APPENDIX F

ICND Exam Updates: Version 1.0

Over time, reader feedback allows Cisco Press to gauge which topics give our readers the most problems when taking the exams. To assist readers with those topics, the author creates new materials clarifying and expanding upon those troublesome exam topics. As mentioned in the introduction to the *CCNA ICND Exam Certification Guide*, the additional content about the exam is contained in a PDF document on this book's companion website located at http://www.ciscopress.com/title/158720083X.

This appendix presents all the latest update information available at the time of this book's printing. To make sure you have the latest version of this document, be sure to visit the companion website to learn whether any more recent versions have been posted since this printing went to press.

This appendix attempts to fill the void that occurs with any print book. In particular, this appendix

- Mentions technical items that might not have been mentioned elsewhere in the book.
- Emphasizes points that might have been mentioned briefly inside the book.
- Reviews several related topics that might have been under separate headings in the original chapters.
- Provides a way to get up-to-the-minute current information about content for the exam.

Always Get the Latest at the Companion Website

You are reading the version of this appendix that was available when your book was printed. However, given that the main purpose of this appendix is to be a living, changing document, it is very important that you look for the latest version on-line at the book's companion website. To do so do the following:

1. Browse to http://www.ciscopress.com/title/158720083X.
2. Select the **Downloads** option under the **More Information** box.

3. Download the latest "ICND Appendix F" document

NOTE Note that the downloaded document has a version number. Compare the version of this print Appendix F (version 1.0) with the latest online version of this appendix, and follow this advice:

- **Same version—**Ignore the PDF that you downloaded from the companion website.
- **Website has a later version—**Ignore this Appendix F in your book, and just read the latest version that you downloaded from the companion website.

Technical Content

The topics in this appendix provide clarifications, technical tidbits, points of emphasis, and the like. The section titles for each topic identify the corresponding chapters of the book relevant to that topic.

Additionally, one section at the end of this appendix is covered in Chapter 7, "Operating Cisco Routers," of the *CCNA INTRO Exam Certification Guide*. This material is included here for those of you who might have a copy of the *CCNA ICND Exam Certification Guide* but not a copy of the *CCNA INTRO Exam Certification Guide*.

VLAN, Trunking, and VTP (Chapter 3)

Chapter 3, "Virtual LANs and Trunking," covers virtual LANs (VLANs) and VLAN Trunking Protocol (VTP). This section provides additional information about configuration for VLANs and VTP, including some of the nuances and potential problems when using VTP.

VLAN and VTP Configuration

Chapter 3 shows how to configure VLANs and VTP from VLAN database mode. You can also configure these features directly from configuration mode. Example F-1 shows how to configure VLANs and VTP; you can compare the equivalent configuration in Examples 3-1 and 3-2 from Chapter 3.

Example F-1 *VLAN and VTP Configuration from Configuration Mode*

```
sw1-2950#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
sw1-2950(config)#interface range fastethernet 0/5 - 8
sw1-2950(config-if)#switchport access vlan 2
% Access VLAN does not exist. Creating vlan 2
sw1-2950(config-if)#exit
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Example F-1 *VLAN and VTP Configuration from Configuration Mode*

```
! Above, VLAN 2 did not yet exist. When the switchport access vlan 2 command was
! used, the switch automatically created VLAN 2.
! Below, VLAN 3 is created explicitly, and given the correct name. At that point,
! when interfaces 9 – 12 are added to VLAN 3, the switch does not issue a message
! saying it is creating a VLAN, because VLAN 3 already exists.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
sw1-2950(config)#vlan 3
sw1-2950(config-vlan)#name Wilma-3
sw1-2950(config-vlan)#exit
sw1-2950(config)#interface range fastethernet 0/9 - 12
sw1-2950(config-if)#switchport access vlan 3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Below, the VTP domain name is set to "fred".
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
sw1-2950#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
sw1-2950(config)#vtp domain fred
Changing VTP domain name from NULL to fred
sw1-2950(config)#^Z
```

The commands shown in the example are very similar to those used in VLAN database mode. For instance, the **vtp domain fred** command is the same in both VLAN database mode and configuration mode. In configuration mode, however, you must use the **name** command to configure a VLAN name inside VLAN configuration mode, in contrast to treating the VLAN name as a parameter in the VLAN database mode **vlan** command. The meaning of the command options in each mode are relatively similar. The generic command syntax for the **vtp** command in configuration mode is as follows:

