial0/0 nbr 172.12.123.3

Updates, HELLOs, and MD5-authenticated packets can now be viewed.

EIGRP Q & A

1. What statement best describes why EIGRP is considered a “hybrid” protocol?

- A. **EIGRP is a combination of IGRP and OSPF.**
- B. **EIGRP has characteristics of distance-vector and link-state protocols.**
- C. **EIGRP considers IGRP and OSPF route metrics when building its route table.**
- D. **EIGRP is not a hybrid; IGRP is.**

ANSWER: B. EIGRP has characteristics of distance-vector protocols, since it initially performs a full exchange of routing tables. EIGRP acts like a link-state protocol in that it uses multicast keepalives.

2. What word or phrase best describes a group of EIGRP routers that exchange routes?

- A. **Administratively Distant**
- B. **Link State**
- C. **Hybrid**
- D. **Autonomous System**
- E. **Neighbors**

ANSWER: D. An “autonomous system” is a logical grouping of EIGRP routers that exchange route information.

3. Two routers running EIGRP are directly connected. One is in AS 100; the other is in AS 200. What statement best describes how these routers will exchange routing information?

- A. **Since they're directly connected, they exchange route tables by default.**
- B. **Since they're directly connected, they cannot exchange routes under any circumstances.**
- C. **The AS number only matters when more than one protocol is involved; the routers will exchange routes normally.**

- D. The routers cannot exchange routes unless an AS number is changed, or unless route redistribution is configured between the two.**

ANSWER: D. Routers in different EIGRP autonomous systems will not exchange routes by default. Routes will only be exchanged in this situation if one router is placed in the other router's AS, or if redistribution is configured between the two autonomous systems.

4. A router is running IGRP AS 100; another router is running EIGRP AS 200. What is the default behavior?

- A. IGRP and EIGRP routers exchange routes by default.**
- B. IGRP and EIGRP routers never exchange routes by default.**
- C. IGRP and EIGRP routers only exchange routes by default if the AS number is the same; these routers will not exchange routes by default.**
- D. IGRP and EIGRP routers can only exchange routes if route redistribution is enabled.**

ANSWER: C. IGRP and EIGRP routers will exchange routes by default IF the AS numbers are the same. IGRP AS 100 and EIGRP AS 200 will not exchange routes by default.

5. Which of the following must occur for two EIGRP routers to become neighbors? (Choose three.)

- A. Hello packets must be exchanged.**
- B. The AS number must match.**
- C. The route metrics must match.**
- D. The metric weights must match.**
- E. The IGRP / EIGRP code must match.**
- F. The subnet mask length must match.**

ANSWER: A, B, D. Hello packets serve as the keepalive for EIGRP, and must be exchanged for the neighbor relationship to take place. The AS number must match, and the metric weights must match. EIGRP does not care about subnet mask length when neighbor relationships are formed.

6. In which table does EIGRP keep the best route for a destination?

- A. The route table.
- B. The topology table.
- C. The neighbor table.
- D. The STP table.

ANSWER: A. EIGRP keeps the best route to a destination in the routing table. EIGRP does not keep an STP table.

7. What statement best describes the contents of the EIGRP topology table?

- A. The topology table contains a logical map of the entire network, regardless of what other protocols are running.
- B. The topology table contains a single summary route for use in case of primary route failure.
- C. The topology table contains all successor routes.
- D. The topology table contains all successors and feasible successors.

ANSWER: D. While the route table contains the best route to a given destination, the topology table contains all feasible successors.

8. What is the difference between a “successor” and a “feasible successor”? (Choose two.)

- A. A successor is the primary route; a feasible successor is a backup route.
- B. A successor is a backup route; the feasible successor is the primary route.
- C. There can only be one successor, but there can be multiple successors.
- D. There can be multiple successors, but only one feasible successor.
- E. There can be multiple successors and multiple feasible successors.
- F. There can be only one successor and only one feasible successor.

ANSWER: A, E. Even though the successor is generally referred to as “the” successor, remember that EIGRP performs equal-cost load balancing by default. If there are two equal-cost best routes to a destination, both routes are considered successors.

9. What algorithm does EIGRP use to determine feasible successors?

- A. **The Dijkstra algorithm.**
- B. **DUAL.**
- C. **Loop-Prevention Algorithm (LPA)**
- D. **Best Path Detector (BPD)**

ANSWER: B. EIGRP uses DUAL to determine feasible successors.

Answers C and D do not exist.

10. By default, how many equal-cost paths to a given destination will be placed into the EIGRP routing table?

- A. **One.**
- B. **Two.**
- C. **Three.**
- D. **Four.**
- E. **None.**

ANSWER: D. EIGRP places up to four equal-cost paths to the same destination into the EIGRP routing table.

11. By default, how many unequal-cost paths to a given destination will be placed into the EIGRP routing table?

- A. **One.**
- B. **Two.**
- C. **Three.**
- D. **Four.**
- E. **None.**

ANSWER: E. No unequal-cost paths are automatically placed into the EIGRP routing table.

12. Your boss has told you to absolutely prevent EIGRP from performing equal-cost load balancing. What command will help you do this?

- A. **no ip eigrp load-balance**
- B. **no eigrp load-balance**
- C. **maximum-path 1**
- D. **eigrp load-balance path 1**
- E. **eigrp load-balance path 0**

ANSWER: C. Setting the maximum-path value to 1 will do the job.
The other four commands do not exist.

13. Which of the following statements is true? (Choose two.)

- A. **EIGRP and IGRP both support variable-length subnet masks.**
- B. **EIGRP supports variable-length subnet masks, but IGRP does not.**
- C. **Neither EIGRP nor IGRP support variable-length subnet masks.**
- D. **EIGRP and IGRP support MD5 authentication.**
- E. **EIGRP supports MD5 authentication, but IGRP does not.**
- F. **Neither EIGRP nor IGRP support MD5 authentication.**

ANSWER: B, E. Remember that the "E" in EIGRP stands for "Enhanced". Two of the enhancements are that EIGRP supports VLSMs and MD5 authentication.

14. Consider the following configuration:

Configuring EIGRP AS 100 on the Frame Relay network.

```
R1#conf t  
R1(config)#router eigrp 100  
R1(config-router)#no auto-summary  
R1(config-router)#network 172.12.123.0 0.0.0.255
```

Which of the following statements is true?

- A. **Auto-summarization is off by default in EIGRP and does not need to be disabled manually.**
- B. **The line "router eigrp 100" should read "router eigrp AS 100".**
- C. **The dotted decimal number at the end of the "network" statement should read "255.255.255.0".**
- D. **The configuration is valid.**

ANSWER: D. The configuration is valid as is. The dotted decimal number at the end of the network statement is a wildcard mask, and is correctly configured. Auto-summarization is on in EIGRP by default.

15. Consider this router output:

R1# < command removed >							
IP-EIGRP neighbors for process 100							
H	Address	Interface	Hold (sec)	Uptime (ms)	SRTT Cnt	RTO Num	Q Seq Type
1	172.12.123.3	Se0/0	12	00:03:26	1 5000	0 3	
0	172.12.123.2	Se0/0	131	00:03:28	1 5000	0 3	

What command produced this output?

- A. **show ip eigrp**
- B. **show ip eigrp neighbor**
- C. **show ip eigrp topology**
- D. **show ip eigrp process**
- E. **show ip neighbor**

ANSWER: B. "show ip eigrp neighbor" displays the IP address, and other information, of the router's EIGRP neighbors.

16. What letter(s) indicates an EIGRP route in a routing table?

- A. D
- B. E
- C. G
- D. EI
- E. EIGRP

ANSWER: A. The letter "D" indicates an EIGRP route.

17. What command displays only the successor(s) for an EIGRP route?

- A. **show ip route**
- B. **show ip eigrp topology**
- C. **show ip eigrp neighbor**
- D. **show ip eigrp successor**
- E. **show ip eigrp route**

ANSWER: A. Remember that the successor is the best route. The only one of these five commands that will display only the best route for an EIGRP destination is "show ip route". The topology table displays successors AND feasible successors.

18. What command assists with EIGRP unequal-cost load balancing?

- A. **ip eigrp load-balance**
- B. **eigrp load-balance unequal**
- C. **variance**
- D. **eigrp variance**
- E. **no ip eigrp equal-cost**

ANSWER: C. The variance command is used to permit unequal-cost load balancing.

The next three questions all refer to the following table:

R1#< command removed >
IP-EIGRP Topology Table for AS(100)/ID(150.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status

P 1.1.1.0/24, 1 successors, FD is 128256
 via Connected, Loopback1
P 2.2.2.0/24, 1 successors, FD is 2297856
 via 172.12.123.2 (2297856/128256), Serial0/0
 via 172.12.123.3 (2323456/409600), Serial0/0
P 3.3.3.0/24, 1 successors, FD is 2297856
 via 172.12.123.3 (2297856/128256), Serial0/0
 via 172.12.123.2 (2323456/409600), Serial0/0
P 172.23.23.0/27, 2 successors, FD is 2195456
 via 172.12.123.3 (2195456/281600), Serial0/0
 via 172.12.123.2 (2195456/281600), Serial0/0
P 172.12.123.0/24, 1 successors, FD is 2169856
 via Connected, Serial0/0

19. What command produced this output?

- A. **show ip eigrp route 100**
- B. **show ip eigrp route**
- C. **show ip eigrp topology**
- D. **show ip eigrp topology passive**
- E. **show ip eigrp successors**

ANSWER: C. "show ip eigrp topology" displays this vital information.

20. The routes are all shown as "P", or "passive". What does this mean, and what action should be taken?
- A. The routes are valid, but because of downstream network problems, they cannot be used. When the problem is resolved, the paths will go to "active" and can be used.
 - B. The routes are valid and can be used.
 - C. The routes are invalid, and DUAL is in the process of discovering the problem. Until that is done, the routes will stay in passive mode.
 - D. The routes are invalid because of an EIGRP misconfiguration.

ANSWER: B. The normal and desired behavior of an EIGRP route is to be in passive mode. A route in active is being recalculated and cannot be used until it goes to passive.

21. Viewing the table, what can be said about load balancing?
(Choose three.)

- A. Both equal-cost and unequal-cost load balancing are in effect.
- B. No load balancing is in effect.
- C. Equal-cost load balancing is in effect, but unequal-cost load balancing is not.
- D. Load balancing is occurring across one link.
- E. Load balancing is occurring across two links.
- F. Load balancing is not occurring across any links.
- G. The "variance 3" command is in use.
- H. The "variance 2" command is in use.
- I. The "variance" command was not configured.

ANSWER: C, D, I. There are multiple routes to networks 2.2.2.0 and 3.3.3.0, but only one successor. That means that equal-cost load balancing is in use, but unequal-cost load balancing is not. Load balancing is taking place for traffic destined for the 172.23.23.0 network, since the cost is exactly the same. (Remember that equal-cost load balancing is enabled by default in EIGRP.) The "variance" command does not have to be configured if only equal-cost load balancing is in effect.

```
R1#show ip eigrp topology
```

```
P 2.2.2.0/24, 1 successors, FD is 2297856
    via 172.12.123.2 (2297856/128256), Serial0/0
    via 172.12.123.3 (2323456/409600), Serial0/0

P 3.3.3.0/24, 1 successors, FD is 2297856
    via 172.12.123.3 (2297856/128256), Serial0/0
    via 172.12.123.2 (2323456/409600), Serial0/0
```

22. Consider the above table. What is the minimum value of the variance command that will be needed to enable unequal-cost load balancing to networks 2.2.2.0 and 3.3.3.0?

- A. **variance 1**
- B. **variance 2**
- C. **variance 3**
- D. **variance 4**
- E. **variance 10**
- F. **variance 20**

ANSWER: B. The Feasible Distance of the successor route to both networks is 2297856. The Feasible Distance of the feasible successor is 2323456. Using “variance 2” effectively multiplies the Feasible Distance of the successor. $2297856 \times 2 = 4595712$. Any route with a Feasible Distance less than 4595712 will participate in unequal-cost load sharing.

The answers with values greater than “2” would work, but note the question asked for the minimum value. Watch details like that when you’re passing the CCNA exam.

EIGRP Lab

Configure EIGRP AS 100 on R1, R2, and R3 over the Frame Relay cloud. Disable EIGRP's automatic summarization with the **no auto-summary** command.

```
Configuring EIGRP Autonomous System 100

R1#conf t
R1(config)#router eigrp 100
R1(config-router)#no auto-summary
R1(config-router)#network 172.12.123.0 0.0.0.255

R2#conf t
R2(config)#router eigrp 100
R2(config-router)#no auto-summary
R2(config-router)#network 172.12.123.0 0.0.0.255

R3#conf t
R3(config)#router eigrp 100
R3(config-router)#no auto-summary
R3(config-router)#network 172.12.123.0 0.0.0.255
```

On R1, run **show ip eigrp neighbor**.

```
R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
          H  Address           Interface   Hold Uptime   SRTT   RTO   Q   Seq Type
                           (sec)      (ms)      Cnt Num
  1  172.12.123.3     Se0/0       11 00:02:45  1 5000 0 1
  0  172.12.123.2     Se0/0      161 00:03:01  1 5000 0 1
```

On each router, add the loopback address to the EIGRP process.

```
R1#conf t
R1(config)#router eigrp 100
R1(config-router)#network 1.1.1.1 0.0.0.0

R2#conf t
R2(config)#router eigrp 100
R2(config-router)#network 2.2.2.2 0.0.0.0

R3#conf t
R3(config)#router eigrp 100
R3(config-router)#network 3.3.3.3 0.0.0.0
```

On each router, run **show ip route eigrp**. R1 has a route for both R2's and R3's loopback. R2 and R3 will only see R1's loopback address, and not each other's. Why?

```
R1#show ip route eigrp
 2.0.0.0/24 is subnetted, 1 subnets
 D  2.2.2.0 [90/2297856] via 172.12.123.2, 00:03:19, Serial0/0
 3.0.0.0/24 is subnetted, 1 subnets
 D  3.3.3.0 [90/2297856] via 172.12.123.3, 00:03:04, Serial0/0

R2#show ip route eigrp
 1.0.0.0/24 is subnetted, 1 subnets
 D  1.1.1.0 [90/2297856] via 172.12.123.1, 00:03:40, Serial0/0.123

R3#show ip route eigrp
 1.0.0.0/24 is subnetted, 1 subnets
 D  1.1.1.0 [90/2297856] via 172.12.123.1, 00:05:17, Serial0/0.31
```

EIGRP uses **Split Horizon** by default to prevent looping. In this lab, though, it prevents full network reachability. R2 and R3 both form neighbor relationships with R1's Serial physical interface. R2 advertises its loopback address to R1's Serial interfaces, as does R3. **Split Horizon does not allow a route to be advertised back out the same interface it was received on.** This prevents R1 from advertising R2's loopback to R3, or R3's loopback to R2.

Split Horizon must be disabled to allow full network reachability in this lab. To do so, run **no ip split-horizon eigrp 100** on R1's Serial interface. When Split Horizon is disabled, that will cause the neighbor

relationships to fail, and then reestablish. Clear the route table with **clear ip route *** and run **show ip route eigrp 100** on both R2 and R3. The appropriate route to the remote loopback address will now appear. From each router, ping the other routers' loopbacks. All pings will succeed.

Disabling Split Horizon on R1's Serial interface to allow full network connectivity.

```
R1#conf t
R1(config)#int serial0/0
R1(config-if)#no ip split-horizon eigrp 100

10:02:23: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.2 (Serial0/0) i
down: split horizon changed
10:02:23: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.3 (Serial0/0) i
down: split horizon changed

10:02:27: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.3 (Serial0/0) i
up: new adjacency
10:02:54: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.2 (Serial0/0) i
up: new adjacency
```

< The adjacencies come down after Split Horizon is changed, but are back within 30 seconds. >

```
R2#show ip route eigrp
 1.0.0.0/24 is subnetted, 1 subnets
 D  1.1.1.0 [90/2297856] via 172.12.123.1, 00:00:06, Serial0/0.123
 3.0.0.0/24 is subnetted, 1 subnets
 D  3.3.3.0 [90/2809856] via 172.12.123.1, 00:00:06, Serial0/0.123
```

```
R3#show ip route eigrp
 1.0.0.0/24 is subnetted, 1 subnets
 D  1.1.1.0 [90/2297856] via 172.12.123.1, 00:00:12, Serial0/0.31
 2.0.0.0/24 is subnetted, 1 subnets
 D  2.2.2.0 [90/2809856] via 172.12.123.1, 00:00:12, Serial0/0.31
```

< R2 now has a route to R3's loopback, and R3 to R2's. Ping results not shown. >

Add the Ethernet segment between R2 and R3 to EIGRP AS 100.

```
R2#conf t
R2(config)#router eigrp 100
R2(config-router)#network 172.23.23.0 0.0.0.255

R3#conf t
R3(config)#router eigrp 100
R3(config-router)#network 172.23.23.0 0.0.0.255
```

Run **show ip eigrp neighbor** on each router.

```
R2#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
H Address           Interface Hold Uptime SRTT  RTO  Q Seq Type
                  (sec)   (ms)    Cnt Num
1 172.23.23.3     Et0/0    12 00:03:29  4  200 0 15
0 172.12.123.1   Se0/0.123 126 00:11:16  40 240 0 15

R3#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
H Address           Interface Hold Uptime SRTT  RTO  Q Seq Type
                  (sec)   (ms)    Cnt Num
1 172.23.23.2     Et0/0    11 00:03:34 1529 5000 0 14
0 172.12.123.1   Se0/0.31 176 00:11:24  40 240 0 16
```

Run **show ip eigrp topology** to look at the Successor and Feasible Successor routes on R1.

