APPENDIX D

INTRO 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 INTRO 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/1587200945.

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 see if 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 online at the book's companion website. To do so do the following:

1. Browse to http://www.ciscopress.com/title/1587200945.
2. Select the **Downloads** option under the **More Information** box.

3. Download the latest "INTRO Appendix D" document

NOTE Note that the downloaded document has a version number. Compare the version of this print Appendix D (Version 1.0) with the latest online version of this appendix and do the following:

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

Ethernet NICs (Chapter 3)

Ethernet network interface cards (NICs) are used to allow a computer to connect to an Ethernet network. Today, many people take for granted that a PC will have an Ethernet NIC, and that it will have a female RJ-45 connector, and that the PC's operating system (OS) will have the proper networking software installed. However, you should consider several factors when choosing an Ethernet NIC:

- **System bus of the PC**—The Ethernet NIC must be compatible with the bus used inside the PC. PCI bus cards are typical today, or cardbus NICs for removable PC cards, common in laptops.
- **Cable type and connector**—Many NICs are ready for the ubiquitous RJ-45 connector with UTP copper cabling. Other NICs expect optical cabling, or use the older access unit interface (AUI), a 9-pin connector, or even the older Bayonet Neill Concelman (BNC) round connector, similar to a CATV connector, which was popular with 10BASE2.

You should also make sure that the PC has the right networking software installed. For the last 5 years plus, almost every OS already had TCP/IP installed, so the networking software and protocols on the computer were already there.

Forwarding Devices and OSI Layers (Chapters 3, 4, and 5)

Forwarding devices typically use logic equivalent to OSI Layers 1, 2, or 3 when forwarding traffic. You have already read about the details in this book, assuming you've finished reading it. Table D-1 summarizes these details in one place.

Table D-1 *Forwarding Devices and OSI Layers*

Device…	OSI Layer	Function
Ethernet repeater	1	Regenerates a new clean square-wave as it forwards all incoming electrical signals.
Ethernet hub	1	Acts as a repeater, but typically has many more physical ports than a repeater.

Table D-1 *Forwarding Devices and OSI Layers (Continued)*

Device…	OSI Layer	Function
Ethernet bridge	2	Listens, learns MAC addresses, makes forwarding/filtering decisions based on Layer 2 Ethernet addresses.
Ethernet switch	2	Same as a bridge, with optimized internal logic and hardware forwarding using ASICs.
Router	3	Forwards packets based on the Layer 3 destination address.
Layer 4 switch	4	Forwarding decisions based in part on the Layer 4 header.
Layer 5–7 switch	5–7	Forwarding decisions based in part on Layers 5–7.
Multilayer switch	Many	Varies, but it is a device capable of the logic for forwarding at Layer 2, 3, and possibly 4–7.

The IP Protocol Field (Chapter 5)

The IP header has a 1-byte field called the Protocol field, which defines the type of header that follows the IP header. For instance, TCP and UDP are the two most popular transport layer protocols used by TCP/IP applications, and these two types of headers typically follow an IP header when examining the contents of a packet. To denote that a TCP header follows a particular IP header, the creator of the packet sets the IP Protocol field to 6. In other cases, the header after the IP header is not TCP or UDP, but some other protocol (such as ICMP) or a routing protocol header. Table D-2 lists some of the more popular values for the IP Protocol field.

Table D-2 *IP Protocol Field Values*

Protocol	IP Protocol Field Values
UDP	17
TCP	6
ICMP	1
EIGRP	88
OSPF	89

RIP and IGRP do not have a protocol number, because those two routing protocols send their messages inside UDP segments.

Banners (Chapter 7)

Cisco routers and switches can display a variety of banners depending on what a router or switch administrator is doing. A banner is simply some text that displays on the screen for

the user. You can be configure a router to display multiple banners, some before login and some after. Table D-3 lists the three most popular banners and their typical use.

Table D-3 *Banners and Their Use*

Banner	Typical use
Message of the Day (MOTD)	Shown before login prompt; for temporary messages that may change from time to time—for example, "Router1 down for maintenance at midnight."
Login	Shown before login prompt but after the MOTD banner; for permanent messages—for instance, "Unauthorized Access Prohibited."
Exec	Shown after login prompt; for information desired to be kept from unauthorized users.