```
vtp {domain domain-name | password password | pruning | v2-mode | {server | client
    | transparent}}
```

For instance, from configuration mode, the **vtp server** command sets the switch to be a VTP server, and the **vtp pruning** command tells the switch to attempt pruning.

VTP Requirements

VTP must meet the following requirements to work on a set of switches:

- VLAN trunking (802.1Q or inter-switch link [ISL]) must be operational on the segments connecting the switches.
- At least one switch must be in VTP server mode.
- The switches must have the same case-sensitive domain name.

- If a password is configured, the switches must have the same case-sensitive password configured.

When you receive a new Cisco 2950 switch, it defaults to VTP server mode, with a null domain name, and with trunking defaulting to "dynamic desirable." Seemingly, when you connect two brand new 2950 switches with a crossover cable, a trunk would form—and that is true. It also seems that because both switches are VTP servers, any VLAN information configured on either switch would be exchanged automatically with the other switch. As it turns out, with the default VTP server's VTP domain name of null, VTP does not send any VTP updates; instead, it waits until a domain name has been set. Therefore, to actually use VTP to dynamically distribute VLAN configuration information between these two switches, the following must happen:

- Trunking must be operational between the two switches. (With the default of "dynamic desirable," this should happen as soon as a crossover cable connects the two switches.)
- A VTP domain name must be defined on one of the VTP server switches.

If the preceding two conditions are met, the VTP server with its domain name set will send VTP advertisements. Interestingly, any VTP server or client switch with a null domain name actually updates its domain name based on the VTP update from the server. For instance, if you remove two brand new 2950s from their cardboard boxes, connect them with a crossover cable, and give switch1 a VTP domain name, switch2 will hear switch1's VTP update and change its domain name to match. A VTP client behaves the same way. *However, a good practice is to go ahead and set the VTP domain name on VTP clients and servers and to set VTP password on all switches.*

The Danger of Multiple VTP Servers

The support of multiple VTP servers introduces the possibility of either accidentally or purposefully eliminating the VLAN configuration for the network. When a VTP client or VTP transparent switch first connects to a switched network, using a trunk, it cannot destroy the VLAN configuration in other switches, because in these modes, the switch does not originate VTP updates. When a newly added VTP server switch attaches via trunk, however, it could replace the VLAN configuration of the other switches with its own—assuming the following are true about this new switch:

- The VTP server switch has the same VTP domain name as all of the other switches.
- The new VTP server switch has a higher revision number in VTP advertisement as compared with the existing switches.
- The new VTP server switch has the same password as the existing switches (if configured).

You can easily determine the revision number and VTP domain name with a Sniffer trace. If a password is used, the Sniffer sees only an MD5 hash of the password, preventing the password from being stolen. Therefore, VTP passwords you can use to prevent denial of service attacks with VTP, where a new switch could essentially delete all VLAN information.

Changing the Allowed VLAN List

Each LAN trunk has a feature called the *allowed VLAN* list. By default, the allowed VLAN list on 2950 switches includes VLANs 1 through 1005 when using standard software, or 1 through 4094 when using the more expensive enhanced software. The **show interfaces** *interface-id* **trunk** command displays the VLAN allowed list. For instance, Example 3-2 in Chapter 3 includes the **show interfaces trunk** command, and shows that the default VLANs 1 through 4094 are allowed.

You can configure switch trunk interfaces to disallow a particular VLAN from the VLAN allowed list on a trunk by using the following interface subcommand**:**

```
switchport trunk allowed vlan {add | all | except | remove} vlan-list
```

For instance, the **add** option permits the switch to add VLANs to the existing list, and the **remove** option permits the switch to remove VLANs from the existing list. The **all** option means all VLANs, so you can use it to reset the switch to its original default setting (permitting VLANs 1–1005 or 1–4094 on the trunk, depending on the software). The **except** option is rather tricky—it adds all VLANs to the list that are not part of the command. For instance, on an enhanced software switch, the **switchport trunk allowed vlan except 100-200** interface subcommand adds VLANs 1 through 99 and 201 through 4094 to the existing allowed VLAN list on that trunk.

Subnetting (Chapter 4)

To complete your CCNA certification, you need to have a fairly deep, confident, and fast ability to answer subnetting questions. With subnetting, you must be very accurate, as well as fast. When practicing, first make sure that you can consistently get the right answer. At that point, in my humble opinion, for the ICND exam, you should be able to answer the following style of questions in 45 seconds or less in order to lessen time pressure on the ICND exam. And for each of these, assume that any subnet mask supplied by the question has one octet that is not a 0 or 255.

- Given a subnet number and mask, what router command configures a router interface with the last valid IP address in the subnet?

- Given an IP address and mask, what is the subnet number?
- Given an IP address and mask, what is the range of valid IP addresses?
- Given a mask and a class (A, B, or C) for the network, into how many subnets can the network be divided?
- Given a mask and a class (A, B, or C) for the network, how many hosts are there on each subnet?
- Given a requirement of *x* hosts per subnet, a total of *y* subnets of the network, and a class (A, B, or C), which valid subnet masks meet this requirement?

  My suggestion of 45 seconds to answer these questions is indeed subjective. If such questions require you to take more than 45 seconds for thought and computation, I recommend that you spend some time improving your speed. The CD-only appendix that supplies 25 more subnetting practice questions to help you get faster.

  Expect questions that require you to perform multiple subnetting steps. You should be able to answer them within 60 to 90 seconds.

- Given a network diagram showing four Ethernets and three serial links, with five PCs shown with their IP addresses and masks, and with router IP addresses and masks listed, select an answer that shows the IP addressing misconfiguration of a single device.
- Figure out what happens in a network when an IP address or mask is misconfigured.
- In general, questions that require the math behind finding the subnet number, broadcast address, and range of valid IP addresses, for three to four subnets, to answer a single question.

  In short, know subnetting! Know it well, and do it confidently and fast.

Improvement for Finding the Last Valid Subnet

The section “What Are the Other Subnet Numbers?” in Chapter 4, “IP Addressing and Subnetting,” explains how to find the list of subnets, given a network number and a static mask used throughout the network. Another easy trick to find the last subnet number is as follows:

> For the last subnet number (called the broadcast subnet), the interesting octet’s value is the same number as the subnet mask’s value in that same octet.

For instance, for network 172.16.0.0, with mask 255.255.255.0, the first subnet (the zero subnet) is 172.16.0.0. The next subnet is 172.16.1.0, then 172.16.2.0, and so on, with the very last subnet (the broadcast subnet) being 172.16.255.0. Note that the interesting octet (third octet) of the last subnet number matches the subnet mask’s third octet. Consider another example. For network 192.168.1.0, with mask

255.255.255.192, the subnets are 192.168.1.0 (zero subnet), 192.168.1.64, 192.168.1.128, and 192.168.1.192 (last and broadcast subnet). The fourth (interesting) octet is 192 in both the last subnet number and in the subnet mask.

Number of Subnets: Subtract 2, or Not?

When choosing a subnet mask, the mask value implies the number of host bits and the number of subnet bits. The number of valid hosts per subnet is found using the formula $2^n - 2$, where n is the number of host bits. That formula is always true.

To find the number of possible subnets, however, you can use one of two formulas. Choosing which formula to use depends on what routing protocol is in use for that question. In short, if the question states or implies that a classful routing protocol is used, then use the formula $2^n - 2$, where n is the number of subnet bits. If the question states or implies that a classless routing protocol is used, use the formula of 2^{n}.

For a good reference of classless and classful routing protocols, and related terminology, refer to Table 7-3 in Chapter 7.

Table F-1 discusses the clues in the question that would imply which of the two formulas to use.

Table F-1 *When to Use Which Formula for the Number of Subnets*

If the question mentions…	Use the $2^n - 2$ rule when the questions says…	Use the 2^n rule when the questions says…
Classless or classful rules	Classful rules	Classless rules
Routing protocol being used	RIP version 1, IGRP	RIP version 2, EIGRP, OSPF
Whether VLSM is used	When VLSM is not used	When VLSM is used

Administrative Distance and Static Routes (Chapter 5)

Chapter 5 covers the concept of *administrative distance*, which is used by a single router when multiple routing protocols discover the same route. Table 5-14 in Chapter 5, "RIP, IGRP, and Static Route Concepts and Configuration," summarizes the default administrative distance values. Not mentioned in Chapter 5, however, is the fact that you can override administrative distance values in a couple of ways:

- A router can be configured to change a routing protocol's administrative distance via the **distance** routing protocol subcommand.
- The **ip route** command, which creates static routes, can set the administrative distance for the statically-configured route.

Example F-2 shows an example with the distance changed in several cases.

Example F-2 *Changing the Administrative Distance of a Route or Routing Protocol*