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(150.1.1.1)
```

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status

P 1.1.1.0/24, 1 successors, FD is 128256
via Connected, Loopback1

P 2.2.2.0/24, 1 successors, FD is 2297856
via 172.12.123.2 (2297856/128256), Serial0/0
via 172.12.123.3 (2323456/409600), Serial0/0

P 3.3.3.0/24, 1 successors, FD is 2297856
via 172.12.123.3 (2297856/128256), Serial0/0
via 172.12.123.2 (2323456/409600), Serial0/0

P 172.23.23.0/27, 2 successors, FD is 2195456
via 172.12.123.3 (2195456/281600), Serial0/0
via 172.12.123.2 (2195456/281600), Serial0/0

P 172.12.123.0/24, 1 successors, FD is 2169856
via Connected, Serial0/0

According to the code list at the top of this command output, the “P” code stands for Passive, and all these routes have a “P” next to them. Is this good? Yes. A “passive” EIGRP route means that it is not currently being calculated by DUAL. An “active” EIGRP route means that it is being calculated. A route that stays in active state cannot be used to transport packets; such a route is said to be “SIA”, or “stuck in active”.

R1 has two Successor routes for the Ethernet network. Why? First, the EIGRP process checks to see if the routes meet the Feasibility Condition. The Feasible Distance, the best metric the router has for that destination, is 2195456. That happens to be the same metric for both possible routes, and since the Advertised Distance (281600) for both routes is less than the Feasible Distance, both routes are Feasible Successors. Since the metric for both paths is exactly the same, equal-cost load balancing will occur, and both routes are placed into the topology table as Successors, and both will be placed into the EIGRP routing table.

R1's current EIGRP routing table. Two equal-cost routes to 172.23.23.0 have been placed into the routing table, but only one for each of the loopbacks, even though there are two routes for each in the topology table.

R1#show ip route eigrp 100

```
2.0.0.0/24 is subnetted, 1 subnets
D  2.2.2.0 [90/2297856] via 172.12.123.2, 00:14:09, Serial0/0
  3.0.0.0/24 is subnetted, 1 subnets
D  3.3.3.0 [90/2297856] via 172.12.123.3, 00:14:11, Serial0/0
  172.23.0.0/27 is subnetted, 1 subnets
D  172.23.23.0 [90/2195456] via 172.12.123.2, 00:14:14, Serial0/0
    [90/2195456] via 172.12.123.3, 00:14:14, Serial0/0
```

Consider R1's two possible routes to R2's loopback and R3's loopback from the EIGRP topology table:

Partial output of R1's EIGRP topology table.

R1#show ip eigrp topology

```
P 2.2.2.0/24, 1 successors, FD is 2297856
  via 172.12.123.2 (2297856/128256), Serial0/0
  via 172.12.123.3 (2323456/409600), Serial0/0

P 3.3.3.0/24, 1 successors, FD is 2297856
  via 172.12.123.3 (2297856/128256), Serial0/0
  via 172.12.123.2 (2323456/409600), Serial0/0
```

Remember: The first number in the parenthesis is the route's Feasible Distance; the second number is the Advertised Distance.

The Feasible Distance for this route is 2297856; that is the best metric the router has for the route. The first route in the list has this FD, and will be the Successor (primary route).

The second route must meet the Feasibility Condition. Is its Advertised Distance lower than the Feasible Distance (FD) of the Successor? Yes. The route's Advertised Distance is 409600; the FD is 2297856. The route meets the Feasibility Condition and is placed into the topology table. It is now a Feasible Successor; it can be used if the Successor fails, but by default, it will not participate in load-sharing. The same can be said for the two paths to R3's loopback.

Configure the EIGRP network to load-balance over these two possible paths to each loopback address with the appropriate **variance** command. Recall that the **variance** command is a multiplier; the router will multiply the Feasible Distance by this value. If a feasible successor has a metric less than that of this equation, the route will be placed into the EIGRP routing table and used for load-balancing.

The Feasible Distance in each case is 2297856; the metric for the Feasible Successor in each case is 2323456. Since that's barely higher than the Feasible Distance, a variance value of 2 will do the job. Configure **variance 2** under the EIGRP process on R1, clear the routing table with **clear ip route ***, and run **show ip route eigrp**.

Setting the EIGRP variance on R1 to 2, and clearing the routing table.

```
R1#conf t
R1(config)#router eigrp 100
R1(config-router)#variance 2

R1#clear ip route *
R1#show ip route eigrp
  2.0.0.0/24 is subnetted, 1 subnets
D  2.2.2.0 [90/2297856] via 172.12.123.2, 00:00:04, Serial0/0
    [90/2323456] via 172.12.123.3, 00:00:04, Serial0/0
  3.0.0.0/24 is subnetted, 1 subnets
D  3.3.3.0 [90/2297856] via 172.12.123.3, 00:00:04, Serial0/0
    [90/2323456] via 172.12.123.2, 00:00:04, Serial0/0
  172.23.0.0/27 is subnetted, 1 subnets
D  172.23.23.0 [90/2195456] via 172.12.123.3, 00:00:04, Serial0/0
    [90/2195456] via 172.12.123.2, 00:00:04, Serial0/0
```

*The **variance** command allows any feasible successor with a metric of less than (2297856 x 2) to participate in load-balancing. R1 can now use both routes to R2's and R3's loopback network.*

Configure authentication on EIGRP AS 100's connections over the Frame Relay cloud only. First, build a key chain called CLOUD with a key-string of GOTMYCCNA ; then configure EIGRP authentication on the appropriate Serial interfaces on all three routers.

```
R1#conf t
R1(config)#key chain CLOUD
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string GOTMYCCNA
R1(config-keychain-key)#interface serial0/0
R1(config-if)#ip authentication mode eigrp 100 md5
10:52:00: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.2
(Serial0/0) is down: authentication mode changed
10:52:00: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 172.12.123.3
(Serial0/0) is down: authentication mode changed
< The EIGRP adjacencies go down when one interface is configured for authentication and the others are not. They will come back up when authentication has been configured correctly on R2 and R3. >
R1(config-if)#ip authentication key-chain eigrp 100 CLOUD

R2#conf t
R2(config)#key chain CLOUD
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string GOTMYCCNA
R2(config-keychain-key)#interface serial0/0.123
R2(config-subif)#ip authentication mode eigrp 100 md5
R2(config-subif)#ip authentication key-chain eigrp 100 CLOUD

R3#conf t
R3(config)#key chain CLOUD
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string GOTMYCCNA
R3(config-keychain-key)#interface s0/0.31
R3(config-subif)#ip authentication mode eigrp 100 md5
R3(config-subif)#ip authentication key-chain eigrp 100 CLOUD
```

On R1, run **show ip eigrp neighbor** to ensure the neighbor relationships came up after authentication was completed.

```
R1#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
      H  Address          Interface  Hold Uptime   SRTT  RTO  Q  Seq Type
                           (sec)      (ms)    Cnt Num
 1  172.12.123.3      Se0/0       12 00:00:06   1 4500 1 0
 0  172.12.123.2      Se0/0      157 00:00:24 1183 5000 0 22
```

Run **debug eigrp packet** to watch Hello and MD5 packets traverse the interface. Run **undebbug all** when done.

```
R1#debug eigrp packet
```

EIGRP Packets debugging is on

(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK,
STUB, SIAQUERY,IAREPLY)

11:03:11: EIGRP: received packet with MD5 authentication, key id = 1

11:03:11: EIGRP: Received HELLO on Serial0/0 nbr 172.12.123.3

11:03:11: AS 100, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely

11:03:15: EIGRP: received packet with MD5 authentication, key id = 1

11:03:15: EIGRP: Received HELLO on Serial0/0 nbr 172.12.123.3

11:03:19: EIGRP: received packet with MD5 authentication, key id = 1

11:03:19: EIGRP: Received HELLO on Serial0/0 nbr 172.12.123.2

Section Ten: Advanced TCP/IP Concepts and Topics

NAT is one of the main reasons CCNA candidates fail the exam; or rather, it's lack of NAT knowledge.

The reason is simple. Many CCNA books completely overlook NAT; if they don't, they explain it broadly or badly. You must know how NAT works, how PAT works, and how to configure static and dynamic NAT. All that information is contained in this chapter, as well as your introduction to access-lists and how to develop a VLSM scheme.

Commands Introduced In This Chapter And Labs:

access-class -- When an access-list is used to limit Telnet access, this command is used in line configuration mode to apply the list to the vty lines.

access-list <number> -- Global command to write a standard or extended access-list.

ip access-group – Used to apply an access-list to an interface.

ip access-list -- Used to create a named access list.

ip summary-address eigrp – Interface-level command used to configure a summarized address in EIGRP.

ip summary-address rip – Interface-level command used to configure a summarized address in RIP.

Standard, Extended, and Named Access Lists

Access Control Lists (ACLs) , applied at the interface level, cause a router to permit or deny packets based on a variety of criteria. The ACL is configured in global mode, but is applied at the interface level. An ACL does not take effect until it is expressly applied to an interface with the **ip access-group** command. Packets can be filtered as they enter or exit an interface.

If a packet enters or exits an interface with an ACL applied, the packet is compared against the criteria of the ACL. If the packet matches the first line of the ACL, the appropriate "permit" or "deny" action is taken. If there is no match, the second line's criteria is examined. Again, if there is a match, the appropriate action is taken; if there is no match, the third line of the ACL is compared to the packet.

This process continues until a match is found, at which time the ACL stops running. If no match is found, a default "deny" takes place, and the packet will not be processed. When an ACL is configured, if a packet is not expressly permitted, it will be subject to the **implicit deny** at the end of every ACL. This is the default behavior of an ACL and cannot be changed.

A standard ACL is concerned with only one thing: the source of the packet. The destination is not important. Extended ACLs consider both the source and destination of the packet, and can consider the port number as well. The numerical range used for each is different: standard ACLs use the ranges 1-99 and 1300-1399; extended lists use 100-199 and 2000 to 2699.

The Wildcard Mask

ACLs use *wildcard masks* to determine what part of a network number should and should not be examined for matches against the ACL.

Wildcard masks are written in binary math, and then converted to dotted decimal for router configuration. Zeroes indicate to the router that this particular bit must match, and ones are used as "I don't care" bits – the ACL does not care if there is a match or not.

Consider this situation: Packets coming into a router's Ethernet0 interface should be compared to the ACL. All packets coming in from network number 196.17.100.0 / 24 should be permitted, and all others should be dropped. The router needs to be told that the first 24 bits of

the network number must match, and that it doesn't matter what the last 8 bits are – "I don't care".

1 st Octet – All bits must match.	00000000
2 nd Octet – All bits must match.	00000000
3 rd Octet – All bits must match.	00000000
4 th Octet – "I don't care"	11111111
Resulting Wildcard Mask:	00000000 00000000 00000000 11111111

Use this binary math chart to convert from binary to dotted decimal:

	128	64	32	16	8	4	2	1
1 st Octet:	0	0	0	0	0	0	0	0
2 nd Octet:	0	0	0	0	0	0	0	0
3 rd Octet:	0	0	0	0	0	0	0	0
4 th Octet:	1	1	1	1	1	1	1	1

Converted to dotted decimal, the wildcard mask is written 0.0.0.255.

Wildcard masks are also used with routing protocols OSPF and EIGRP. Consider a router with the following interfaces:

Interface serial0	172.12.12.12 /28
Interface serial1	172.12.12.17 /28

The two interfaces are on different subnetworks. Serial0 is on the 172.12.12.0 /28 subnet, where Serial1 is on the 172.12.12.16 /28 subnet. Wildcard masks allow OSPF to be run only on Serial0.

The wildcard mask will require the first 28 bits to match 172.12.12.0; the mask doesn't care what the last 4 bits are.

1 st Octet: All bits must match.	00000000
2 nd Octet: All bits must match.	00000000
3 rd Octet: All bits must match.	00000000
4 th Octet: First four bits must match.	00001111
Resulting Wildcard Mask:	00000000 00000000 00000000 00001111

Use this binary math chart to convert from binary to dotted decimal:

	128	64	32	16	8	4	2	1
1 st Octet:	0	0	0	0	0	0	0	0
2 nd Octet:	0	0	0	0	0	0	0	0
3 rd Octet:	0	0	0	0	0	0	0	0
4 th Octet:	0	0	0	0	1	1	1	1

Converted to dotted decimal, the wildcard mask is 0.0.0.15.

Configuring Standard Access Lists

There are several points worth repeating before beginning to configure standard ACLs:

- Standard ACLs consider only the source IP address for matches.
- The ACL lines are run from top to bottom; if there is no match on the first line, the second is run; if no match on the second, the third is run, and so on until there is a match, or the end of the ACL is reached. **This top-to-bottom process places special importance on the order of the lines.**
- There is an **implicit deny** at the end of every ACL. If packets are not expressly permitted, they are implicitly denied.

If Router 3's Ethernet interface should only accept packets with a source network of 172.12.12.0, the ACL will be configured like this:

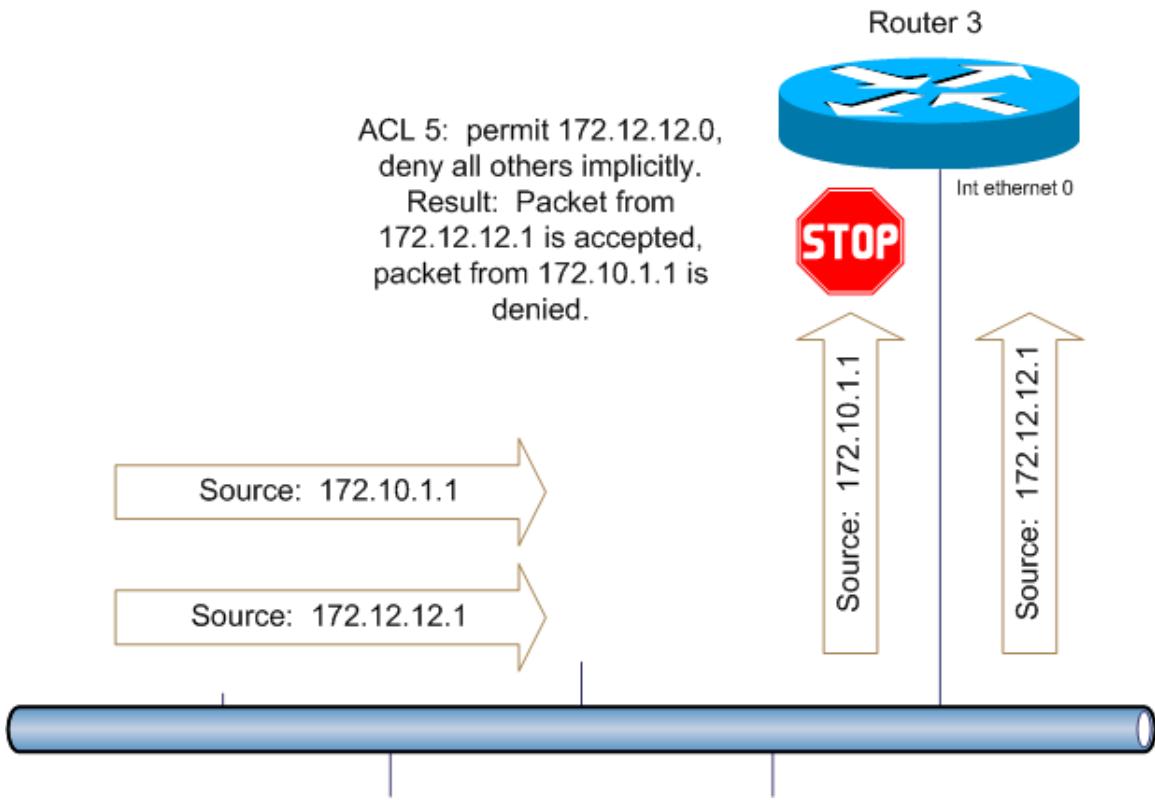
```
R3#conf t  
R3(config)#access-list 5 permit 172.12.12.0 0.0.0.255
```

The ACL consists of only one explicit line, one that permits packets from source IP address 172.12.12.0 /24. The implicit deny, which is not configured or seen in the running configuration, will deny all packets not matching the first line.

The ACL is then applied to the Ethernet0 interface:

```
R3#conf t  
R3(config)#interface e0  
R3(config-if)#ip access-group 5 in
```

The access-list is written in global configuration mode, and then applied to the desired interface with the ip access-group command, followed by the ACL number and the direction in which the ACL is applied, either in or out.



Adding Remarks

Access lists can become quite large and intricate. If one engineer writes an ACL and another engineer comes in six months later to troubleshoot an issue, that second engineer may not know what the ACL was trying to accomplish. It's good form to add a remark line or two to describe what an ACL was written for. To do so, use the **remark** ACL command:

```
R3#conf t
R3(config)#access-list 5 permit 172.12.12.0 0.0.0.255
R3(config)#access-list 5 remark Permit network 172.12.12.0 only.
```

Using "Host" and "Any" for Wildcard Masks of 0.0.0.0 and 255.255.255.255

It is acceptable to configure a wildcard mask of all ones or all zeroes. A wildcard mask of 0.0.0.0 means the address specified in the ACL line

must be matched exactly; a wildcard mask of 255.255.255.255 means that all addresses will match the line.

Wildcard masks have the option of using the word **host** to represent a wildcard mask of 0.0.0.0. Consider a configuration where only packets from IP source 10.1.1.1 should be allowed and all other packets denied. The following configurations are both valid:

```
R3#conf t  
R3(config)#access-list 6 permit 10.1.1.1 0.0.0.0
```

```
R3(config)#conf t  
R3(config)#access-list 7 permit host 10.1.1.1
```

These two access lists perform the same task. In the first, the wildcard mask of 0.0.0.0 means that all 32 bits of the address must be matched for the ACL to be applied; in other words, only the specified address will be a match.

In the second ACL, the keyword “host” is used to indicate to the router that the address following the command is the specific address to be matched.

The keyword **any** can be used to represent a wildcard mask of 255.255.255.255.

Consider a situation where packets sourced from 172.18.18.0 /24 will be denied, but all others will be permitted:

```
R3#conf t  
R3(config)#access-list 15 deny 172.18.18.0 0.0.0.255  
R3(config)#access-list 15 permit any
```

The first line of the ACL denies traffic from 172.18.18.0/24. The second line permits all traffic, using the “any” keyword instead of typing out a network number and a wildcard mask of 255.255.255.255.

The previous example also illustrates the importance of configuring the ACL with the lines in the correct order to get the desired results. What would be the result if the lines were reversed?

Reversing the lines of the previous ACL.