Example 7-3 in Chapter 7 shows an example of the banner command, with some explanation as to the nuances of its use. Additionally, the following example (Example D-1) shows all three types of banners from Table D-3, with a user login that shows the banners in use.

Example D-1 *Banner Configuration*

```
AS(config)#banner motd # This is banner motd line 1
Enter TEXT message.  End with the character '#'.
this banner tends to have temporary info, eg router down at midnight
this is line three, ending in the delimeter #
AS(config)#banner login $ This is banner login, line 1, different delimiter
Enter TEXT message.  End with the character '$'.
this banner tends to be permanent info, eg stay out of my router
This is line three, ending with the delimeter $
AS(config)#banner exec z This is banner exec line 1, with an unusual delimiter
Enter TEXT message.  End with the character 'z'.
this banner tends to have info only for authorized users
this is exec banner line 3 z
AS(config)#^Z
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, the three banners are created in configuration mode. Note that any
! delimiter can be used, as long as the character is not part of the message
! text.
! Below, the user of this router quits the console connection, an logs back in,
! seeing the motd and login banners, then the password prompt, and then the
! exec banner. Note that the user typed in a password at the password prompt,
! but it is not displayed on the screen - so the password is not displayed in the
! text below, either.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
AS#quit


AS con0 is now available
```

Example D-1 *Banner Configuration (Continued)*

```
Press RETURN to get started.



This is banner motd line 1
this banner tends to have temporary info, eg router down at midnight
this is line three, ending in the delimeter
This is banner login, line 1, different delimiter
this banner tends to be permanent info, eg stay out of my router
This is line three, ending with the delimeter

User Access Verification

Password:

This is banner exec line 1, with an unusual delimeter
this banner tends to have info only for authorized users
this is exec banner line 3
```

Enable Secret and Password Encryption (Chapter 7)

The **enable** command moves you from user EXEC mode (with a prompt of **hostname>**) to privileged EXEC mode (with a prompt of **hostname#**). A router or switch can be configured to require a password according to the following rules:

- If the global configuration command **enable password** *actual-password* is used, it defines the required "enable password." This password is listed as *clear text* in the configuration file.
- If the global configuration command **enable secret** *actual-password* is used, it defines the required "enable password." This password is listed as an *encrypted value* in the configuration file.
- If *both commands* are used, the password set in the **enable secret** command defines which password is required.

When the **enable secret** command is configured, the router or switch automatically encrypts the password. Example D-2 shows an example of the creation of the **enable secret** command, its format, and its deletion.

Example D-2 *Encryption and the* **enable secret** *Command*

```
Router3(config)#enable secret ?
  0       Specifies an UNENCRYPTED password will follow
  5       Specifies an ENCRYPTED secret will follow
  LINE    The UNENCRYPTED (cleartext) 'enable' secret
```

continues

Example D-2 *Encryption and the* **enable secret** *Command (Continued)*

```
  level  Set exec level password

Router3(config)#enable secret fred
Router3(config)#^Z
Router3#show running-config
! all except the pertinent line has been omitted!
enable secret 5 $1$ZGMA$e8cmvkz4UjiJhVp7.maLE1

Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#no enable secret fred
Router3(config)#^Z
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! The enable secret command has been removed from the running config
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

By using the (recommended) **enable secret** command, rather than the **enable password** command, the enable password is automatically encrypted. Example D-2 uses the **enable secret fred** command, setting the password text to fred. However, the syntax **enable secret 0** fred could have been used, with the **0** implying that the password that followed was clear text. The **show running-configuration** command in Example D-3 shows how the password is stored in the configuration file, with the *5* representing the fact that the password is encrypted, and the gobbledy-gook long text string being the encrypted password.

Thankfully, to delete the enable secret password, you can simply use the **no enable secret** command, and type in the clear-text password. For instance, in Example D-2, the command **no enable secret fred** completely deletes the enable secret password. Also, to change the enable secret password, you could use the **enable secret** *another-password* command, with *another-password* simply meaning that you put in a new text string for the new password.

You can encrypt console and vty passwords using the **service password-encryption** global configuration command. The actual passwords are still set using the same **password** commands described in Chapter 7, "Operating Cisco Routers." The presence or absence of the **service password-encryption** global configuration command dictates if the passwords are encrypted or not, as follows:

- When the **service password-encryption** command has been configured, all existing console and vty passwords are immediately encrypted.
- If the **service password-encryption** command has already been configured, any future changes to the passwords are encrypted.

- If the **no service password-encryption** command is later used, the passwords remain encrypted, until they are changed—at which point they show up in clear text.

Example D-3 shows this behavior.

> **NOTE** The **show running-config | begin line vty** command lists the running configuration, beginning with the first line that contains the text **line vty**. This is just a shorthand way to see a smaller part of the running configuration.

Example D-3 *Encryption and the* **service password-encryption** *Command*