```
router eigrp 1
distance eigrp 121 170
!
ip route 172.31.101.0 255.255.255.0 172.31.3.2 11
ip route 172.31.102.0 255.255.255.0 serial0/0.2 12
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, EIGRP has been changed to use AD 121 for internal routes and
! AD 170 (the default) for external routes. Also, two static routes
! are shown, one referencing a next-hop IP address, and the other
! referencing an outgoing point-to-point subinterface. Each ip route
! command sets the AD at the end of the command.
! Below, the show ip route command lists the  updated AD values.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Router1#show ip route
Mar  1 00:22:40.050: %SYS-5-CONFIG_I: Configured from console by consoleute
Codes: C - connected, S - static, 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
       i - IS-IS, 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

     172.31.0.0/24 is subnetted, 10 subnets
C       172.31.130.0 is directly connected, Serial0/0.1
C       172.31.140.0 is directly connected, Serial0/0.2
C       172.31.3.0 is directly connected, Serial0/1
C       172.31.2.0 is directly connected, FastEthernet0/0
D       172.31.4.0 [121/2172416] via 172.31.3.2, 00:01:05, Serial0/1
D       172.31.103.0 [121/2195456] via 172.31.130.3, 00:01:05, Serial0/0.1
S       172.31.102.0 is directly connected, Serial0/0.2
S       172.31.101.0 [11/0] via 172.31.3.2
Router1#show ip route 172.31.102.0
Routing entry for 172.31.102.0/24
  Known via "static", distance 12, metric 0 (connected)
  Redistributing via eigrp 1
  Advertised by eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via Serial0/0.2
      Route metric is 0, traffic share count is 1
```

Note also that the **show ip route** command lists the administrative distance values for EIGRP and one of the static routes, but not the other static route. IOS considers any static routes that use the outgoing interface option, rather than the next-hop address, as being *directly connected* static routes, and in that case, the administrative distance is not displayed in the **show ip route** command output. To see the associated administrative distance for those routes, use the **show ip route** *subnet-number* command, as shown at the end of the example.

RIP Version 2 Configuration (Chapter 5)

As mentioned in the introduction to Chapter 5 of this book, Appendix D of the *ICND Exam Certification Guide* provides some background information on and compares regarding the popular interior IP routing protocols. (Appendix D is a reprint of Chapter 14 from the INTRO Exam Certification Guide; if you have not yet read Appendix D, or Chapter 14 from the INTRO Exam Certification Guide, definately do so before taking the exam.) Appendix D covers some information about RIP version 2, including comparisons with RIP version 1, but it does not include information regarding RIP version 2 configuration; the configuration basics are covered here.

To configure a router to use only RIP version 2, and not version 1, use the **version 2** router subcommand. Back in Chapter 5, Figure 5-11 and Example 5-7 display a network diagram and the related configuration, **show**, and **debug** commands with RIP version 1. Example F-3 that follows shows the RIP version 2 configuration for the same design, with two small changes. In this case, all the links are up, plus the subnet mask used on the LAN at Seville uses a /26 prefix rather than a /24 as shown in Chapter 5. By using a different mask, Example F-3 can show RIP version 2 working with VLSM. Example F-3 also shows a sample RIP **debug** on the Albuquerque router.

Example F-3 *RIP-2 Sample Configuration for Routers in Figure 5-11 from Chapter 5*