```
R3#conf t  
R3(config)#access-list 15 permit any  
R3(config)#access-list 15 deny 172.18.18.0 0.0.0.255
```

If the lines were reversed, traffic from 172.18.18.0 /24 would be matched against the first line of the ACL. The first line is “permit any”, meaning all traffic is permitted. The traffic from 172.18.18.0/24 matches that line, the traffic is permitted, and the ACL stops running.
The statement denying the traffic from 172.18.18.0 is never run.

Extended Access Control Lists

Extended ACLs allow both the IP source and destination address to be matched. The source port, destination port, and protocol type can also be matched.

R3 has a new restriction. The previous ACL has been removed, and now packets sourced from network 172.50.50.0 / 24 should not be permitted if they are destined for network 172.50.100.0. All other packets should be allowed. The ACL is configured as follows:

Examining the configuration of an extended access control list.

```
R3#conf t
R3(config)#access-list 150 deny ip 172.50.50.0 0.0.0.255 172.50.100.0 0.0.0.255
R3(config)#access-list 150 permit ip any any
R3(config)#no ip access-group 5 in
R3(config)#ip access-group 150 in
```

Line 1:

Access-list 150: Extended lists use ranges 100 – 199 and 2000 – 2699.

Deny: Matching packets will be denied.

Ip : The protocol to be matched.

172.50.50.0 0.0.0.255: The source address to be used when matching packets.

172.50.100.0 0.0.0.255: The destination address to be used when matching packets.

For a packet to match this line, both the source and destination must match.

Line 2:

Permit: Matching packets will be permitted.

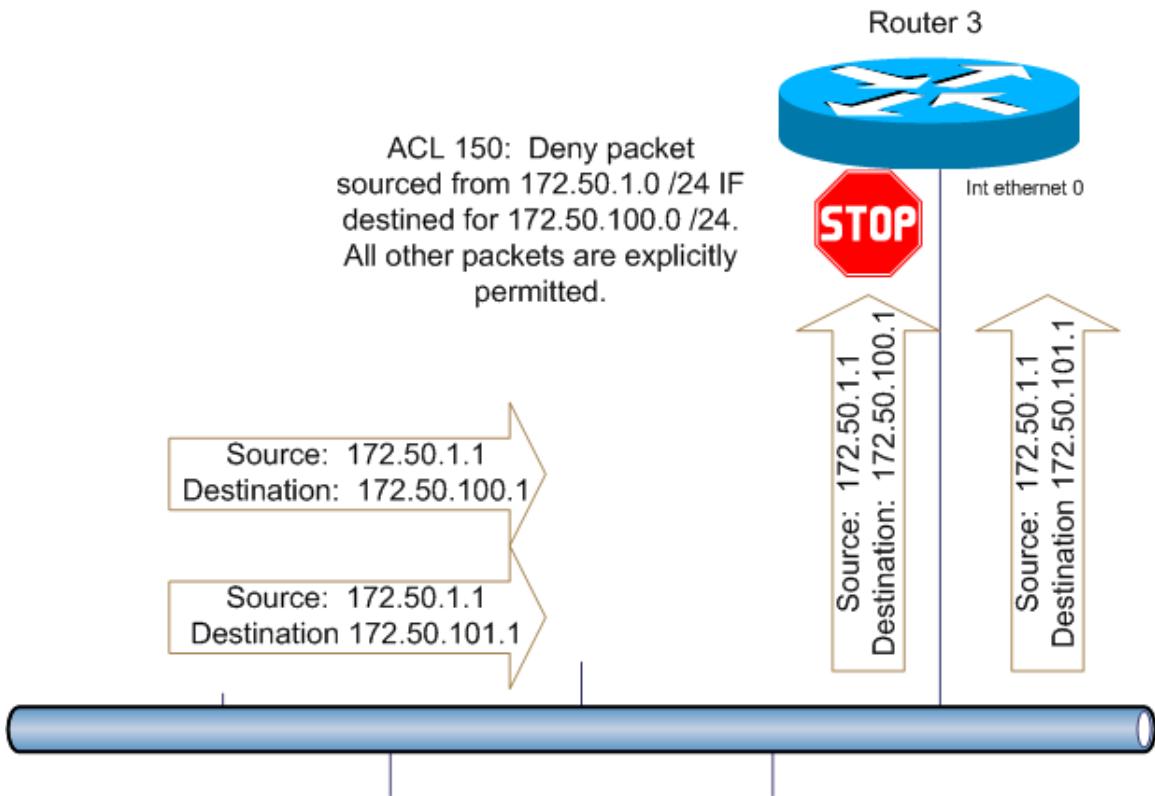
Ip : The protocol to be matched.

Any: Placed where the source address is specified; any source matches.

Any: Placed where the destination address is specified; any destination matches.

The effect of line 2 is to permit all traffic that wasn't matched by line 1.

Note that the previous ACL was removed before the new one was applied. A Cisco router interface can have a maximum of two ACLs applied; one outbound and one inbound.



The diagram illustrates that with an extended ACL, packets from the same source can be permitted or denied depending on the destination. The packets destined for 172.50.100.1 are denied by line one of the ACL. The packets destined for all other networks are permitted by line two.

In The Real World...

There are hundreds of port numbers, and while some are common enough to recall easily, others are not. You will not have access to this feature on the CCNA exam, but in real life, use IOS Help to view the available port numbers:

```
R3(config)#access-list 175 deny tcp any any eq ?
<0-65535> Port number
bgp    Border Gateway Protocol (179)
chargen Character generator (19)
cmd    Remote commands (rcmd, 514)
daytime Daytime (13)
discard Discard (9)
domain Domain Name Service (53)
echo   Echo (7)
```

IOS Help will not list every protocol and port number in existence, but will list a number of common and not-so-common port numbers.

Named Access Lists

Named ACLs are just that – rather than using a number to identify them, names are used. Consider a router with 75 ACLs. If the routers are given intuitive names, it can be much easier to see what the author of the list was trying to do.

The syntax of a named ACL is slightly different than the numbered type, but the operation is the same, as is the use of **host** and **any**.

A router with a Serial interface that should allow no traffic from network 175.56.56.0 /24 to leave that interface regardless of destination, but should allow all other traffic, would be configured as follows:

Examining a named access list.

```
R3#conf t  
R3#ip access-list extended NO_NETWORK56_OUT  
R3(config-ext-nacl)#deny ip 175.56.56.0 0.0.0.255 any  
R3(config-ext-nacl)#permit ip any any
```

Line 1:

Ip access-list : Note the “ip” in front of the access-list command. This indicates a named access list.

Extended: The type of named ACL, either extended or standard.

NO_NETWORK56_OUT: The name of the ACL. This should be an intuitive name, one that describes what the list is attempting to do.

Line 2: A “DENY” ACL line denying traffic from source network 175.56.56.0, with any destination.

Line 3: “PERMIT IP ANY ANY” permits IP traffic with any source and any destination.

Applying the named ACL:

```
R3#conf t  
R3(config)#interface serial0  
R3(config-if)#ip access-group NO_NETWORK56_OUT out
```

The named ACL is applied to the interface in the same fashion as a numbered standard or extended list.

Using An ACL To Limit Telnet Access

An ACL can be used to indicate what host or hosts can telnet to a router. The syntax of the ACL is the same, but it is applied in a slightly different manner.

Users telnet to a router by using the *virtual terminal lines*, referred to on the router as vty lines. The ACL is configured, and then applied to the vty lines with the **access-class** command.

Consider a situation where only the user at host 10.17.17.17 should be allowed to telnet to the router, using password CCNA. The ACL is configured and applied as follows:

Examining the configuration of an ACL applied to VTY lines for telnet access.

```
R3#conf t
R3(config)#access-list 78 permit host 10.17.17.17
< Using the host keyword, this ACL permits only host 10.17.17.17, and uses the implicit deny to deny all other users. >

R3#conf t
R3(config)#line vty 0 4
< To configure telnet login, passwords, and the access-class, enter line configuration mode with this command. The 0 and 4 refer to the lines being configured; when used with this command, the numbers represent a range. Here, the range of VTY lines 0 – 4 are being configured at one time. >
R3(config-line)#login
< Login is now allowed. >
R3(config-line)#password CCNA
< The password for login is CCNA. >
R3(config-line)#access-class 78 in
< ACL 78 is applied for incoming telnet access. >
```

Route Summarization

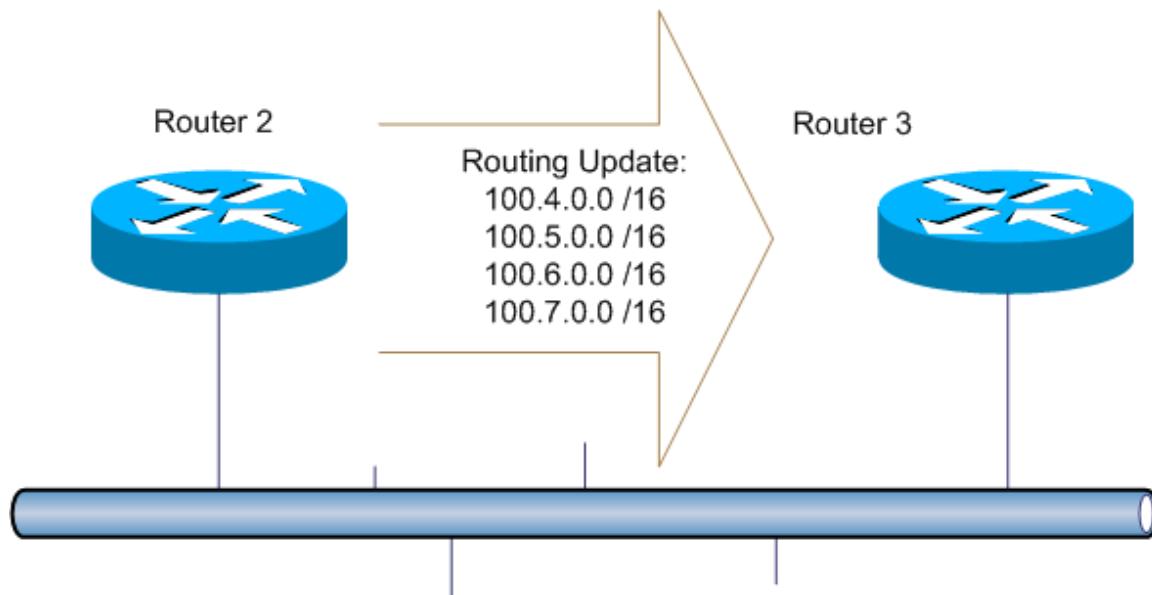
Used primarily in larger networks, route summarization is a technique used to keep the routing tables as compact as possible while keeping an accurate picture of the network topology.

Route summarization offers several advantages. When the router looks for a route to a given destination in the routing table, it will look at all possible routes in search of the longest match for the destination in question. The larger the table, the more time this takes. Large routing tables are also a drain on router memory.

RIP, EIGRP, and OSPF all use different syntax to perform route summarization, but the concept is the same. The first step toward proper route summarization is proper network planning.

Consider a router connected to four different subnets. If the subnets are numerically consecutive, or *contiguous*, the task of route summarization is much simpler. If the subnets are not consecutive, or *non-contiguous*, accurate route summarization may not be possible.

R2 is sending a routing update to R3 for the networks 100.4.0.0 /16, 100.5.0.0 /16, 100.6.0.0 /16, and 100.7.0.0 /16. Without route summarization, R2 will send the four individual routes.



To configure a single route summary for these networks, write the network numbers out in binary math.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
100.4.0.0	01100100	00000100	00000000	00000000
100.5.0.0	01100100	00000101	00000000	00000000
100.6.0.0	01100100	00000110	00000000	00000000
100.7.0.0	01100100	00000111	00000000	00000000

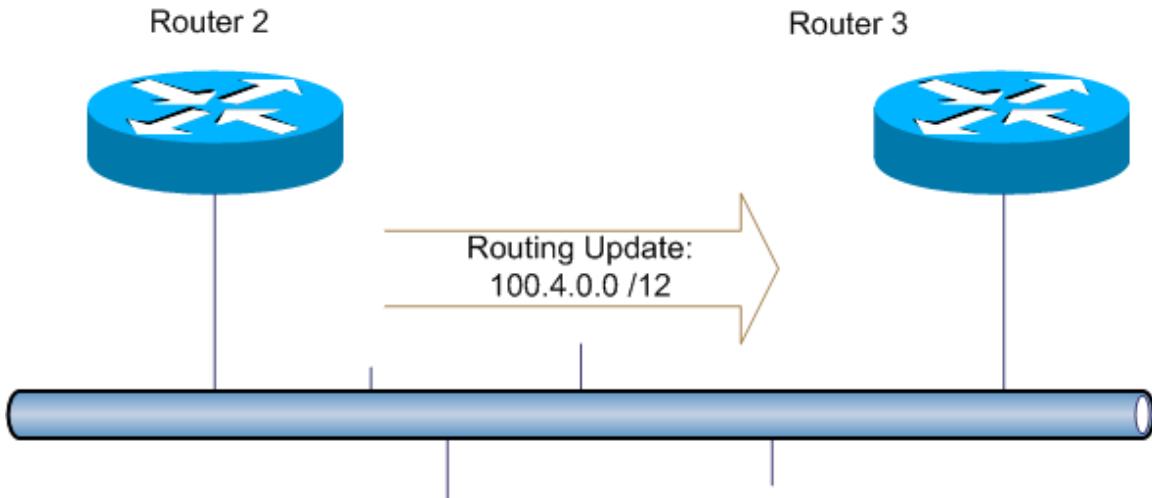
Moving left to right, determine what bits each network number have in common.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
100.4.0.0	01100100	00000100	00000000	00000000
100.5.0.0	01100100	00000101	00000000	00000000
100.6.0.0	01100100	00000110	00000000	00000000
100.7.0.0	01100100	00000111	00000000	00000000

Each of the network has the first 14 bits in common. The resulting network number is the summary for the network. Here, the network summary address is 100.4.0.0.

The subnet mask for the summary must now be determined. The mask is built by putting “1” in for each of the common bits of the summary network number, and “0” for the “don’t care” bits. In this example, the binary mask would be 11111111 11111110 00000000 00000000, resulting in a mask of 255.252.0.0.

The final summary address is 100.4.0.0 255.252.0.0.



Route summarization may include networks that don't exist (yet). The previous example is a "clean" route summarization; it includes only the four named networks and no others. What if the routes to be summarized were 100.1.0.0, 100.2.0.0, 100.3.0.0, and 100.4.0.0?

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
100.1.0.0	0110100	00000001	00000000	00000000
100.2.0.0	0110100	00000010	00000000	00000000
100.3.0.0	0110100	00000011	00000000	00000000
100.4.0.0	0110100	00000100	00000000	00000000

The four networks have 13 bits in common, left to right. The resulting network number is the summary for the network, which is 100.0.0.0. The subnet mask is determined by putting "1" in for each common bit and "0" for the rest, resulting in a mask of 255.248.0.0. The final summary address is 100.0.0.0 255.248.0.0.

This address and mask are accurate for the four networks given. A routing issue could arise because this summary also includes networks 100.5.0.0, 100.6.0.0, and 100.7.0.0:

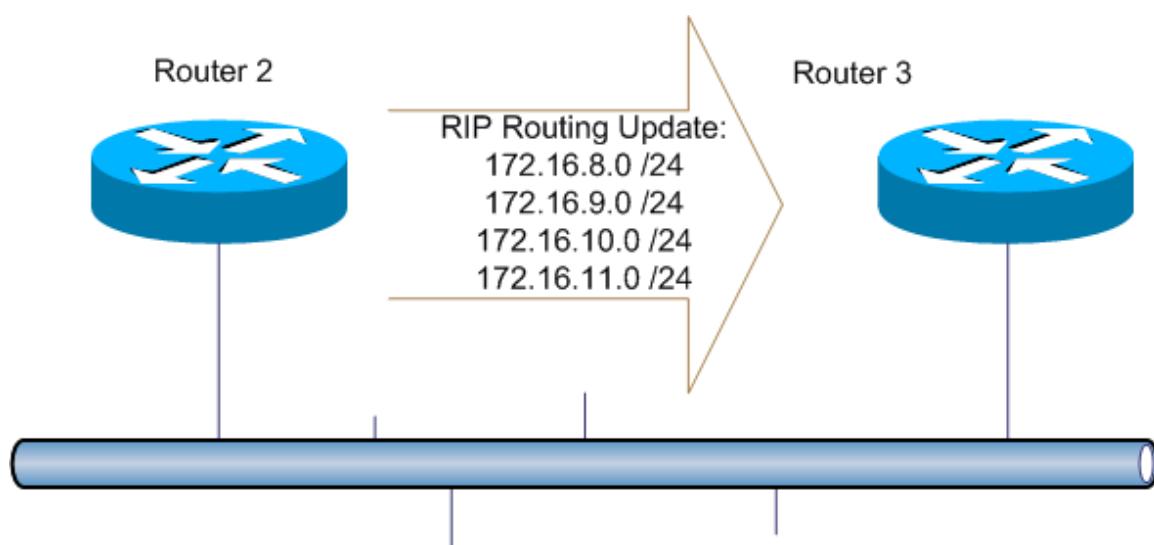
	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
100.1.0.0	0110100	00000001	00000000	00000000
100.2.0.0	0110100	00000010	00000000	00000000
100.3.0.0	0110100	00000011	00000000	00000000
100.4.0.0	0110100	00000100	00000000	00000000
100.5.0.0	0110100	00000101	00000000	00000000
100.6.0.0	0110100	00000110	00000000	00000000
100.7.0.0	0110111	00000111	00000000	00000000

This does not make the summary address wrong; it does allow for routing problems if these additional three networks were placed elsewhere on the network. Once the summary address and mask have been determined, write out the next consecutive (contiguous) network number to see if the summary address will also represent that address. If so, be aware of the potential routing issues of using that network number in another section of the network.

Route Summarization With RIP

RIP uses the interface-level command **ip summary-address rip** to configure summary addresses.

R2 and R3 are running RIP version 2. R2 is advertising the following networks to R3: 172.16.8.0 /24, 172.16.9.0 /24, 172.16.10.0 /24, and 172.16.11.0 /24.



To summarize these networks, convert them from dotted decimal to binary, and determine the bits the networks have in common from left to right:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
172.16.8.0	10101100	00010000	00001000	00000000
172.16.9.0	10101100	00010000	00001001	00000000
172.16.10.0	10101100	00010000	00001010	00000000
172.16.11.0	10101100	00010000	00001011	00000000

The networks have 22 bits in common. The resulting network number is 172.16.8.0. Using 1s for the in-common bits and 0s for the rest, the resulting subnet mask is 11111111 11111111 11111100 00000000, which in dotted decimal is 255.255.252.0.

The summary address is 172.16.8.0 255.255.252.0. With RIP, advertise the summary address with the interface-level command **ip summary-address rip**.

Configuring RIP route summarization.

R2 is running RIP version 2 and is advertising four separate routes to R3.