```
Router3#show running-config | begin line vty
line vty 0 4
 password cisco
 login
Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#service password-encryption
Router3(config)#^Z
Router3#show running-config | begin line vty
line vty 0 4
 password 7 070C285F4D06
 login
end
Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#no service password-encryption
Router3(config)#^Z
Router3#show running-config | begin line vty
line vty 0 4
 password 7 070C285F4D06
 login
end
Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#line vty 0 4
Router3(config-line)#password cisco
Router3(config-line)#^Z
Router3#show running-config | begin line vty
line vty 0 4
 password cisco
 login
```

Basic Switch Configuration (Chapter 8)

To complete your CCNA certification, you need to understand how to configure several features on a Cisco LAN switch—in particular, the configuration on the 2950 model series.

If you have the ICND Exam Certification Guide, take the time to read over the section "Typical Basic Administrative Configuration" in Chapter 1, "LAN Switching Review and Configuring Cisco 2950 LAN Switches." If you don't have that book, read over this short section, which is a direct excerpt of that section from the ICND book. Also, make sure that you have read Chapters 8 through 11 of this book first because that will help solidify some of the material mentioned here.

A Cisco switch will come up and work, with all ports in VLAN 1, without any configuration. However, you will typically want to configure something. Example D-4 shows a typical initial configuration session on a 2950 switch, along with some other commands that help point out what the configuration has accomplished.

Example D-4 *Basic Configuration of a 2950 Switch*

```
Switch>enable
Switch#
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname fred
fred(config)#enable secret cisco
fred(config)#line con 0
fred(config-line)#password barney
fred(config-line)#login
fred(config-line)#line vty 0 15
fred(config-line)#password wilma
fred(config-line)#login
fred(config-line)#interface fastethernet0/5
fred(config-if)#speed 100
fred(config-if)#duplex half
fred(config-if)#
00:23:49: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
  changed state to down
00:23:52: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
00:23:54: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
  changed state to up
fred(config-if)#
fred(config-if)#shutdown
fred(config-if)#
00:24:33: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to
  administratively down
00:24:34: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
  changed state to down
fred(config-if)#
fred(config-if)#no shutdown
fred(config-if)#
fred(config-if)#
00:24:42: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
```

Example D-4 *Basic Configuration of a 2950 Switch (Continued)*

```
00:24:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
  changed state to up
fred(config-if)#exit
fred(config)#interface vlan 1
fred(config-if)#ip address 10.1.1.1 255.255.255.0
fred(config-if)#no shutdown
00:25:07: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:25:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
  state to up
fred(config-if)#exit
fred(config)#ip default-gateway 10.1.1.254
fred(config)#^Z

fred#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]

fred#show startup-config
Using 1613 out of 393216 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname fred
!
enable secret 5 $1$sgBC$CWUWtIwBJ1G1zedlEIYr5/
!
spanning-tree extend system-id
!
interface FastEthernet0/1
 no ip address
!
interface FastEthernet0/2
 no ip address
!
interface FastEthernet0/3
 no ip address
!
interface FastEthernet0/4
 no ip address
!
interface FastEthernet0/5
 no ip address
```

continues

Example D-4 *Basic Configuration of a 2950 Switch (Continued)*

```
 duplex half
 speed 100
!
!
! Lines omitted for brevity
!
interface Vlan1
 ip address 10.1.1.1 255.255.255.0
!
ip classless
ip default-gateway 10.1.1.254
ip http server
!
line con 0
 password barney
 login
line vty 0 4
 password wilma
 login
line vty 5 15
 password wilma
 login
!
end

fred#quit

fred con0 is now available

Press RETURN to get started.