```
router rip
network 10.0.0.0
version 2
Albuquerque#debug ip rip
RIP protocol debugging is on
00:36:04: RIP: received v2 update from 10.1.4.252 on Serial0
00:36:04:      10.1.2.0/24 -> 0.0.0.0 in 1 hops
00:36:04:      10.1.5.0/24 -> 0.0.0.0 in 1 hops
00:36:04:      10.1.3.192/26 -> 0.0.0.0 in 2 hops
00:36:08: RIP: sending v2 update to 224.0.0.9 via Serial0 (10.1.4.251)
00:36:08:      10.1.1.0/24 -> 0.0.0.0, metric 1, tag 0
00:36:08:      10.1.6.0/24 -> 0.0.0.0, metric 1, tag 0
00:36:08:      10.1.3.192/26 -> 0.0.0.0, metric 2, tag 0
```

Note a couple of important items in the **debug** output of Example F-3. RIP version 2 sends updates to multicast IP address 224.0.0.9, as opposed to a broadcast IP address of 255.255.255.255 used by RIP version 1. By multicasting the RIP-2 updates, RIP-2 allows the devices that are not using RIP-2 to ignore the updates and not waste processing cycles. The **debug** messages show the subnet masks as part of the routing updates, in prefix style, reaffirming RIP-2's support of VLSM. Note that Seville's Ethernet subnet (10.1.3.192/26) is shown, with the correct subnet mask.

Configuring routers to use only RIP-1, or only RIP-2, is easy—just ignore or include the **version 2** subcommand, respectively. However, IOS can run both RIP versions in the same internetwork, using the **ip rip send version** interface subcommand. Essentially, the configuration tells the router whether to send RIP-1 style updates, RIP-2 style updates, or both, for each interface. For instance, in that same Figure 5-11 from Chapter 5 again, suppose that the Yosemite router (lower left) will use RIP-1, and the other two routers will use RIP-2. Examples F-4 and F-5 show the configurations on Albuquerque and Yosemite to accomplish the feat.

Example F-4 *Configuration on Albuquerque for RIP-Version Coexistence*

```
interface ethernet 0
ip addr 10.1.1.251 255.255.255.0
interface serial 0
ip addr 10.1.4.251 255.255.255.0
ip rip send version 1
ip rip receive version 1
interface serial 1
ip address 10.1.6.251 255.255.255.0
!
router rip
network 10.0.0.0
version 2
```

Example F-5 *Configuration on Yosemite for RIP-Version Coexistence*

```
interface ethernet 0
ip addr 10.1.2.252 255.255.255.0
interface serial 0
ip addr 10.1.4.252 255.255.255.0
interface serial 1
ip address 10.1.5.252 255.255.255.0
!
router rip
network 10.0.0.0
```

The RIP-2 configuration logic works just like RIP-1—RIP-2 updates are sent and received on each interface that is matched by a **network** command. Because Yosemite will send and receive only RIP-1 updates, however, the other two routers need the appropriate interface subcommands to tell the router to send and receive RIP-1 updates to and from Yosemite. For instance, on Albuquerque, with the **version 2** command configured, the router sends RIP-2 updates and expects to receive only RIP-2 updates. The **ip rip send version 1** and **ip rip receive version 1** commands override those defaults, telling IOS to both send and receive RIP-1 updates on that interface.

IGRP and EIGRP Metric Comparison (Chapters 5 and 6)

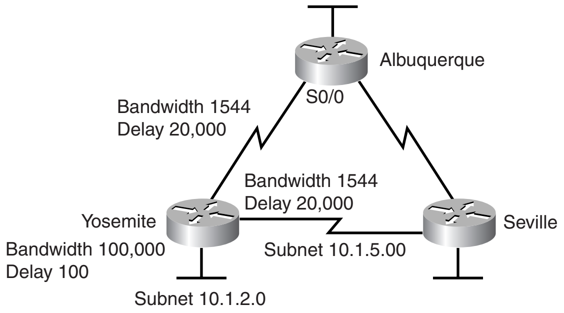
IGRP computes its metric based on the bandwidth, delay, reliability, load, and maximum transmission unit (MTU). With default settings, only the bandwidth and delay are considered, and the calculation reduces to the following:

$$\text{Metric} = \frac{10^7}{\text{least-bandwidth (kbps)}} + \text{cumulative delay}$$

In this formula, the term *least-bandwidth* represents the lowest-bandwidth link in the route; this value uses a unit of kilobits per second. For instance, if the slowest link in a route is a 10 Mbps Ethernet link, the first part of the formula is $10^7 / 10^4$, which equals 1000, because 10 Mbps is equal to 10,000 kbps (10^4 kbps). The delay value used by the formula is the sum of all the delay values for all links in the route, with a unit of "tens of microseconds." You can set both bandwidth and delay for each link, using the cleverly named **bandwidth** and **delay** interface subcommands.

Example F-6 shows an example of the bandwidth, delay, and calculated metrics. The output comes from the familiar three-router network, as shown in Figure F-1. In this case, all links are up, plus EIGRP is in use. That means that the calculation includes an additional multiplier of 256, essentially making the formula EIGRP_metric = IGRP_metric * 256.

Figure F-1 *Network Showing Bandwidth and Delay for EIGRP Metric Calculation*



Example F-6 *EIGRP Metric Calculation Example (Albuquerque)*

```
Albuquerque#show interfaces s 0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.1.4.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
! lines omitted for brevity
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, the show interfaces command shows the bandwidth and delay
! values for Albuquerque's S0/0 interface. Note that the bandwidth
! is shown in Kbps, and the delay in microseconds.
! Below, the show ip route command lists the calculated metrics.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Albuquerque#show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, 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