```
R2(config-if)#router rip  
R2(config-router)#version 2  
R2(config-router)#no auto  
R2(config-router)#network 172.16.0.0
```

On R3, the four routes all show as separate RIP routes.

```
R3#show ip route rip  
    172.16.0.0/24 is subnetted, 4 subnets  
R    172.16.8.0 [120/1] via 172.23.23.2, 00:00:02, Ethernet0  
R    172.16.9.0 [120/1] via 172.23.23.2, 00:00:02, Ethernet0  
R    172.16.10.0 [120/1] via 172.23.23.2, 00:00:02,  
Ethernet0  
R    172.16.11.0 [120/1] via 172.23.23.2, 00:00:02,  
Ethernet0
```

Using the summary route derived from the last exercise, the **ip summary-address rip** command is configured on the serial interface.

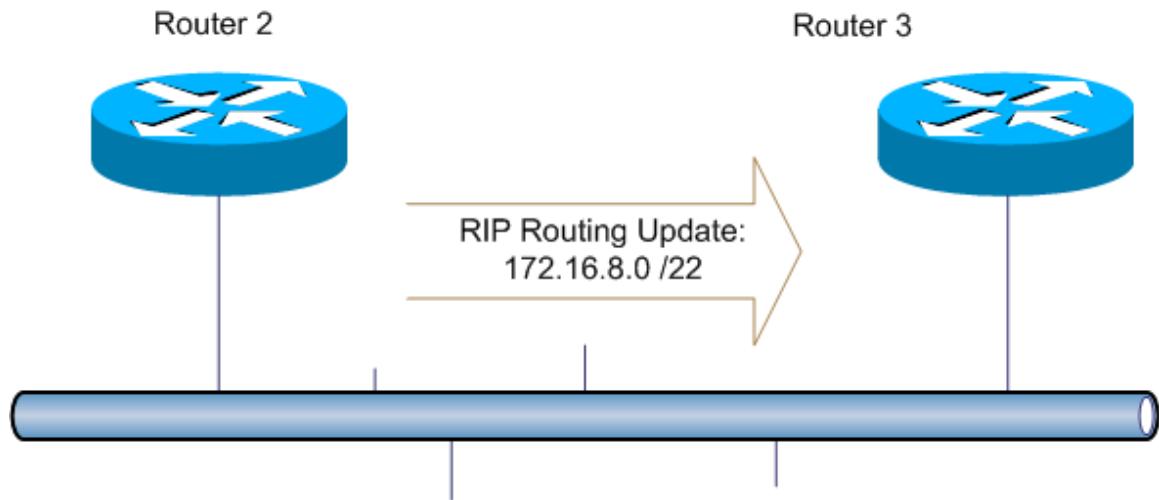
```
R2#conf t  
R2(config)#int ethernet0  
R2(config-if)#ip summary-address rip 172.16.8.0 255.255.252.0
```

After clearing the routing table, R3 now shows one summary address for the four routes.

Note the /22 mask.

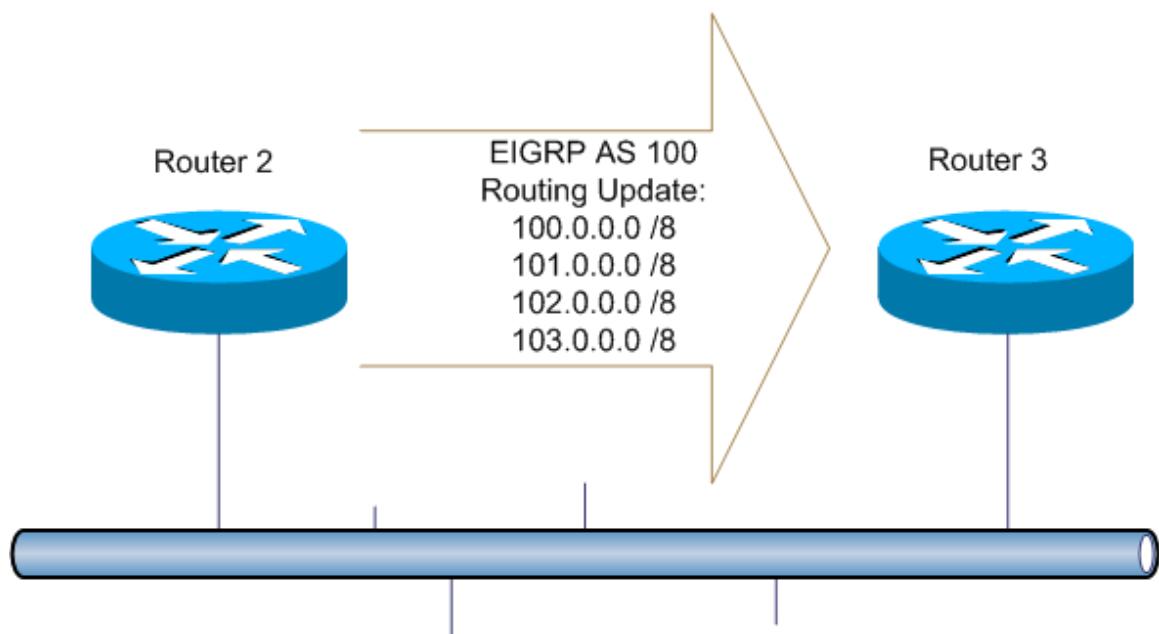
```
R3#clear ip route *  
R3#show ip route rip  
    172.16.0.0/22 is subnetted, 1 subnets  
R    172.16.8.0 [120/1] via 172.23.23.2, 00:01:24, Ethernet0
```

R2 now sends a single route to R3.



Route Summarization With EIGRP

EIGRP uses an interface-level command for route summarization as well. R2 is advertising four routes to R3 via EIGRP: 100.0.0.0 /8, 101.0.0.0 /8, 102.0.0.0 /8, and 103.0.0.0 /8.



R3's routing table contains the four routes:

```
R3#show ip route eigrp
D 100.0.0.0/8 [90/2297856] via 172.23.23.2, 00:04:23, Ethernet0
D 101.0.0.0/8 [90/2297856] via 172.23.23.2, 00:04:23, Ethernet0
D 102.0.0.0/8 [90/2297856] via 172.23.23.2, 00:04:23, Ethernet0
D 103.0.0.0/8 [90/2297856] via 172.23.23.2, 00:04:23, Ethernet0
```

To summarize the networks, convert them from dotted decimal to binary, and determine how many in-common bits exist from left to right.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
100.1.0.0	01100100	00000001	00000000	00000000
101.1.0.0	01100101	00000001	00000000	00000000
102.1.0.0	01100110	00000001	00000000	00000000
103.1.0.0	01100111	00000001	00000000	00000000

The networks have six bits in common, resulting in the network number 100.0.0.0. The summary network mask is determined by placing 1s in the mask for the in-common bits and 0s for the rest. The binary mask is 11111100 00000000 00000000 00000000, which in dotted decimal is 252.0.0.0

The summary address and mask have been determined to be 100.0.0.0 252.0.0.0. The interface-level command **ip summary-address eigrp** will summarize the routes.

Configuring route summarization in EIGRP.

```
R2#conf t  
R2(config)#interface ethernet0  
R2(config-if)#ip summary-address eigrp 100 100.0.0.0 252.0.0.0
```

*The routes are summarized with the **ip summary-address eigrp** interface-level command. The four more-specific routes will be removed from the neighbor routing tables, and a single summary entry will take its place. Note that the EIGRP AS is referred to in the summary command.*

R3#show ip route eigrp

```
D 100.0.0.0/6 [90/2297856] via 172.23.23.2, 00:02:33, Ethernet0
```

Determining and Configuring Variable-Length Subnet Masks

Deciding on what VLSM to use is simply a matter of determining how many host bits will be needed for the required number of hosts, and charting the VLSM configuration as this is done for each network.

It Bears Repeating...Why Subtract Two?

The two subnets that will not be used in your CCNA exam preparation are the “zero subnet” (all binary zeroes) and the “broadcast subnet” (all binary ones).

Both the “all-zeroes” and “all-ones” subnets are available for use on a Cisco router, but Cisco recommends you not use them. For exam purposes, use the formula as shown and do not consider either of these networks to be valid.

Consider a network given the network 192.168.56.0. There will be five subnets that require the following number of hosts:

	Hosts Required
Network A	2
Network B	15
Network C	30
Network D	48

It's good practice on both the CCNA exam and the job to chart subnets as they are created. In accordance with the CCNA exam, “subnet

zero" will be considered invalid and will not be used. It does have to be accounted for, though. The subnet mask /30 gives us the smallest amount of hosts, two, so use that mask to get rid of subnet zero with the fewest addresses wasted.

The subnet number must have a zero in every bit that the subnet mask does. These are the host bits. When configuring VLSM, the question to ask is, "What is the smallest unused network number that can be created **when all host bits are 0?**"

The picture becomes much clearer when the subnet number and /30 mask are converted to binary:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.0	11000000	10101000	00111000	00000000
/30 Subnet Mask	11111111	11111111	11111111	11111100

Only the last two bits of the subnet 192.168.56.0 have to be "0" in order to match the subnet mask. The question "What is the smallest unused network number that can be created when all host bits are 0?" is answered with the address 192.168.56.0.

It's good practice to keep a chart of VLSMs **as they are created** to ensure overlapping does not take place.

Subnet	Network / Mask	Network Address	Valid Hosts	Broadcast Address
< subnet zero, not in use >	192.168.56.0 / 30	56.0	56.1, 56.2	56.3

Network A requires two hosts, so again two host bits will be needed. The same chart previously drawn is used again:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.0	11000000	10101000	00111000	00000000
/30 Subnet Mask	11111111	11111111	11111111	11111100

Again, only the last two bits of the network number have to be zero. The question "What is the smallest unused network number that can be created while keeping the host bits at 0?" has a different answer, though, since 192.168.56.0 has already been used. By changing the last network bit (the 6th bit of the final octet) to "1", it yields network

number 192.168.56.4, and the last two bits of that network number are "0" as well:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.4	11000000	10101000	00111000	000001 00
/30 Subnet Mask	11111111	11111111	11111111	111111 00

This is the network number and subnet mask to use for Network A, and is added to the VLSM table:

Subnet	Network / Mask	Network Address	Valid Hosts	Broadcast Address
< subnet zero, not in use >	192.168.56.0 / 30	56.0	56.1, 56.2	56.3
Network A	192.168.56.4 / 30	56.4	56.5, 56.6	56.7

Network B requires 15 host addresses, which requires five host bits. (2 to the 5th power is 32, yielding 30 host addresses. 2 to the 4th power is 16, but subtracting the network address and broadcast address only leaves 14 valid host addresses.)

Convert the network number and a /27 mask to binary:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.0	11000000	10101000	00111000	000 00000
/27 Subnet Mask	11111111	11111111	11111111	111 00000

The question: "What is the smallest unused network number that can be created while leaving the host bits at 0?"

The answer: The smallest value that can be created with the network portion of the 4th octet (the first three bits) is 000, but that would yield 192.168.56.0, which has already been used. A binary value of 001 would yield 192.168.56.32, which has not yet been used. Use this network and a /27 subnet mask for Network B.

Subnet	Network / Mask	Network Address	Valid Hosts	Broadcast Address
< subnet zero,	192.168.56.0 / 30	56.0	56.1, 56.2	56.3

not in use >				
Network A	192.168.56.4 / 30	56.4	56.5, 56.6	56.7
Network B	192.168.56.32 / 27	56.32	56.33 – 56.62	56.63

Network C requires 30 host addresses; a mask of /27 will result in exactly that many host addresses. Refer to the chart of the network number and the /27 bit subnet mask:

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.0	11000000	10101000	00111000	00000000
/27 Subnet Mask	11111111	11111111	11111111	11100000

The question “What is the smallest **unused** network number that can be created while leaving the host bits at 0?” now has a different answer, since network 192.168.56.32 was just used. The next-smallest value that can be created with the network portion of the network number is 010 in binary, or 64 in decimal, yielding the network number 192.168.56.64. Use this network with a /27 bit mask for Network C.

Subnet	Network / Mask	Network Address	Valid Hosts	Broadcast Address
< subnet zero, not in use >	192.168.56.0 / 30	56.0	56.1, 56.2	56.3
Network A	192.168.56.4 / 30	56.4	56.5, 56.6	56.7
Network B	192.168.56.32 / 27	56.32	56.33 – 56.62	56.63
Network C	192.168.56.64 / 27	56.64	56.65 – 56.94	56.95

Network D requires 48 host addresses; six host bits will be needed. (Two to the sixth power is 64, leaving 62 valid host addresses.) Compare the network number to a subnet mask of /26.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
Network 192.168.56.0	11000000	10101000	00111000	00000000
/26 Subnet Mask	11111111	11111111	11111111	11000000

The question: "What is the smallest **unused** network number that can be created while leaving the host bits at zero?" There are only two network bits in the 4th octet, representing 128 and 64 from left to right. The network number 64 is used by Network C; the next smallest possible value is 128. Use this network for Network D, along with a subnet mask of /26.

Subnet	Network / Mask	Network Address	Valid Hosts	Broadcast Address
< subnet zero, not in use >	192.168.56.0 / 30	56.0	56.1, 56.2	56.3
Network A	192.168.56.4 / 30	56.4	56.5, 56.6	56.7
Network B	192.168.56.32 / 27	56.32	56.33 – 56.62	56.63
Network C	192.168.56.64 / 27	56.64	56.65 – 56.94	56.95
Network D	192.168.56.128 / 26	56.128	56.128 – 56.254	56.255

The VLSM configuration is complete, and the single network 192.168.56.0 has been used to configure an entire network's IP addressing.

In The REAL World...

Do not rely on any VLSM methodologies that involve memorizing charts. Use your understanding of binary math to configure VLSMs. Memorization does nothing for understanding what is actually going on, and tables memorized for an exam fade quickly after the exam is taken. When asked to configure VLSMs in a job interview or on the job, a chart memorized weeks or months ago is not going to leap to mind. An understanding of binary math is essential for success in any CCNP or CCIE course as well. Develop this skill now and it will serve you well on exams and on the job.

A true IT professional is not in the memorization business; a true IT professional is in the knowledge business.

IP Address Conservation

With a finite number of IP addresses and an ever-expanding Internet, steps must be taken to conserve these addresses. Cisco routers offer several methods of IP address conservation.

Classless InterDomain Routing

Classless InterDomain Routing (CIDR) is an IP address conservation methodology used by Internet Service Providers (ISP). ISPs will assign a subnet of a major network number rather than the entire number. CIDR also allows the ISP to perform route summarization, where a single summary route will represent many networks.

CIDR delivers a dual benefit by saving IP addresses and allowing for smaller Internet routing tables due to the route summarization.

RFC 1918 Private Addresses

If a network is not going to connected to the Internet, it would be acceptable to use duplicates of network numbers that are already registered. Logically, any number could be selected, but RFC 1918 defines a series of networks that are not used by *any* public networks (that is, networks connected to the Internet).

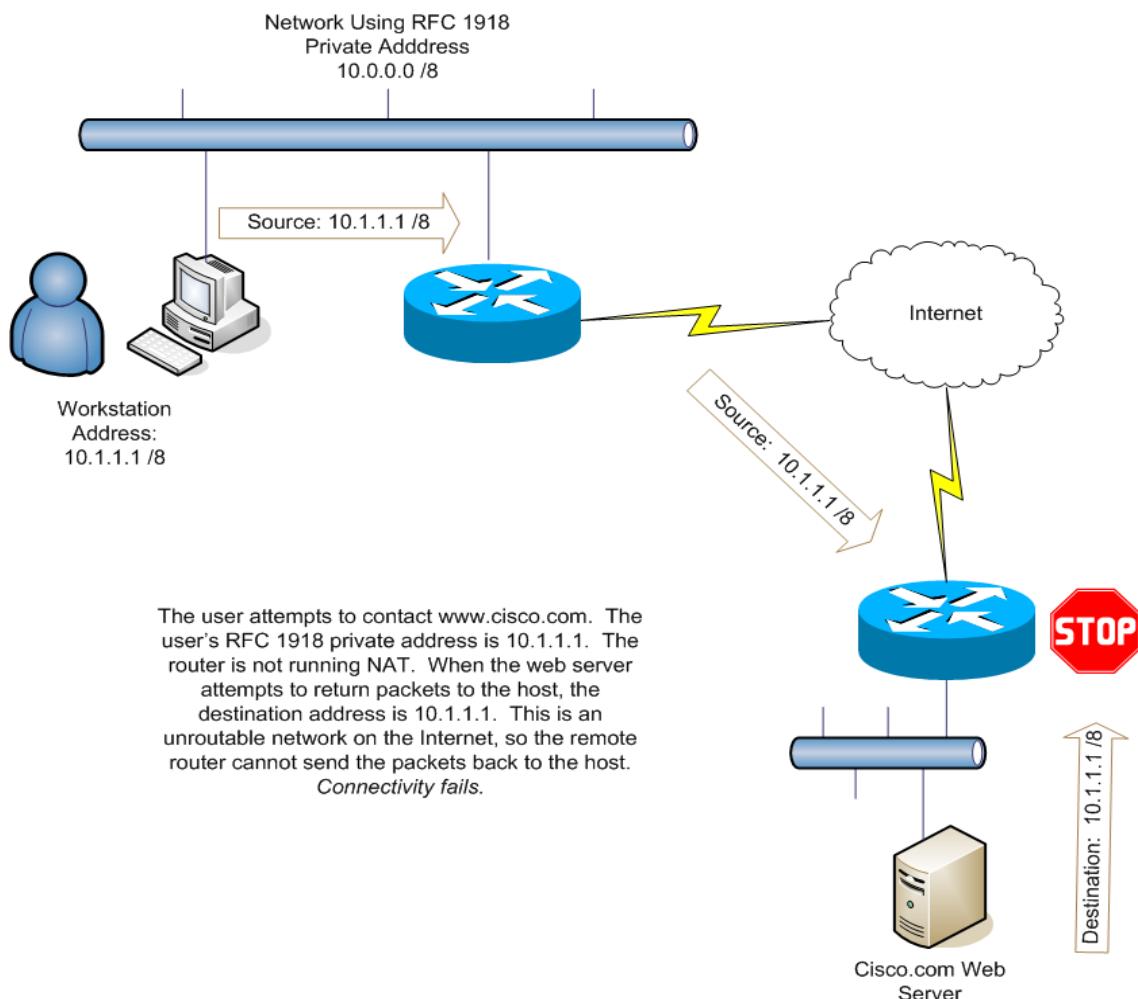
The RFC 1918 Private Addresses

Class A	10.0.0.0 / 8
Class B	172.16.0.0 / 12
Class C	192.168.0.0 /16

For both exam success and real-world success, you have to know these by heart.

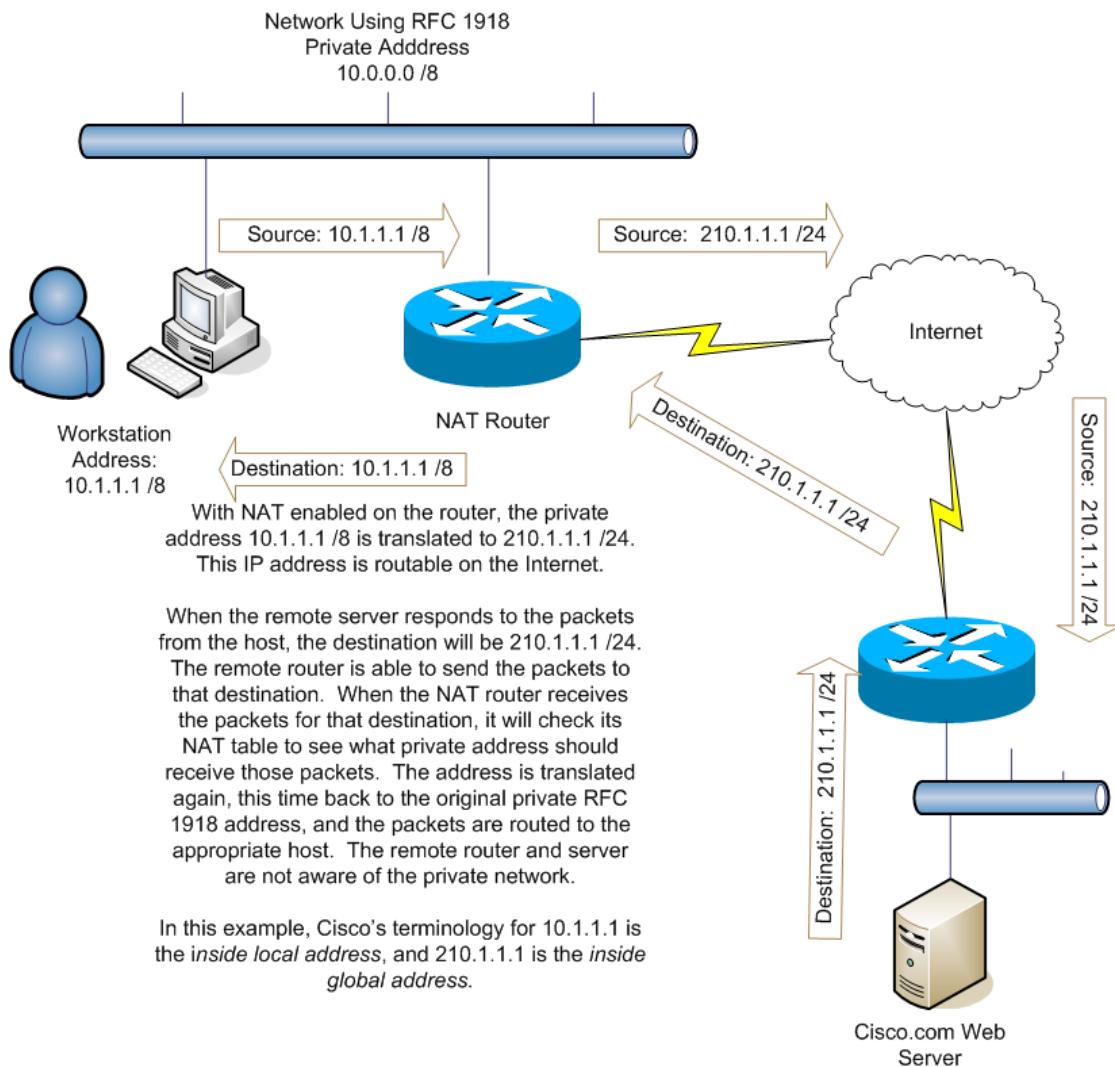
Network Address Translation

Using private addresses is fine until a host using a private address wants to communicate with a device on the Internet. Consider what happens if a workstation with a private IP address attempts to contact www.cisco.com. Cisco's web server would receive a packet from a host with a source address on an RFC 1918 network. How would the server know how to respond to the private address if it's not used anywhere on the internet?



The user attempts to contact www.cisco.com. The user's RFC 1918 private address is 10.1.1.1. The router is not running NAT. When the web server attempts to return packets to the host, the destination address is 10.1.1.1. This is an unrouteable network on the Internet, so the remote router cannot send the packets back to the host.
Connectivity fails.

Network Address Translation (NAT) resolves this issue. As the packet leaves the router that will send it to the Internet host, the private address is removed and a valid IP address for Internet communication replaces it. The destination host never sees the private address, only the public one. When the packet returns, the NAT-enabled router will remove the public IP address, replace it with the previous private IP address, and the packet is routed back to the host. The private address is never seen on the Internet.



Static NAT

If a limited number of hosts on a private network need Internet access, *static NAT* may be the appropriate choice. Static NAT maps a private address to a public one.

There are three internal PCs on an RFC 1918 private network, using addresses 10.5.5.5, 10.5.5.6, and 10.5.5.7. The router's Ethernet0 interface is connected to this network, and the Internet is reachable via the Serial0 interface. The IP address of the Serial network is 210.1.1.1 /24, with all other addresses on the 210.1.1.0/24 network available.

Three static mappings are needed to use Static NAT. The interfaces must be configured for NAT as well.

Configuring the interfaces for Network Address Translation. The Ethernet network is the “inside” network; the Serial interface leading to the Internet is the “outside” network.

```
R3(config)#interface ethernet0
R3(config-if)#ip address 10.5.5.100 255.0.0.0
R3(config-if)#ip nat inside
R3(config-if)#interface serial0
R3(config-if)#ip address 210.1.1.1 255.255.255.0
R3(config-if)#ip nat outside
```

The static mappings are created and verified.

```
R3#conf t
R3(config)#ip nat inside source static 10.5.5.5 210.1.1.2
R3(config)#ip nat inside source static 10.5.5.6 210.1.1.3
R3(config)#ip nat inside source static 10.5.5.7 210.1.1.4
```

R3#show ip nat translations

Pro Inside global	Inside local	Outside local	Outside global
---	10.5.5.5	---	---
---	10.5.5.6	---	---
---	10.5.5.7	---	---

“show ip nat translations” displays the mappings. Note the terms “inside global” and “inside local”.

R3#show ip nat statistics

Total active translations: 3 (3 static, 0 dynamic; 0 extended)

Outside interfaces: Serial0

Inside interfaces: Ethernet0

Hits: 0 Misses: 0

Expired translations: 0

“show ip nat statistics” displays the number of static and dynamic mappings.

Dynamic NAT

Static NAT is fine for a few hosts, but consider a private network with 150 hosts. It would be unwieldy at best to have 150 static NAT statements on your router.

Dynamic NAT allows a pool of public IP addresses to be created. The public IP addresses are mapped to a private address as needed, and the mapping is dropped when the communication ends.

Like Static NAT, Dynamic NAT requires the interfaces connected to the Internet and the private networks be configured with "ip nat outside" and "ip nat inside", respectively.

Using the previous network example, R3 is now configured to assign an address from a NAT pool to these three network hosts as needed:

Examining the configuration of Dynamic NAT.

```
R3#conf t  
R3(config)#interface ethernet0  
R3(config-if)#ip nat inside  
R3(config-if)#interface serial0  
R3(config-if)#ip nat outside
```

As with Static NAT, the inside and outside interfaces are identified.

```
R3#conf t  
R3(config)#ip nat inside source list 1 pool NATPOOL  
R3(config)#ip nat pool NATPOOL 200.1.1.2 200.1.1.5 netmask 255.255.255.0
```

*An access-list will be used to identify the hosts that will have their addresses translated by NAT. The **nat inside source** command calls that list and then names the NAT pool to be used.*

The next line of the config defines the pool, named NATPOOL. The two addresses listed are the first and last addresses of the pool, meaning that 200.1.1.2, 200.1.1.3, 200.1.1.4, and 200.1.1.5 are in the pool, all using a mask of 255.255.255.0. Take care not to include the serial address of the NAT router in the pool

```
R3#conf t  
R3(config)#access-list 1 permit 10.5.5.0 0.0.0.255
```

The access list permits all hosts on 10.5.5.0/24, meaning that any host on that subnet can use an address from the NAT pool to communicate with Internet-based hosts.

Show ip nat statistics will display the name and configuration of the NAT pool.

```
R3#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces: Serial0
Inside interfaces: Ethernet0
Hits: 0 Misses: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list 1 pool NATPOOL refcount 0
pool NATPOOL: netmask 255.255.255.0
    start 200.1.1.2 end 200.1.1.5
    type generic, total addresses 4, allocated 0 (0%), misses 0
```

Four addresses are available in the NAT pool. What if the network has 50 hosts and ten of them want to connect to an Internet host simultaneously?

NAT allows multiple hosts to use the same public IP address via *Port Address Translation (PAT)*. Generally referred to as “overloading”, the private address and port number will be translated to a public address *and port number*, allowing the same IP address to support multiple hosts. The router will differentiate the connections by using a different port number for each translation, even though the same IP address will be used.

Port Address Translation is simple to configure. Instead of referring to a NAT pool with the **ip nat inside source** command, name the outside interface followed by the word “overload”.

Examining the configuration of Port Address Translation (PAT)

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface ethernet0
R3(config-if)#ip nat inside
R3(config-if)#interface serial0
R3(config-if)#ip nat outside
R3(config-if)#ip nat inside source list 1 interface serial0 overload
R3(config)#access-list 1 permit 10.5.5.0 0.0.0.255
```

“overload” indicates that the IP address of the named interface will be the only one used for NAT, but that a different port number will be used for each translation, allowing the router to keep the different translations separate while using only a single IP address.

Telnet

Telnet is a protocol used for accessing remote networking devices. It allows a computer to act as a remote terminal.

To telnet to a remote device from a Cisco router, use the **telnet** command followed by the IP address of the device:

```
R2#telnet 172.12.123.1  
Trying 172.12.123.1 ... Open
```

Password required, but none set

```
[Connection to 172.12.123.1 closed by foreign host]
```

A common Telnet error is the omission of a password on the virtual terminal lines (VTY lines). Without this password, no user will be able to telnet to the device, and the message "password required, but not set" will be displayed to the user attempting to telnet to the device.

Configuring R1 to allow telnet access.

```
R1#conf t  
R1(config)#line vty 0 4  
R1(config-line)#login  
R1(config-line)#password CCNAPASSED
```

The password CCNAPASSED has been set. With this configuration, any user who knows this password can telnet into the router. This is a very basic password scheme and is not recommended for a production network.

R2 attempts to telnet to R1 again:

```
R2#telnet 172.12.123.1  
Trying 172.12.123.1 ... Open  
User Access Verification
```

Password: < CCNAPASSED was entered here. Passwords do not appear on the screen as they are entered. >

R1>en

% No password set

*< Users are placed into user exec mode by default. No **enable secret** or **enable password** has been set, so the users cannot enter privileged exec mode. >*

Either an **enable password** or **enable secret** must be set on a router to allow a user telnetting to that router to enter privilege exec mode.

```
R1#conf t  
R1(config)#enable secret CISCO
```

An enable secret password is configured on R1.

```
R2#telnet 172.12.123.1  
Trying 172.12.123.1 ... Open  
User Access Verification
```

```
Password: < VTY line password “CCNAPASSED” is entered here. >  
R1>en < User attempts to enter privilege exec mode. >  
Password: <Enable secret password “CISCO” is entered here. >  
R1# < User is in privileged exec mode. >
```

To end a telnet session, type exit.

```
R1#exit  
[Connection to 172.12.123.1 closed by foreign host]  
R2#  
The user ends the Telnet session and is back at the R2 prompt.
```

In The REAL World...