     10.0.0.0/24 is subnetted, 5 subnets
D       10.1.2.0 [90/2172416] via 10.1.4.2, 00:10:54, Serial0/0
C       10.1.1.0 is directly connected, Ethernet0/0
C       10.1.6.0 is directly connected, Serial0/1
D       10.1.5.0 [90/2681856] via 10.1.4.2, 00:10:54, Serial0/0
C       10.1.4.0 is directly connected, Serial0/0
```

First, consider the route from Albuquerque to 10.1.2.0/24. The constraining bandwidth is 1544, meaning 1544 kbps, on Albuquerque's S0/0 interface. So, the first part of the formula is 10^7 / 1544, rounded down to an integer value of 6476. Next, the cumulative delay is added, with a unit of "tens of microseconds." Note that the **show interfaces** command output lists delay in microseconds. So, Albuquerque's delay of 20,000 microseconds counts as a delay of 2000, and Yosemite's fa0/0 delay of 100 microseconds counts as a delay of 10. So, the "cumulative delay" in the formula is 2000 + 10 = 2010, which is added to the 6476 calculated for bandwidth—giving a total of 8486. Finally, because EIGRP is used, the formula multiplies 8486 times 256, giving 2,172,416, which is the metric shown in Example F-6 for subnet 10.1.2.0.

The calculation for Albuquerque's route to 10.1.5.0 through Yosemite follows the same kind of logic, with the constraining bandwidth still being 1544 kbps, but a cumulative delay total of 4000: 2000 from Albuquerque's S0/0 interface plus 2000 more from Yosemite's S0/1 interface. So, the formula computes to the following:

$$(6476 + 2000 + 2000) * 256 = 2,681,856$$

The result is the same number shown for subnet 10.1.5.0/24 in the **show ip route** command output in Example F-6.

Finally, although neither IGRP nor EIGRP uses hop count in its calculations, they both actually include a hop count mechanism to help prevent loops. You might recall that RIP allows a maximum valid hop count of 15, with 16 implying an infinite metric (or failed) route. Similarly, IGRP has a maximum valid hop count of 255, and EIGRP uses a maximum of 224, with both values defaulting to 100, although neither hop-count value is used to calculate the metric.

Route Summarization Terminology (Chapter 7)

Chapter 7 covers a topic described as *route summarization*, as well as a topic called *autosummarization*. Some texts use the term *manual summarization* to mean the same thing as what Chapter 7 calls *route summarization*. The use of the word *manual* emphasizes a key difference compared with autosummarization, because manual summarization relies on the explicit configuration of the routing protocol to know exactly what summarized route to advertise. Conversely, with autosummarization, the routing protocol automatically figures out the classful network to summarize, plus autosummarization is enabled by default for those routing protocols that support it. Table F-2 lists the routing protocols and some of the key details of support for manual summarization and autosummarization.

Table F-2 *Manual Summarization and Autosummarization Summary*

Feature	RIP-1	IGRP	RIP-2	EIGRP	OSPF
Supports autosummarization	Yes	Yes	Yes	Yes	No
Autosummarization can be disabled	No	No	Yes	Yes	N/A
Supports Manual Route Summarization	No	No	Yes	Yes	Yes

ISDN DDR Names and PPP CHAP Authentication (Chapter 10)

Most dial-on-demand routing (DDR) implementations also use Point-to-Point Protocol (PPP) Challenge Handshake Authentication Protocol (CHAP) for authentication. When a router uses DDR to dial multiple different sites, good security practices suggest that each remote site should use a different CHAP name and password. This section reviews some of the interrelationships between DDR configuration and the associated CHAP configuration.

Imagine a router named Core, with 10 branch routers named BR1, BR2, and so on. Router Core's configuration includes 10 **username** commands, one for each branch router, for example, **username BR1 password fred**. Then, when router Core dials BR1's phone number, router Core could be configured to only accept the user name BR1 for that call; thus, you configure better security than if router Core accepts any of the names configured in the 10 **username** commands.

You can configure a router to require a particular name when calling each particular site when using two of the three styles of DDR configuration shown back in Chapter 10, "ISDN and Dial-on-Demand Routing." First, when using legacy DDR with the **dialer map** command, you can configure the rquired name using the **dialer map** interface subcommand. Additionally, when using DDR dialer profiles, you can configure the required name with the **dialer remote-name** dialer interface subcommand. (The configuration option to use Legacy DDR with the **dialer string** command does not provide a way to define what name the remote site should supply.)

Example 10-5 in Chapter 10 shows a classic case of configuring the name required when dialing each site, along with the related PPP CHAP configuration. The following list shows the progression of events when a dial is made in the network explained in Example 10-5, and shows how IOS uses the names configured in the **dialer map**, **dialer remote-name**, and **username** commands of that example.

1. SanFrancisco dials 14045551234 based on some interesting traffic occurring.
2. The ISDN call is set up.

3. PPP then begins initialization, including CHAP, with LosAngeles sending its CHAP username, random number, and hash back to SanFrancisco. (You might want to refer back to Chapter 9, "Point-to-Point Leased-Line Implementation," for a review of CHAP.)

4. SanFrancisco's logic first checks to make sure the CHAP name is LosAngeles because of the *name* parameter in the **dialer map broadcast name LosAngeles 14045551234** command.

5. SanFrancisco then performs normal PPP CHAP processing of the received information, using SanFrancisco's **username LosAngeles password Clark** command to find the password associated with the name LosAngeles.

6. If the CHAP processing completes, PPP continues and completes initialization of PPP LCP, along with any Layer 3 control protocols such as IP Control Protocol (IPCP).

For dialer profiles, the same steps occur, except that the name checked at Step 4 is configured with the **dialer remote-name** dialer interface subcommand. (See Example 10-10 in Chapter 10.)

NOTE For anyone with a book that is printing 4 or earlier: Example 10-10 was updated due to errors. Go to http://www.ciscopress.com/title/158720083X, click the **Errata** link in the **More Information** box, and download the errata file to get the latest and most accurate Example 10-10.

Interestingly, the **dialer map** *name* parameter can be omitted, essentially skipping Step 4 in the preceding process. Likewise, when configuring dialer profiles, omitting the **dialer remote-name** command also essentially skips Step 4's logic.

NOTE The names and passwords used with DDR and PPP CHAP are case-sensitive.

The DDR process is rather involved. You can find examples of many of the most useful **show** and **debug** commands related to DDR in Chapter 10 in the section titled "ISDN and DDR **show** and **debug** Commands," particularly in Example 10-7. Example F-7 adds a few more commands that are not in Example 10-7. In particular, note the **debug ppp negotiation** command, which supplies messages relating to all facets of PPP negotiation, and **debug ppp authentication**, which only lists CHAP or PAP-related messages.

Example F-7 follows the same style as Chapter 10's Example 10-7 in that one **debug** is enabled, the dial is made and then taken down, and then the dial is repeated with only the other **debug** setting enabled. By doing so, you can see exactly what messages are generated by each debug.

Example F-7 *Additional DDR Troubleshooting Commands*

```
SanFrancisco#debug ppp negotiation
PPP protocol negotiation debugging is on
SanFrancisco#ping 172.16.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 122.16.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, a debug is enabled, and the dial trigger with the ping command.
! Below, the debug messages start with PPP, then LCP, then followed by the CHAP
! related messages. Note that challenge and response messages are sent in pairs,
! one for each direction.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
00:43:05: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
00:43:05: BR0/0:1 PPP: Using dialer call direction
00:43:05: BR0/0:1 PPP: Treating connection as a callout
00:43:05: BR0/0:1 PPP: Phase is ESTABLISHING, Active Open [0 sess, 0 load]
00:43:05: BR0/0:1 LCP: O CONFREQ [Closed] id 5 len 15
00:43:05: BR0/0:1 LCP:    AuthProto CHAP (0x0305C22305)
00:43:05: BR0/0:1 LCP:    MagicNumber 0x07AD731A (0x050607AD731A).
00:43:07: BR0/0:1 LCP: I CONFREQ [REQsent] id 3 len 15
00:43:07: BR0/0:1 LCP:    AuthProto CHAP (0x0305C22305)
00:43:07: BR0/0:1 LCP:    MagicNumber 0x0272DFDC (0x05060272DFDC)
00:43:07: BR0/0:1 LCP: O CONFACK [REQsent] id 3 len 15
00:43:07: BR0/0:1 LCP:    AuthProto CHAP (0x0305C22305)
00:43:07: BR0/0:1 LCP:    MagicNumber 0x0272DFDC (0x05060272DFDC)
00:43:07: BR0/0:1 LCP: TIMEout: State ACKsent
00:43:07: BR0/0:1 LCP: O CONFREQ [ACKsent] id 6 len 15
00:43:07: BR0/0:1 LCP:    AuthProto CHAP (0x0305C22305)
00:43:07: BR0/0:1 LCP:    MagicNumber 0x07AD731A (0x050607AD731A)
00:43:07: BR0/0:1 LCP: I CONFACK [ACKsent] id 6 len 15
00:43:07: BR0/0:1 LCP:    AuthProto CHAP (0x0305C22305)
00:43:07: BR0/0:1 LCP:    MagicNumber 0x07AD731A (0x050607AD731A)
00:43:07: BR0/0:1 LCP: State is Open
00:43:07: BR0/0:1 PPP: Phase is AUTHENTICATING, by both [0 sess, 0 load]
00:43:07: BR0/0:1 CHAP: O CHALLENGE id 3 len 28 from "SanFrancisco"
00:43:07: BR0/0:1 CHAP: I CHALLENGE id 3 len 27 from "LosAngeles"
00:43:07: BR0/0:1 CHAP: O RESPONSE id 3 len 28 from "SanFrancisco"
00:43:07: BR0/0:1 CHAP: I SUCCESS id 3 len 4
00:43:07: BR0/0:1 CHAP: I RESPONSE id 3 len 27 from "LosAngeles"
00:43:07: BR0/0:1 CHAP: O SUCCESS id 3 len 4
00:43:07: BR0/0:1 PPP: Phase is UP [0 sess, 0 load]
```

Example F-7 *Additional DDR Troubleshooting Commands (Continued)*

```
00:43:07: BR0/0:1 IPCP: O CONFREQ [Closed] id 3 len 10
00:43:07: BR0/0:1 IPCP:    Address 172.16.2.2 (0x0306C00A0001)
00:43:07: BR0/0:1 CDPCP: O CONFREQ [Closed] id 3 len 4
00:43:07: BR0/0:1 IPCP: I CONFREQ [REQsent] id 3 len 10
00:43:07: BR0/0:1 IPCP:    Address 172.16.2.1 (0x0306C00A0002)
00:43:07: BR0/0:1 IPCP: O CONFACK [REQsent] id 3 len 10
00:43:07: BR0/0:1 IPCP:    Address 172.16.2.1 (0x0306C00A0002)
00:43:07: BR0/0:1 CDPCP: I CONFREQ [REQsent] id 3 len 4
00:43:07: BR0/0:1 CDPCP: O CONFACK [REQsent] id 3 len 4
00:43:07: BR0/0:1 IPCP: I CONFACK [ACKsent] id 3 len 10
00:43:07: BR0/0:1 IPCP:    Address 172.16.2.1 (0x0306C00A0001)
00:43:07: BR0/0:1 IPCP: State is Open
00:43:07: BR0/0:1 CDPCP: I CONFACK [ACKsent] id 3 len 4
00:43:07: BR0/0:1 CDPCP: State is Open
00:43:07: BR0/0 IPCP: Install route to 172.16.2.1
00:43:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, the last two highlighted lines show the completion of higher-layer PPP
! protocols IPCP and CDPCP.
! Below, show dialer interface lists signaling info under "BRI0/0", which is
! the D channel; the current call under "BRI0/0:1", which is one of the B
! channels; and "BRI0/0:2, which is the other B channel, which is currently idle.
! It also shows the reason why the dial was made.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
SanFrancisco#show dialer interface bri0/0