A Telnet password scheme of placing a single password on the VTY lines is not a secure one. A series of usernames and passwords can be configured, where individual users have their own passwords. (This series of usernames and passwords is referred to as the “local database”.) Users telnetting into the router can automatically be placed in privilege exec mode as well, bypassing the need for a global enable secret password.

```
R1#conf t  
R1(config)#username CBRYANT privilege 15 password CCIE  
R1(config)#username DTRUMP password FIRED  
R1(config)#username HKRAVIS password NABISCO  
R1(config)#line vty 0 4  
R1(config-line)#login local
```

A local database of authorized users is created with the **username / password** command. A different password is assigned to each user. In VTY line configuration mode, the command **login local** indicates that the local database will be used for telnet authorization.

No password is configured for the VTY lines when using the local database.

In the first username / password command, **privilege 15** is included. When this user successfully authenticates, that user will be placed into privilege exec mode automatically, without having to enter the enable secret password. The other two users will be put into user exec mode and will need to know the enable secret password to enter privilege exec mode.

Users who are not listed in the local database will be unable to telnet to the router.

Examining the results of the previous configuration.

```
R2#telnet 172.12.123.1
Trying 172.12.123.1 ... Open
User Access Verification
Username: CBRYANT
Password:
R1#
```

User CBRYANT was able to telnet into the router and was placed into privilege mode automatically due to the “privilege 15” command in the username / password config.

```
R2#telnet 172.12.123.1
Trying 172.12.123.1 ... Open
User Access Verification

Username: DTRUMP
Password:
R1>
```

User DTRUMP can telnet into the router, but is placed into user exec mode upon authenticating. The user will need the enable secret password to enter privilege exec mode.

```
R2#telnet 172.12.123.1
Trying 172.12.123.1 ... Open
User Access Verification
```

```
Username: RJOHNSON
% Login invalid
```

User RJOHNSON is not listed in the local database on R1 and cannot authenticate.

There are multiple ways to authenticate remote users, which are covered in more detail in the CCNP and CCIE curriculum. For now, know the above method of authenticating individual users and how to place them into privilege exec mode without having to enter the enable secret password.

Using and Resolving Hostnames For Remote Access

Hostnames are configured on Cisco devices with the global command **hostname**. Rather than type the full IP address in a Telnet command, the hostname can be used.

To do so, the router must have a method of resolving the hostname to the desired IP address. There are two ways of doing so. The first is the global command **ip host**.

Building an IP Host Table

The global command **ip host** is followed by that device's IP address.

Building an IP Host table and using a hostname to telnet to R1.

```
R2#conf t  
R2(config)#ip host R1 172.12.123.1
```

```
R2#R1  
Trying R1 (172.12.123.1)... Open  
User Access Verification
```

```
Username: CBRYANT  
Password:  
R1#
```

*The **ip host** command resolves the hostname R1 to the IP address 172.12.123.1. When telnetting to R1, only “R1” is entered; if no command is specified, the router assumes an attempt to connect with a remote host is being made.*

Using DNS To Resolve Hostnames

A DNS server can be used to resolve hostnames as well. The router will need to be informed of the IP address of the DNS server with the **ip name-server** command.

When a command is mistyped on a Cisco router, the default behavior of the router is to attempt to resolve it via DNS. To prevent this behavior, enter the global command **no ip domain-lookup**. Of course, to use DNS to resolve hostnames, **ip domain-lookup** would have to be reenabled if it's been turned off.

Examining the behavior of a Cisco router involving DNS.

R2#contin

Translating "contin"...domain server (255.255.255.255)

% Unknown command or computer name, or unable to find computer address

A command is mistyped as “contin”. The Cisco router’s default behavior is to resolve this unknown command via DNS. A DNS server location has not been configured, so the router broadcasts for a resolution, which does not come. The DNS lookup attempt must time out before the configuration can continue.

R2#conf t

R2(config)#**no ip domain-lookup**

R2#contin

Translating "contin"

% Unknown command or computer name, or unable to find computer address

With “no ip domain-lookup” configured, the router doesn’t attempt a DNS resolution. It sees there is no local resolution configured and almost immediately sends a message to the console that the name can’t be resolved.

R2#conf t

R2(config)#**ip domain-lookup**

R2(config)#**ip name-server 10.1.1.1**

R2#contin

Translating "contin"...domain server (10.1.1.1)

A DNS server is installed on the network with the IP address 10.1.1.1. DNS lookup is reenabled with the command **ip domain-lookup**, and the IP address of the DNS server is specified with the **ip name-server** command.

Manipulating The Configuration Register To Recover Passwords

WARNING: Do not manipulate the configuration register unless you are SURE of the effect.

The process for recovering passwords varies from device to device. Refer to www.cisco.com/univercd for password recovery specifics for a particular Cisco device.

The password recovery method examined here is for 2500 and 2600 routers.

An engineer who finds themselves locked out of a router can view and change the password by changing the configuration register.

The router must first be rebooted and a “break” performed when the router is booting up. This break sequence can also vary depending on what program is used to access the router, but <CTRL- BREAK> is the usual key combination.

The router will now be in ROM Monitor mode. From the rom monitor prompt, change the default configuration register of 0x2102 to 0x2142 with the global **config-register** command. Reload the router with the letter **I** on a 2500 or the **reset** command with a 2600.

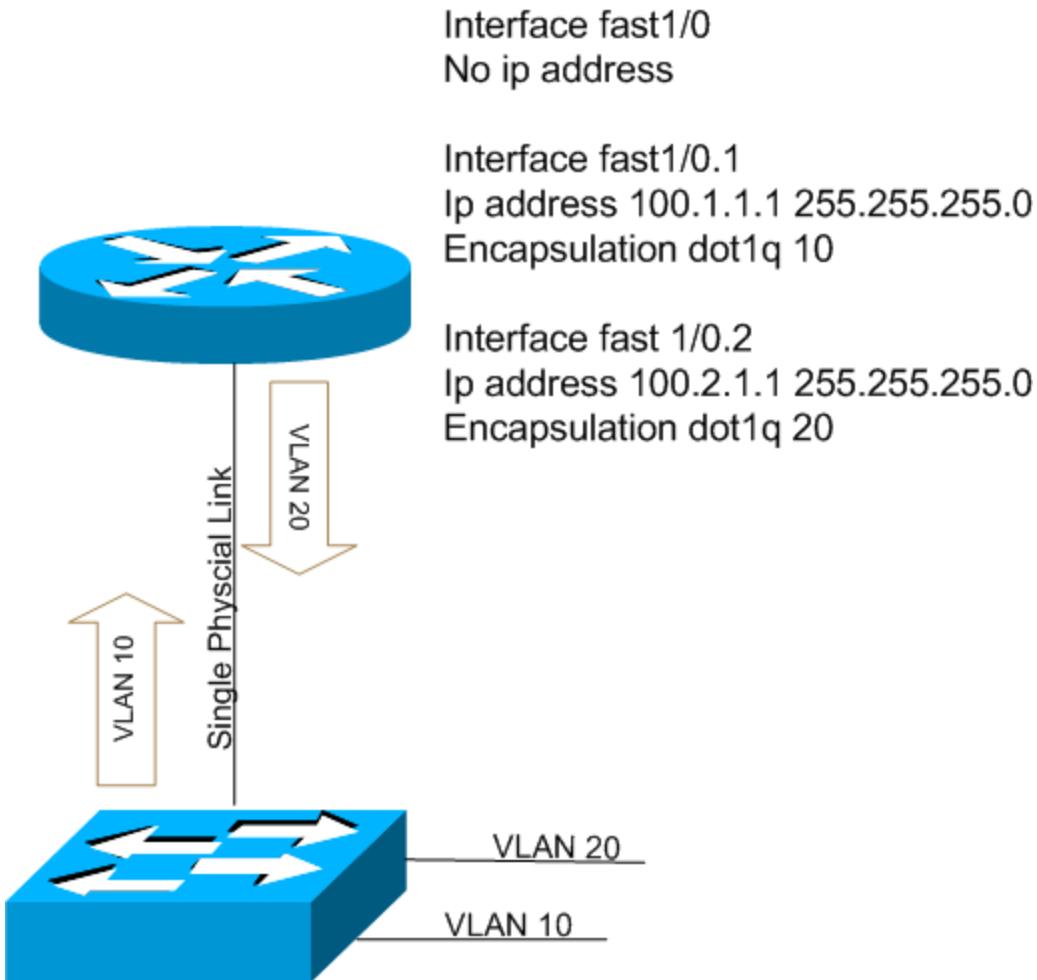
When the router reloads, you’ll be prompted to enter Setup mode. Answer “N”, copy the running configuration to the startup configuration, and set a new enable secret password.

Set the configuration register back to 0x2102 and reload the router.

Running A Trunk Line Between A Router And A Switch

As mentioned in Section One, a trunk line allows traffic from multiple VLANs to flow over interconnected switches. Traffic from multiple VLANs can also be configured to run over a single physical link between a router and a switch. This technique is referred to in the real world as “router on a stick”.

Consider this example: A single switch contains two VLANs. VLAN 10 has hosts in 100.1.1.0 /24, and VLAN 20 has hosts in 100.2.1.0 /24. By connecting this single switch to a Fast Ethernet port on a Cisco router, and configuring “router on a stick” on that Fast Ethernet port, routing between the two VLANs can take place over a single physical link.



Note that the physical Fast Ethernet interface has no IP address. The interface is broken up into logical interfaces, one per VLAN that will be routed. Interface fast1/0.1 is given an IP address in VLAN 10, and fast1/0.2 is given an IP address in VLAN 20. The encapsulation type is then specified, and the VLAN number that interface is part of is then named. Hosts in the two VLANs can now communicate with each other via the directly connected router.

In The REAL World...

With the evolving capability of switches to actually perform routing as well, “router on a stick” is not used as often as it once was. However, it’s still an important capability to know.

The 2950 switch, currently tested in the CCNA exam, is not capable of routing.

Maximum Transmission Units

The Maximum Transmission Unit, or MTU, is the largest an IP packet can be before it is fragmented into smaller parts.

The default MTU of an Ethernet port is 1500. If a router receives a packet larger than this, the router will fragment the packet and send the fragments toward their destination. Any routers further down the same path will continue to route these fragments as they normally would. The reassembly is done by the end host.

FTP and TFTP

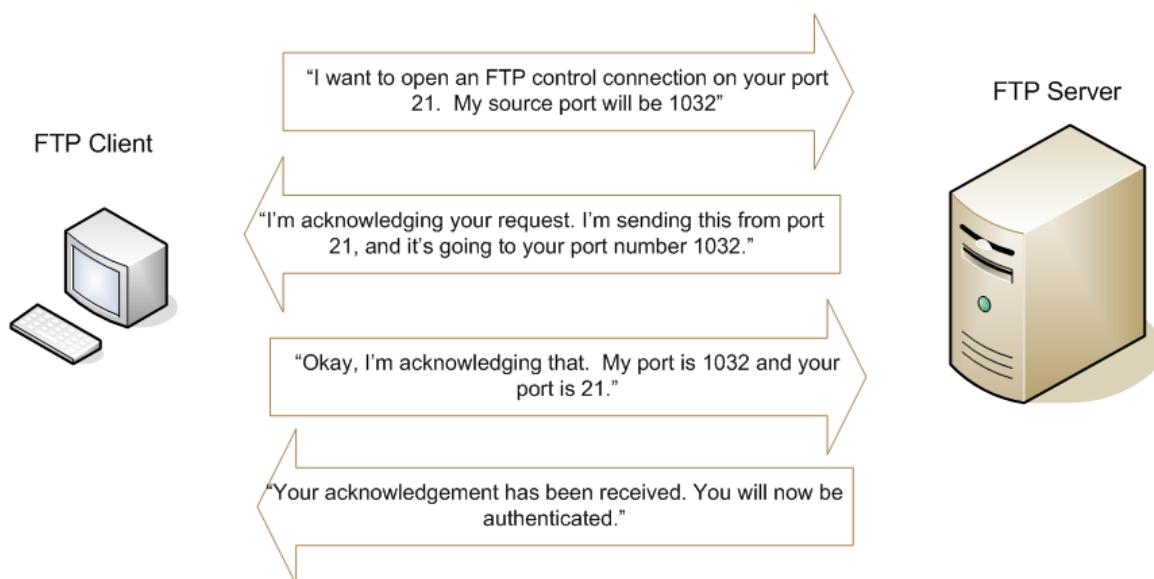
File Transfer Protocol (FTP), and its less robust companion, Trivial File Transfer Protocol (TFTP), are used to do just that – transfer files.

Your typical end users will generally use FTP for two reasons. First, it does give more options. Secondly, there are several programs available that make FTP very easy to use.

FTP is TCP-based, and is connection-oriented. When the end user wants to connect with an FTP server, a TCP connection is established using the FTP server's well-known port number, 21.

The source port used by the FTP Client will be a unused port number greater than 1024. There is usually some kind of simple authentication required for connection, and users are generally allowed only to gain access to certain files rather than all files on the FTP server.

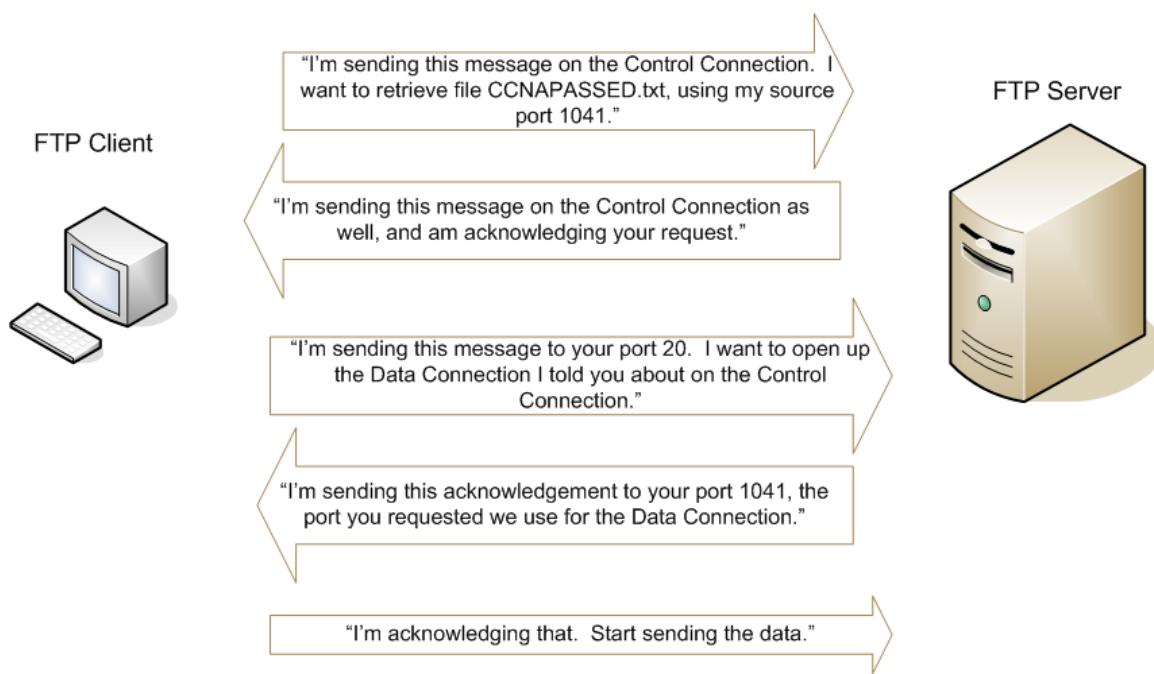
The FTP Control Connection Process.



The FTP client will actually have two separate connections to the FTP server. Remember how ISDN has bearer channels and data channels? FTP clients will have a *control connection*, which is the connection responsible for the initial request for communication, and any authentication schemes. When the FTP client actually requests a file, a separate connection is opened for the file transfer. This is the *data connection*.

The FTP Client will inform the FTP server of its desire for a file over the Control Connection. This request will include the source port the Client will be using. This port number will also be an unused port greater than 1024. The Server will then acknowledge the request, and the client will then open the Data Connection. The Server will acknowledge the Data Connection, the Client then acknowledges the acknowledgement, and data can then be transferred.

The FTP Data Connection Process



It's a common misconception that the Server tells the Client what port numbers are going to be used. As illustrated here, the Client makes the decision as to what port number it's going to use, not the Server.

Trivial File Transfer Protocol, as the name implies, is much simpler than FTP. TFTP does not offer the end user the features that FTP does, but it uses less resources, such as memory.

TFTP runs over UDP, so there is no error recovery or a need for an established connection before sending the data. However, TFTP does use a sequence number in each packet. These are used for acknowledgements and to indicate a possible need to resend the data.

	Protocol	Connections	Comparative Resource Need
FTP	TCP	Two; a control connection and a data connection.	Greater than TFTP.
TFTP	UDP	None. UDP is connectionless.	Less than FTP.

Advanced IGP Q & A

1. What command is used to apply an access list?

- A. **access-group**
- B. **ip access-group**
- C. **access-list**
- D. **ip access-list**
- E. **access-list apply**
- F. **ip acl apply**

ANSWER: B. The access list is created with the "access-list" command, but it's applied to the interface with "ip access-group". Don't forget the "ip" when applying the command.