BRI0/0 - dialer type = ISDN

Dial String      Successes   Failures    Last DNIS   Last status
14045551234              2          0    00:00:07       successful
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0/0:1 - dialer type = ISDN
Idle timer (150 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=172.16.2.2, d=172.16.2.1)
Time until disconnect 146 secs
Connected to 14045551234 (LosAngeles)

BRI0/0:2 - dialer type = ISDN
Idle timer (150 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Next, but not shown, the dial is taken down, and the old debug disabled.
```

continues

Example F-7 *Additional DDR Troubleshooting Commands (Continued)*

```
! Below, a new debug is enabled, and a new dial triggered. Note that the messages
! are the same ones shown with debug ppp negotiation, but in this case, only
! authentication messages are shown.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
SanFrancisco#debug ppp authentication
PPP authentication debugging is on
SanFrancisco#ping 172.16.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:

00:39:33: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
00:39:33: BR0/0:1 PPP: Using dialer call direction
00:39:33: BR0/0:1 PPP: Treating connection as a callout.
00:39:35: BR0/0:1 CHAP: O CHALLENGE id 1 len 28 from "SanFrancisco"
00:39:35: BR0/0:1 AUTH: Started process 0 pid 99
00:39:35: BR0/0:1 CHAP: I CHALLENGE id 1 len 27 from "LosAngeles"
00:39:35: BR0/0:1 CHAP: O RESPONSE id 1 len 28 from "SanFrancisco"
00:39:35: BR0/0:1 CHAP: I SUCCESS id 1 len 4
00:39:35: BR0/0:1 CHAP: I RESPONSE id 1 len 27 from "LosAngeles"
00:39:35: BR0/0:1 CHAP: O SUCCESS id 1 len 4.
```

Clarifications on Frame Relay Local and Global DLCI Addressing (Chapter 11)

Chapter 11, "Frame Relay," explains the fact that data-link connection identifiers (DLCIs) are locally significant. In other words, DLCIs have meaning only on the local access link between a router (Frame Relay DTE) and the Frame Relay switch (Frame Relay DCE) to which it connects. However, some Frame Relay providers assign DLCIs such that you can think of them as global values, more like LAN MAC addresses. These details are described in the section "DLCI Addressing Details," in Chapter 11.

When taking the Cisco exams, if a figure for a question shows three or more routers, you should be able to easily decide whether the figure implies local or global DLCI values. For instance, Figure 11-10 (in Chapter 11) shows a main site with four permanent virtual circuits (PVCs), one to each remote site. However, only one DLCI is shown beside the main site router, implying the use of global addressing. If local DLCIs were used, the figure would need to show a DLCI per PVC beside the main site router. Figure 11-7 depicts how local DLCIs are typically shown, with a DLCI beside each PVC.

In cases where a figure for a question shows only two routers, the figure might not imply whether local or global DLCI addressing is used. In those cases, look for clues in the question, answers, and any configuration. The best clues relate to the following fact:

> On any given router, only local DLCI values are in the configuration or **show** commands.

For instance, from Chapter 11, Figure 11-19 shows global DLCIs, with DLCI 51 beside the Atlanta router. However, the **frame-relay interface-dlci** commands in Example 11-8 and the Atlanta **show** commands in Example 11-12 list DLCIs 52, 53, and 54. Although Figure 11-19 makes it obvious that global addressing is used, even if only two routers had been shown, the **show** commands and configuration commands could have helped identify the correct DLCIs to use.

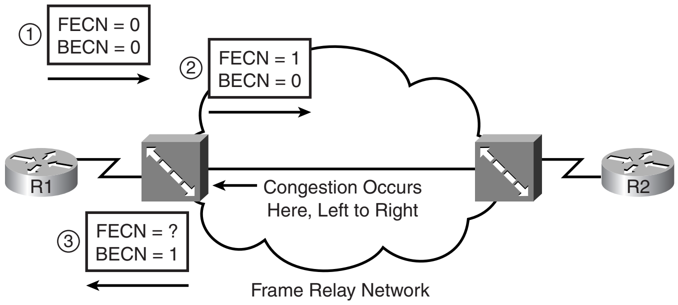
Frame Relay FECN and BECN (Chapter 11)

If a router has a T1 Frame Relay access link, but only a 128-kbps committed information rate (CIR), the router can send a lot more data into the Frame Relay network than the business contract with the Frame Relay provider allows. The Frame Relay provider can forward traffic beyond the 128-kbps CIR rate, but they guarantee to forward an average of only 128 kbps. So, IOS includes a feature called *Traffic Shaping*, which enables a router to send some packets, wait, send more, wait again, and so on, so that the average sending rate is slower than the actual clock rate of the interface. For instance, with a T1 access link and a 128-kbps CIR, Traffic Shaping could be defined to only send an average of 256 kbps over time. The idea is that the Frame Relay provider will probably discard a lot of traffic if the router averages sending data at close to T1 speed, but maybe the provider will not discard any traffic if the average rate is only 256 kbps.

You can set Traffic Shaping to use a single speed, or to adapt and range between two speed settings. When configured to adapt between two speeds, if the network is not congested, the higher speed is used; when the network is congested, the router adapts so that it shapes using the lower rate.

To adapt the shaping rates, the routers need a way to know whether congestion is occurring—and that's where Forward Explicit Correction Notification (FECN) and Backward Explicit Correction Notification (BECN) are used. Figure F-2 shows the basic use of the FECN and BECN bits.

Figure F-2 *Basic Operation of FECN and BECN*



FECN and BECN are bits in the Frame Relay header. At any point—either in a router or inside the Frame Relay cloud—a device can set the FECN bit, meaning that this frame itself has experienced congestion. In other words, there is congestion in the forward direction of that frame. In Figure F-2, at Step 1, the router sends a frame, with FECN=0. The Frame Relay switch notices congestion, and sets FECN=1 at Step 2.

The goal of the whole process, however, is to get the sending router—R1 in this figure—to slow down. So, knowing that it set FECN in a frame at step 2 in the figure, the Frame Relay switch *can* set the BECN bit in the next frame going back to R1 on that virtual circuit (VC), shown as Step 3 in the figure. The BECN tells R1 that congestion occurred in the direction opposite, or backward, from the direction of the frame—in other words, that congestion occurred for the frame sent by R1 to R2. R1 can then choose to slow down (or not), depending on how Traffic Shaping is configured.

Interface Status (Various Chapters)

Several IOS commands, most notably the **show interfaces** command, supply two words or phrases to identify whether an interface is working. To be functional, the command output must list "up and up." This short section covers a few scenarios that affect the settings of these two indicators.

Although not always true, the first word or phrase typically refers to the OSI Layer 1 status, with the second word or phrase referring to the Layer 2 status. For instance, the first indicator would list "down" when

- There is no cable installed.
- The other end of the cable is not connected to anything else.
- The other end of the cable is installed, but the interface on the other devices is administratively disabled—for example, a router could be cabled to the switch, but the switch might have a **shutdown** command on that interface.
- For a serial link, the CSU/DSU might be installed and configured, but the link from the phone company might be down.

  The first indicator can be "up," but the second indicator might be "down," under several conditions, including the following:

- Mismatched configuration for the Data Link protocol. (For instance, for a router and switch using Ethernet, one might have been configured for ISL trunking, whereas the other had not.)
- For a serial link, one end might be configured for Frame Relay, the other defaulting to HDLC.

- For point-to-point serial links created with a DCE and DTE cable, a missing **clock rate** command on the DCE side.
- For Frame Relay, the access link might be up, but the Frame Relay switch and router might be using different LMI protocols.
- For PPP, CHAP authentication failure, which causes PPP LCP to fail.
- Anything that causes a router to cease to hear keepalive messages on the link.

The Configuration Register and the Boot Sequence (Chapter 7 of the *CCNA INTRO Exam Certification Guide*)

For those who have the *CCNA ICND Exam Certification Guide* but not the *CCNA INTRO Exam Certification Guide*, this section contains a short description of the configuration register and boot process of routers. The following text is derived from the *CCNA INTRO Exam Certification Guide*, Chapter 7.

The *configuration register* is a 16-bit software register in a router, and its value is set using the **config-register** global configuration command. This register tells the router at boot time whether to use a full-featured IOS, ROMMON (a rudimentary small OS with no IP routing function), or RXBOOT (a limited-feature IOS with some IP capabilities). In fact, only the low-order 4 bits of the configuration register direct the router as to what IOS to load; because of that, the last 4 bits are together called the *boot field* of the configuration register. If the boot field is hex 0, ROMMON is loaded. If the boot field is hex 1, RXBOOT mode is used. For anything else, the router attempts to load a full-featured IOS.

Other configuration register bits are interesting as well. (The configuration register's bits are numbered right to left, 0 through 15, so bit 0 is on the right when viewed in Cisco documents.) You can set the speed of the console port to a speed other than the default of 9600 bps. Also, with a value of 0x2142, bit 6 is set—which tells the router to ignore the startup-config file in NVRAM upon next reboot. Ignoring the startup-config file is a key part of the password-recovery process. (For more information on password recovery, go to Cisco.com and search on "password"; typically the first URL listed contains password recovery procedures.) You can also browse to the URL http://www.cisco.com/warp/customer/474/. Note that these web pages require a valid login.

At boot time, assuming the boot field of the configuration register is not a hex 0 or 1, the router will attempt to load a fully functional IOS image. To pick the IOS to load, the router then looks for **boot system** configuration commands in the startup configuration file. If there are no **boot system** commands, the router takes the default action, which is to load the first IOS file in Flash memory. Table F-3 summarizes how a router uses the various options for the **boot system** command.

Table F-3 *Impact of the* ***boot system*** *Command on Choice of IOS*

Boot System Commands	Result
No **boot system** command	Tries loading the following, in order: first file in Flash memory; broadcasts looking for TFTP server and a default filename; IOS in ROM; or uses ROM Monitor.
boot system ROM	IOS from ROM is loaded.
boot system flash	The first file from Flash memory is loaded.
boot system flash *filename*	IOS with the name filename is loaded from Flash memory.
boot system tftp *filename* **10.1.1.1**	IOS with the name filename is loaded from the TFTP server.
Multiple boot system commands, any variety	An attempt occurs to load IOS based on the first boot command in the configuration. If that fails, the second boot command is used, and so on, until one succeeds.