2. Which statement is true of the term "implicit deny"?

- A. **All interfaces deny packets unless an access-list allowing packets is applied to that interface.**
- B. **An access-list denies all packets unless they're explicitly permitted.**
- C. **An access-list permits all packets, and an implicit deny statement must be added to the end of the access-list, or all packets will go through.**
- D. **A standard access-list denies all packets unless explicitly permitted, but extended and named access-lists permit all packets unless explicitly denied.**

ANSWER: B. When configuring an access list, it is vital to remember that all packets will be denied by the ACL unless explicitly permitted. In effect, Cisco ACLs have a "deny any" statement at the end of them, even though you won't actually see it in the running configuration.

3. Which of the following numerical ranges can be used for standard access lists? (Choose two.)

- A. 0 – 99
- B. 100 – 199
- C. 700 – 799
- D. 1200-1299
- E. 1300-1399
- F. 2000-2699

ANSWER: A, E. Standard access-lists use the ranges 0 – 99 and 1300 – 1399.

4. Once a match in an access-list is found, what action does the router take?

- A. **The router will stop checking lines in the access-list, and take the appropriate action of the line that matched, either “permit” or “deny”.**
- B. **The router will check the rest of the access-list for “better” matches, according to the longest-match rule.**
- C. **The router will check the next line of the access-list to ensure the appropriate action is taken.**
- D. **The router will check the implicit deny statement of the access-list.**

ANSWER: A. When a match is found in an access-list, the router will stop running the ACL and will take the appropriate action. It's important to have the lines of the ACL in the right order to get the desired effect.

5. Which of the following statements is true regarding wildcard masks? Choose two.

- A. **Zeroes indicate a bit that must match.**
- B. **Zeroes indicate a bit that doesn't have to match.**
- C. **Ones indicate a bit that must match.**
- D. **Ones indicate a bit that doesn't have to match.**

ANSWER: A, D. When configuring wildcard masks, a zero indicates a bit that must match, and a one indicates a bit that doesn't have to match. This is the opposite of a subnet mask, where zeroes indicate a bit that doesn't have to match, and a one indicates a bit that does have to match.

6. What keyword represents the wildcard mask 0.0.0.0 ?

- A. "any"
- B. "all"
- C. "host"
- D. "zeroes"

ANSWER: C. The wildcard mask 0.0.0.0 means that only the single address referenced to on this line in the ACL is being affected. The keyword "host" means the same thing.

Consider a line of an ACL that should deny traffic from the single host 172.16.78.57. Both of the following lines will do this:

Access-list 17 deny 172.16.78.57 0.0.0.0

Access-list 17 deny host 172.16.78.57

Note that the "host" keyword goes in front of the IP address in question, and the wildcard mask 0.0.0.0 goes behind the IP address.

7. Your supervisor has told you to prevent packets from entering the Serial0 interface of a particular router if the packets originated from network 172.16.1.0 /25, and to allow all other packets. Which of the following is the appropriate configuration to do so?

- A. **Access-list 5 deny 172.16.1.0 0.0.0.255**
Access-list 5 permit all

Interface serial0
ip access-group 5 in

- B. **Access-list 5 deny 172.16.1.0 255.255.255.128**
Access-list 5 permit any

Interface serial0
ip access-group 5 in

- C. **access-list 5 deny 172.16.1.0 0.0.0.127**
access-list 5 permit any

interface serial0
access-group 5 in

**D. access-list 5 deny 172.16.1.0 0.0.0.127
access-list 5 permit any**

**interface serial0
ip access-group 5 in**

**E. access-list 5 deny 172.16.1.0 0.0.0.255
access-list 5 permit all**

**interface serial0
ip access-group 5 in**

ANSWER: D. The approach to take to any such question is to see what needs to be permitted or denied, configure that line of the ACL, and then determine from the question whether the rest of the traffic should be accepted or denied.

The network named in the question is 172.16.1.0 /25, and this traffic should be denied. The wildcard mask that will be in this line of the ACL must state that the first 25 bits of the network must match the network named in the ACL, and that the last 7 bits don't matter.

Because you know binary math rather than memorizing a chart, you can write out what that wildcard mask needs to be. Remember that with wildcard masks, zeroes mean "this bit must match", and ones mean "this bit doesn't have to match".

Breaking down the wildcard mask:

1 st Octet – All bits must match.	00000000
2 nd Octet – All bits must match.	00000000
3 rd Octet – All bits must match.	00000000
4 th Octet – 1 st bit must match, the rest do not matter.	01111111
Resulting Wildcard Mask:	00000000 00000000 00000000 01111111

Obviously, the first three octets of the wildcard mask will be 0.0.0. Using your knowledge of binary math, the value of the final octet is quickly revealed:

	128	64	32	16	8	4	2	1
1 st Octet:	0	0	0	0	0	0	0	0
2 nd Octet:	0	0	0	0	0	0	0	0
3 rd Octet:	0	0	0	0	0	0	0	0
4 th Octet:	0	1	1	1	1	1	1	1

Adding up all the values in the 4th octet represented with a "1" gives us a total of 127. The wildcard mask is 0.0.0.127.

We know the ACL line to deny the network named in the question will be: access-list 5 deny 172.16.1.0 0.0.0.127. Answers A, B, and E are eliminated.

The ACL in choices C and D are the same. Why is C wrong and D right? Because C is using "access-group 5" to apply the ACL, and D is using "ip access-group 5", the correct command.

Watch these small details when you're taking and passing the CCNA exam, and all other Cisco exams for that matter.

The first few times you work through a binary math question, whether it be a VLSM, subnetting, or wildcard masking question, it might take you a few minutes. With *continual practice*, you will be totally prepared for any of these questions on the CCNA exam and on the job.

You have now seen that you use binary math with OSPF, EIGRP, variable-length subnet masks, subnetting, and wildcard masks in ACLs. This is why I'm so adamant about CCNA candidates truly understanding binary math. Once you practice binary math, you'll see it's actually quite easy.

Keep working!

8. Which of the following is true of a standard access list? Choose two.
- A. Standard ACLs consider only the source of a packet when making a permit / deny decision.
 - B. Standard ACLs consider only the destination of a packet when making a permit / deny decision.
 - C. Standard ACLs consider both the source and destination of a packet.
 - D. Standard ACLs contain an implicit deny.
 - E. Standard ACLs don't contain an implicit deny; only extended ACLs do.

ANSWER: A, D. Standard ACLs only look at the source of a packet. To permit or deny a packet on the basis of a destination, an extended ACL must be configured. Both standard and extended ACLs contain an implicit deny statement at the end.

9. You are configured access-list 5, and wish to prevent the implicit deny statement from ever occurring. Which of the following statements will do this?

- A. no access-list 5 implicit
- B. no access-list 5 ip implicit
- C. access-list 5 permit any any
- D. access-list 5 permit any
- E. This can't be done; the implicit deny statement is always run at the end of a standard ACL.

ANSWER: D. The statement "access-list 5 permit any" will do just that; it will permit any packets, and the implicit deny statement would never be run. Remember that once a match is found in the ACL, the router stops checking the ACL. In this case, the "permit any" statement will match any packet, and the implicit deny statement is never seen by the router.

10. Consider the following configuration:

```
R3#conf t  
R3(config)#access-list 5 permit 172.12.12.0 0.0.0.255  
R3(config)#interface serial0  
R3(config-if)#ip access-group 5 in
```

What will happen to packets received on interface Serial1 from network 172.12.12.0 /24?

- A. They will be denied due to the implicit deny statement.
- B. They will be denied because the configuration on the interface is incorrect.
- C. They will be permitted because ACL 5 is correctly configured and correctly applied to the interface.
- D. They will be permitted for other reasons.

ANSWER: D. They will be permitted, but not because of ACL 5. That ACL is applied to interface serial0, and the question is asking about packets received on Serial1.

Watch the details.

11. How many access-lists can be applied on a router?

- A. One per interface, either inbound or outbound.
- B. Two per interface, one inbound and one outbound.
- C. One per router, either inbound or outbound.
- D. Two per router, one inbound and one outbound.

ANSWER: B. Cisco routers permit two ACLs to be placed on an interface; one inbound and one outbound. Multiple inbound or outbound ACLs cannot be placed on a single interface.

12. As a responsible network engineer, you want to leave an explanation as to what you were filtering with access-list 5. Which of the following commands will help you do this?

- A. remark
- B. appendix

- C. **reason**
- D. **purpose**
- E. **name**

ANSWER: A. The “remark” command will allow you to leave a description of the purpose of your access-list for any engineers that follow you.

13. What keyword represents a wildcard mask of 255.255.255.255?

- A. **host**
- B. **any**
- C. **all**
- D. **32**
- E. **0**

ANSWER: B. The “any” option in an ACL represents a wildcard mask of 255.255.255.255. That wildcard mask means that any address will match.

14. Consider the following ACL:

```
R3#conf t
R3(config)#access-list 15 permit any
R3(config)#access-list 15 deny 172.18.18.0 0.0.0.255
R3(config)#interface serial0
R3(config-if)#ip access-group 15 out
```

What will happen when packets with a source of 172.18.18.0 /24 attempt to exit interface serial0?

- A. **The packets will be denied because the ACL is incorrectly applied.**
- B. **The packets will be denied because one line in the ACL explicitly denies that traffic.**
- C. **The packets will be permitted because the ACL is incorrectly applied.**
- D. **The packets will be permitted because of the “permit any” statement.**

ANSWER: D. The “permit any” statement will allow any packet to go through.

Remember that an ACL is checked from the top down, and once a match is found, that's it. The first line of this ACL will permit any packet; the deny statement will never be checked since the first line matches everything. The order of the lines in an ACL is vital.

15. Consider the following configuration:

```
R3#conf t
R3(config)#access-list 15 deny 172.27.27.0 0.0.0.255
R3(config)#access-list 15 permit any
R3(config)#interface serial0
R3(config-if)#ip access-group 15 out
```

What will happen to packets with a source of 172.27.27.0 /24 when sent out interface Serial0?

- A. The packets will be implicitly denied.
- B. The packets will be explicitly denied.
- C. The packets will be dropped; the ACL is incorrectly configured.
- D. The packets will be sent; the ACL is incorrectly applied.

ANSWER: B. The packets will be explicitly denied. The first line of the ACL matches the packets named in the question, so they are denied. Packets from any other source will match against line two and will be sent.

16. Which of the following statements are true of extended access lists? Choose four.

- A. Either the source or destination can be considered, but not both.
- B. Both the source and destination are considered.
- C. Source port and destination port numbers can be considered.
- D. Port numbers cannot be considered when configuring an extended ACL; only named access lists can do that.
- E. The keyword "host" can be used in place of a wildcard mask of 0.0.0.0.
- F. The keyword "host" can be used in place of a wildcard mask of 255.255.255.255.
- G. The keyword "any" can be used in place of a wildcard mask of 0.0.0.0.
- H. The keyword "any" can be used in place of a wildcard mask of 255.255.255.255.

ANSWER: B, C, E, H. Extended access lists match against both the source and destination address, and port numbers can be configured in extended ACLs. The keyword "host" represents a wildcard mask of 0.0.0.0, and "any" represents a wildcard mask of 255.255.255.255.

17. You wish to remove access-list 37 from interface serial0. What command should be run on the interface to do so?

- A. **no ip access-list 37**
- B. **no ip access-group 37**
- C. **no access-list 37**
- D. **no access-group 37**

ANSWER: B. Anytime you're removing an ACL from an interface, use the same command you use to apply it to the interface in the first place with a "no" in front of it. Don't forget the "ip". The command "no access-list 37" is valid, but in global configuration mode, not interface mode. That command actually removes the ACL from the router.

18. Which of the following statements is true of an extended ACL?

- A. **Both the source and destination have to match a line of the ACL for that line's permit / deny action to take place.**
- B. **Either the source or destination can match the ACL line for the permit / deny statement to take place. They don't both have to match.**
- C. **The source and destination both have to match only if port numbers are used in the ACL. Otherwise, only the source or destination have to match.**
- D. **Only the source has to match. The destination is listed for security purposes.**

ANSWER: A. For a match to occur in a line on an extended ACL, both the source and destination must match.

19. What well-known port number is used to represent DNS in an ACL?

- A. **23**
- B. **53**
- C. **80**

- D. 110**
- E. 1024**

ANSWER: B. DNS is represented by port number 53.

20. What well-known port number is used to represent Telnet in an ACL?

- A. 23**
- B. 53**
- C. 80**
- D. 110**
- E. 1024**

ANSWER: A. Telnet runs on port 23.

21. What well-known port number is used to represent HTTP in an ACL?

- A. 23**
- B. 53**
- C. 80**
- D. 110**
- E. 1024**

ANSWER: C. HTTP's port number is 80.

22. You want to configure a named standard access list. The name of the list will be STOP_BROADCASTS. Which of the following lines is the correct way to start writing this list?

- A. access-list 5 name STOP_BROADCASTS**
- B. access-list 110 name STOP_BROADCASTS**
- C. access-list standard STOP_BROADCASTS**
- D. access-list STOP_BROADCASTS**
- E. ip access-list STOP_BROADCASTS**
- F. ip access-group STOP_BROADCASTS**

ANSWER: E. The configuration of named access-lists begins with "ip access-list" followed by "standard" or "extended", then the name of the ACL.

23. You've created a named ACL, "STOP_NETWORK_110". What interface-level command will be used to apply this ACL to a given interface?

- A. **ip access-list name STOP_NETWORK_110 out**
- B. **ip access-group name STOP_NETWORK_110 out**
- C. **ip access-list STOP_NETWORK_110 out**
- D. **ip access-group STOP_NETWORK_110 out**
- E. **ip name list STOP_NETWORK_110 out**
- F. **ip list name STOP_NETWORK_110 out**

ANSWER: D. A named ACL is applied to an interface in the same way that a numbered ACL is, with "ip access-group" on the interface.

24. You are configuring an ACL to apply to the VTY lines, for use when users attempt to telnet to your router. Your configuration is at this stage:

```
R3#conf t
R3(config)#access-list 78 permit host 122.12.12.12
R3#conf t
R3(config)#line vty 0 4
R3(config-line)#login
R3(config-line)#password CCNA
R3(config-line)#

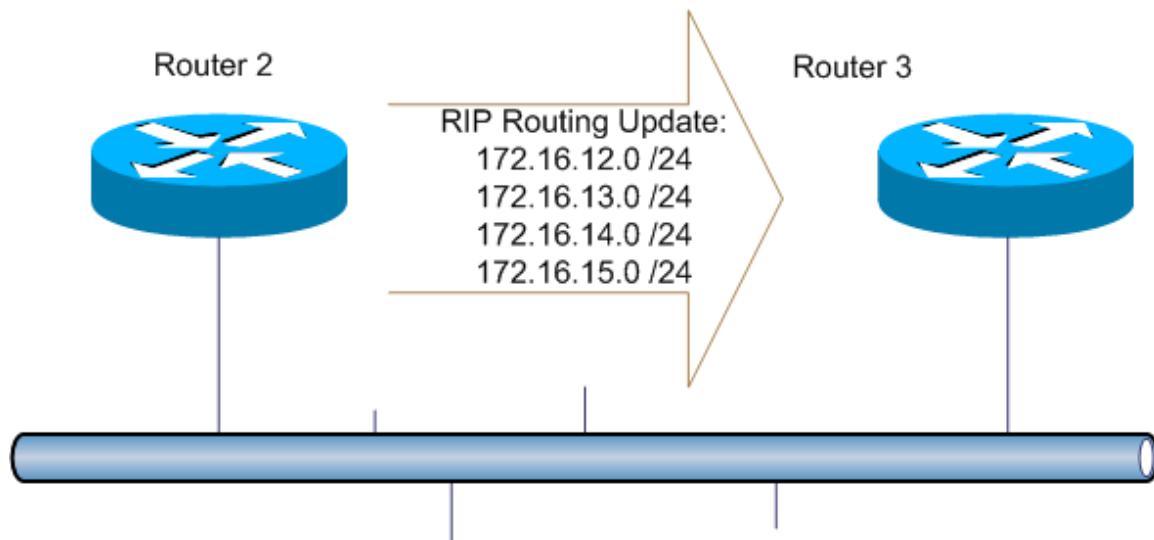
```

What command is needed on the final line?

- A. **access-group 78 in**
- B. **ip access-group 78 in**
- C. **access-list 78 in**
- D. **ip access-list 78 in**
- E. **access-class 78 in**
- F. **ip access-class 78 in**

ANSWER: E. The "access-class" command is used to apply an ACL to VTY lines.

25. Consider the following network diagram:



You want to send a single route from Router 2 to Router 3, representing these four routes. Which of the following interface-level statements will perform this task?

- A. **ip summary-address rip 172.16.12.0 255.255.252.0**
- B. **ip summary-address rip 172.16.12.0 0.0.3.255**
- C. **ip summary-address rip 172.16.12.0 255.255.255.252**
- D. **ip summary-address rip 172.16.12.0 0.0.0.3**
- E. **No summarization statement can do this.**

ANSWER: A. Again, your knowledge of binary math makes this question easy for you. To configure a summary for these four routes, write them out in binary, remembering that the binary math values from left to right are 128, 64, 32, 16, 8, 4, 2, and 1.

	1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
172.16.12.0	10101100	00010000	00001100	00000000
172.16.13.0	01100100	00010000	00001101	00000000
172.16.14.0	01100100	00010000	00001110	00000000
172.16.15.0	01100100	00010000	00001111	00000000

Working from left to right, we see that each address has 22 bits in common. The resulting network number is the summary address

Chris Bryant, CCIE #12933 396

www.thebryantadvantage.com

© 2004 The Bryant Advantage

itself. Adding the numbers up in the 22 common bits and placing a "0" for the rest, we get a network number of 172.16.12.0. That's the summary address.

Now we need a subnet mask to go with this address. We have 22 common bits, bits that must match. The rest of the bits are "don't care" bits. In subnet masking, "1" represents the "must match" bits and "0" represents the "don't care" bits. This gives us a subnet mask of 11111111 11111111 11111100 00000000, resulting in a subnet mask of 255.255.252.0.

The final summary address is 172.16.12.0 255.255.252.0.

Notes

26. At what level of an internetwork is CIDR generally applied?

- A. At the user's desktop.
- B. On the local switches.
- C. On the local routers.
- D. At the Internet Service Provider.
- E. On the remote site's routers.
- F. On the remote site's switches.

ANSWER: D. The ISP uses CIDR to save addresses. It's not configured at the local network level.

27. Which of the following is true of the IP address 10.17.17.17? Choose three.

- A. It is an RFC 1918 private address.
- B. It is not an RFC 1918 private address.
- C. This address is appropriate for use as a inside local address with NAT.
- D. This address is appropriate for use as a inside global address with NAT.
- E. This address is appropriate for use on an interface that directly connects to the Internet.
- F. This address is not appropriate for use on an interface that directly connects to the Internet.

ANSWER: A, C, F. The RFC 1918 private addresses are 10.0.0.0/8, 172.16.0.0 /12, and 192.168.0.0 /16. These addresses are appropriate for use as inside local addresses with NAT and should not be leaked out to the Internet.

28. You have an Ethernet network configured with network 10.0.0.0 /8. Your Serial interface is configured with IP address 222.1.1.1 /24, and is directly connected to the Internet. You have three hosts on this network that you wish to have total Internet connectivity. Using NAT to do so, which of the following interface configurations is the correct one?

A. R3(config)#interface ethernet0

```
R3(config-if)#ip address 10.1.1.1 255.0.0.0  
R3(config-if)#ip nat  
R3(config-if)#interface serial0  
R3(config-if)#ip address 222.1.1.1 255.255.255.0  
R3(config-if)#ip nat
```

B. R3(config)#interface ethernet0

```
R3(config-if)#ip address 10.1.1.1 255.0.0.0  
R3(config-if)#ip nat inside  
R3(config-if)#interface serial0  
R3(config-if)#ip address 222.1.1.1 255.255.255.0  
R3(config-if)#ip nat outside
```

C. R3(config)#interface ethernet0

```
R3(config-if)#ip address 10.1.1.1 255.0.0.0  
R3(config-if)#ip nat outside  
R3(config-if)#interface serial0  
R3(config-if)#ip address 222.1.1.1 255.255.255.0  
R3(config-if)#ip nat inside
```

D. R3(config)#interface ethernet0

```
R3(config-if)#ip address 10.1.1.1 255.0.0.0  
R3(config-if)#nat inside  
R3(config-if)#interface serial0  
R3(config-if)#ip address 222.1.1.1 255.255.255.0  
R3(config-if)#nat outside
```

ANSWER: B. The answers are all close, but the only correct one is B. "A" leaves the "inside" and "outside" off the NAT command; "C" has the "inside" and "outside" backwards; "D" leaves the "ip" off the NAT command. Be prepared for questions where the answers look a lot alike.

29. Now that you've configured the interfaces, you decide to use static NAT to finish the configuration. Which of the following will successfully map three inside local addresses from the previous question's subnet to three appropriate inside global addresses?
- A. R3(config)#**ip nat inside source static 10.1.1.1 222.1.1.2**
R3(config)#**ip nat inside source static 10.1.1.2 222.1.1.3**
R3(config)#**ip nat inside source static 10.1.1.3 222.1.1.4**
 - B. R3(config)#**ip nat inside static source 10.1.1.1 222.1.1.2**
R3(config)#**ip nat inside static source 10.1.1.2 222.1.1.3**
R3(config)#**ip nat inside static source 10.1.1.3 222.1.1.4**
 - C. R3(config)#**ip nat inside static 10.1.1.1 222.1.1.2**
R3(config)#**ip nat inside static 10.1.1.2 222.1.1.3**
R3(config)#**ip nat inside static 10.1.1.3 222.1.1.4**
 - D. R3(config)#**ip nat inside source 10.1.1.1 222.1.1.2**
R3(config)#**ip nat inside source 10.1.1.2 222.1.1.3**
R3(config)#**ip nat inside source 10.1.1.3 222.1.1.4**

ANSWER: A. The correct syntax for this command is only listed in answer "A".

Advanced TCP/IP Concepts Lab

Before beginning the lab, a routing protocol must be configured. The protocol should be RIPv2, OSPF, or EIGRP. Each router must be able to ping the loopbacks on each of the other two routers and the Serial interface connected to the Frame Relay cloud. R2 and R3's Ethernet interfaces should be able to be pinged by every router. **The BRI interface and the directly connected interface between R1 and R3 should be shut down.**

With the **access-list** command, configure R1 so that only packets from the 172.12.123.0 /24 network can enter the Serial interface. Test the configuration by sending a **ping** on R2 from both 172.12.123.2 and 2.2.2.2.

Allowing only traffic from 172.12.123.0/24 to enter the Serial interface on R1.

```
R1#conf t
R1(config)#access-list 1 permit 172.12.123.0 0.0.0.255
< Wildcard masks are used with access lists. There is an implicit deny at the end of every
access list; any traffic that is not expressly permitted is implicitly denied. >
R1(config)#interface serial0/0
R1(config-if)#ip access-group 1 in
< Access lists are applied to interfaces with the ip access-group command. The direction
the access-list is applied in follows that command. >
```

A ping will be sent from R2 from two different addresses. A ping such as the ones sent in labs up to this point are seen by the remote router as having originated from the interface it left the other router in. For example, running ping 172.12.123.1 from R2 will result in a ping with a source address of 172.12.123.2. Since this address falls in the “permit” statement of the access-list configured above, the traffic will be let through at R1’s serial interface, and the ping succeeds.

```
R2#ping 172.12.123.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.12.123.1, timeout is 2 seconds:
!!!!!
On R1, run “show ip access-list” to see matches against every statement in the access-
list.
R1#show ip access-list
Standard IP access list 1
    permit 172.12.123.0, wildcard bits 0.0.0.255 (5 matches)
```

The matches refer to the five ping packets that were received from R2.

Sending an extended ping from R2.

*An extended ping offers a great many options, and the one to use here is to set the source address. To send an extended ping, simply enter **ping** without an IP address following it.*

R2#ping

Protocol [ip]:

Target IP address: 172.12.123.1

Repeat count [5]:

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]: y

Source address or interface: loopback0

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [n]:

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.12.123.1, timeout is 2 seconds:

Packet sent with a source address of 2.2.2.2

U.U.U

Success rate is 0 percent (0/5)

The key is in the extended commands. The default for this is “N”, but by answering “Y”, the source interface of the ping can be changed as shown. The ping sent from the loopback address 2.2.2.2 does not go through. That traffic is blocked by the access-list on R1.

To be able to see how many packets are denied by a standard ACL, the implicit deny statement must be explicitly configured. Show ip access-list will then show the denied packets as well as the permitted ones.

R1#conf t

R1(config)#no access-list 1

R1(config)#access-list 1 permit 172.12.123.0 0.0.0.255

R1(config)#**access-list 1 deny any**

< The implicit “deny any” is expressly configured so packets denied by it will show in “show ip access-list, as seen below. > Note: Not all IOS versions will show the number of matches !

R1#show ip access-list

Standard IP access list 1

 permit 172.12.123.0, wildcard bits 0.0.0.255 (4 matches)

deny any (8 matches)

On R3, write a standard ACL that denies traffic from IP address 1.1.1.1, but permits all other IP traffic with the **access-list** and **ip access-group** commands.

```
R3#conf t  
R3(config)#access-list 1 deny 1.1.1.1  
R3(config)#access-list 1 perm any  
R3(config)#interface serial 0.31  
R3(config-if)#ip access-group 1 in
```

The first line of the ACL denies traffic from 1.1.1.1, and the second permits all other traffic. The order of the lines in an ACL is vital. If these lines were reversed and “access-list 1 permit any” was the first line, all traffic would be permitted, including traffic from 1.1.1.1. The deny statement would never be reached.

From R1, ping 172.12.123.3, first with a regular ping, then with an extended ping from source 1.1.1.1.

```
R1#ping 172.12.123.3  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.123.3, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms  
R1#ping  
Protocol [ip]:  
Target IP address: 172.12.123.3  
Repeat count [5]:  
Datagram size [100]:  
Timeout in seconds [2]:  
Extended commands [n]: y  
Source address or interface: 1.1.1.1  
Type of service [0]:  
Set DF bit in IP header? [no]:  
Validate reply data? [no]:  
Data pattern [0xABCD]:  
Loose, Strict, Record, Timestamp, Verbose[none]:  
Sweep range of sizes [n]:  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.12.123.3, timeout is 2 seconds:  
Packet sent with a source address of 1.1.1.1  
U.U.U  
Success rate is 0 percent (0/5)  
As expected, the ping from 172.12.123.1 is good, but the ping from 1.1.1.1 was stopped by the ACL on R3.
```

On R3, run **show ip access-list** to view the number of packets that have been permitted and denied.

```
R3#show ip access-list
Standard IP access list 1
deny 1.1.1.1 (5 matches)
permit any (20 matches)
```

The pings sourcing from 1.1.1.1 were stopped at the serial interface. All other traffic is being permitted.

Using an extended ACL on R3, prevent traffic from coming into the router's Ethernet interface if the source is 172.23.23.2 and the destination is 3.3.3.3.

Configuring and testing an extended ACL.

To define a source and destination in an ACL, an extended ACL must be used. The numeric ranges for extended ACLs are 100-199 and 2000 - 2699.

```
R3#conf t
R3(config)#access-list 125 deny ip host 172.23.23.2 host 3.3.3.3
R3(config)#access-list 125 perm ip any any
```

The first line of the ACL uses the "host" option. This takes the place of a wildcard mask of 0.0.0.0; that is, the host option means that the IP address that follows it is the only IP address to be affected. It's used twice in this ACL, since a specific source address and a specific destination address are being denied.

The second line uses the "any" option. This takes the place of a wildcard mask of 255.255.255.255. Since "any" is used twice, once for the source and once for the destination, all traffic is affected by this line.

The ACL is then applied to the Ethernet interface. There is now one ACL on the Ethernet interface and one on the serial interface. The rule is that two ACLs can be applied to a single interface, one affecting outgoing traffic and another affecting incoming traffic.

```
R3(config)#interface ethernet0/0
R3(config-if)#ip access-group 125 in
```

From R2, ping 172.23.23.3 and 3.3.3.3 with regular pings. After doing so, run **show ip access-list** on R3.

```
R2#ping 3.3.3.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

```
R2#ping 172.23.23.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.23.23.3, timeout is 2 seconds:

!!!!

The pings to 3.3.3.3 fail, but the pings to 172.23.23.3 succeed. Since the standard ping command was used, the source IP address of the ping is the exiting interface, 172.23.23.2.

```
R3#show ip access-list
```

Standard IP access list 1

deny 1.1.1.1 (8 matches)

permit any (70 matches)

Extended IP access list 125

deny ip host 172.23.23.2 host 3.3.3.3 (8 matches)

permit ip any any (386 matches)

Both ACLs configured on R3 are shown. List 125 is denying the specific packets with a source of 172.23.23.2 and a destination of 3.3.3.3. All other packets are going through.

In The REAL World...

For CCNA and CCNP exams, you're expected to have some of these ACL ranges memorized. On the job, you can view the ranges with Cisco IOS Help:

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#access-list ?
<1-99>      IP standard access list
<100-199>    IP extended access list
<1000-1099>  IPX SAP access list
<1100-1199>  Extended 48-bit MAC address access list
<1200-1299>  IPX summary address access list
<1300-1999>  IP standard access list (expanded range)
<200-299>    Protocol type-code access list
<2000-2699>  IP extended access list (expanded range)
<300-399>    DECnet access list
<400-499>    XNS standard access list
<500-599>    XNS extended access list
<600-699>    Appletalk access list
<700-799>    48-bit MAC address access list
<800-899>    IPX standard access list
<900-999>    IPX extended access list
dynamic-extended Extend the dynamic ACL absolute timer
rate-limit     Simple rate-limit specific access list
```

Simulator questions regarding ACL configuration on the CCNA and CCNP exams will most likely not offer IOS Help. Be familiar with the more common ACL ranges, particularly for IP Standard and IP Extended lists.

On R2, use the **ip access-list** command to prevent any traffic from interface 3.3.3.3. Apply this named ACL to the Ethernet interface.

```
R2#conf t
R2(config)#ip access-list standard BLOCKNETWORK3
R2(config-std-nacl)#deny host 3.3.3.3
R2(config-std-nacl)#perm any
R2(config-std-nacl)#interface ethernet0/0
R2(config-if)#ip access-group BLOCKNETWORK3 in
```

*To configure a named access list, use the **ip access-list** command, followed by "standard" or "extended", and then the name of the ACL. Make the name intuitive. Apply a named ACL with the **ip access-group** command, just as if the list were a numbered ACL.*

From R3, send an extended ping that sources from 3.3.3.3 to 172.23.23.2. When the ping fails, run **show ip access-list** on R2 to ensure the ACL is blocking the packets.

```
R3#ping
Protocol [ip]:
Target IP address: 172.23.23.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 3.3.3.3
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.23.23.2, timeout is 2 seconds:
Packet sent with a source address of 3.3.3.3
.....
Success rate is 0 percent (0/5)\

R2#show ip access-list
Standard IP access list BLOCKNETWORK3
  deny 3.3.3.3 (5 matches)
  permit any (18 matches)
```

The pings with a source address of 3.3.3.3 are blocked by the ACL.

On R3, write a standard ACL that permits only host 172.12.123.1. Allow the explicit deny to prevent all other addresses. Apply the access-list to the VTY lines to allow only this address to telnet into R3 with the **access-class** command. Set a password of CCNA for telnet access.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#access-list 5 permit 172.12.123.1
R3(config)#line vty 0 4
< Configures the VTY lines, used for Telnet access. >
R3(config-line)#login
< Allows login with a password that must be configured under the VTY lines. >
R3(config-line)#password CCNA
< Password to be used for Telnet access. >
R3(config-line)#access-class 5 in
< The access-list is applied to VTY lines with the access-class command. Only the user
specified in the ACL will be able to Telnet to this router. >
```

From R1 and R2, telnet to 172.12.123.3.

```
R1#telnet 172.12.123.3
Trying 172.12.123.3 ... Open

User Access Verification

Password:
R3>logout

R2#telnet 172.12.123.3
Trying 172.12.123.3 ...
% Connection refused by remote host
```

From R1, the telnet succeeds. While performing this lab, notice that the password never appears when telnetting to the router, nor does the cursor move.

From R2, the telnet attempt fails. The console message is simply that the remote host refused it. It was refused because only R1's serial address is permitted by the ACL applied to the VTY lines; the implicit deny stops all other telnet attempts.

On R3, run **show ip access-list**.

```
R3#show ip access-list
Standard IP access list 1
    deny  1.1.1.1 (8 matches)
    permit any (430 matches)
Standard IP access list 5
    permit 172.12.123.1 (6 matches)
Extended IP access list 125
    deny ip host 172.23.23.2 host 3.3.3.3 (18 matches)
    permit ip any any (1248 matches)
```

Note the “permit any” statements on the first two ACLs continue to accrue as the lab progresses, as routing update packets are being sent around the network. The number and frequency depends on the routing protocol.

On R1, use the **ip host** command to configure the router to telnet to 172.12.123.3 when “R3” is typed. (No quotation marks.)

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip host R3 172.12.123.3
```

```
R1#R3
Trying R3 (172.12.123.3)... Open
```

User Access Verification

```
Password:
R3>en
Password:
R3#
```

*After configuring the **ip host** command, simply entering “R3” on R1 will telnet to 172.12.123.3.*

A Final Word....

I'd like to thank you for choosing The Bryant Advantage CCNA Course Guide and Lab Workbook to further your Cisco certification study efforts. Having worked my way from the CCNA to the CCIE, I know the challenges you face when you're beginning your Cisco studies, and I'm hopeful this book has helped you on the way to achieving all your professional goals.

After capturing your CCNA certification, I hope you'll move on to the CCNP, and then perhaps the CCIE. While these certifications do take time, discipline, and dedication, they can open just as many doors for you as they have for me. There is no field in the world that rewards individual effort and incentive more than information technology. The Cisco professional certifications are a sign to current and potential employers that you have that discipline and incentive.

And please visit www.thebryantadvantage.com for many other CCNA and CCNP study aids, including our popular Flash Card Books and The Bryant Advantage *Mastery Series*, coming very soon. Don't forget that the purchase price of this workbook can be applied toward my custom-written CCNA Course, offered both in-person and on the Internet, during the week and on weekends.

I also hope you'll take the time to drop me a line at
chris@thebryantadvantage.com when you pass your CCNA exam!

To your success,

Chris Bryant
Cisco Certified Internetwork Expert #12933