User Access Verification

Password:
fred>enable
Password:
fred#
```

Rather than just list the configuration commands, this example shows you everything that came across the screen when typing the commands in configuration mode.

The example begins with the user logging in to the switch. Because no configuration had been added at this point, the switch did not ask for a console password or enable password. Next, the user got into configuration mode, setting the name of the switch with the **hostname fred** command. Notice that the command prompt immediately changes to begin with **fred**,

because the prompt starts with the host name. This is just another proof that that the switch IOS accepts configuration commands immediately—so be careful out there!

Next, the user sets the **enable secret** password to **cisco**, the console password to **barney**, and the vty (Telnet) password to **wilma**. The **login** commands tell the switch to require a password at the console and for Telnet sessions, respectively, and the **password** commands tell it what passwords to expect. Often, the console and Telnet passwords are the same value, because both let you get into user mode; I used two different passwords in the example just to make the point that they can differ.

With this configuration, when a user Telnets to the switch, the switch prompts for a password, expecting **wilma**. Similarly, the switch prompts for a password at the console, expecting **barney**. Both methods put the user into user mode. To get into privileged mode, the user uses the **enable** command, and types the enable secret password of **cisco** when prompted, to get into privileged mode. (An example of that process at the console is shown at the very end of the example.)

NOTE The **enable secret** and **enable password** commands both define the password needed in order to get into enable mode. The **enable password** command also defines the enable password. If only one of these two commands is in the configuration, only the password defined by that command is used. If both are configured, the **enable secret** password is used. Why two commands? The **enable password** command came first, but even with encryption, breaking the password was easy to do. The **enable secret** command uses a hash algorithm to store the password value in the configuration, which makes breaking the password very difficult, and more secure.

Next, the user issues the **interface Fastethernet 0/5** command to enter interface configuration mode. While there, the **duplex** and **speed** commands tell the switch to force these settings, rather than use the autonegotiated settings. But the PC on the other end of the cable on interface fastethernet 0/5 had already negotiated for 100 Mbps, full-duplex—and the new duplex setting of half-duplex takes effect immediately! So, that interface will no longer work for a short time.

The messages that clutter the example, immediately after the changing of the speed and duplex settings, actually confirm that interface fastethernet 0/5 was temporarily unusable. The switch issues informational messages when events occur, and sends them to the console by default. So these messages tell you that the switch brought the interface down, because of the duplex mismatch. The next message tells you the interface is then back up again as a result of the switch and the device negotiating to use half duplex.

Next, the example just shows the basic operation of the **shutdown** and **no shutdown** commands. The **shutdown** command puts an interface in a "down" status administratively, so that the interface cannot pass traffic. The **no shutdown** command brings the interface back up. The example shows the informational messages that tell you that the interface has changed status after each command.

The switch needs an IP address in order to allow people to Telnet to the switch and to manage the switch. The switch also needs to know a "default gateway," just as an end-user PC would. The default gateway is the IP address of a router connected to the switch; the switch sends IP packets to that router in order to send them to IP hosts that are not on the LAN created by the switch.

To configure the IP address, you first use the **interface vlan 1** command, because the IP address of the 2950 switch is configured on that interface. Next, the **ip address** command sets the IP address and subnet mask. (IP subnet masks are covered Chapter 12, "IP Addressing and Subnetting.") Finally, the **ip default-gateway** command, a global command, sets the default IP gateway for the switch.

Now that the configuration has been changed, you should save the configuration so that it will not be lost when the switch is reloaded. The **copy running-config startup-config** command does just that, as seen in the example.

Finally, the **show startup-config** command lists the newly-stored startup configuration. Remember, the previous **show startup-config** command at the end of Example D-4 implied that the startup-config was empty; now, the startup-config has a configuration that will be used upon the next reload of the switch. The **show startup-config** output highlights the configuration commands added earlier in the example.

Boot Sequence and the Configuration Register (Chapter 7)

Chapter 7 covers some details about where a router stores Cisco IOS Software images, how the boot process works, and the role of the configuration register. However, some of the details of the boot sequence were not overtly spelled out; those details are listed next. (If you have not already read Chapter 7, you should do so before reading this section.)

The following list defines the sequence of where a router looks for its IOS, under certain assumptions. The assumptions are as follows:

- The boot field of the configuration register is between hex 2 and F, inclusive.
- No **boot system** commands have been configured.

These two assumptions are true of a newly purchased router that you just took out of the cardboard box. The router then uses the following sequence to find its IOS image to load:

1. Looks in Flash memory, taking the first file in Flash if multiple files exist. (This IOS is a fully functional IOS.)
2. If Flash is empty, the router uses IP broadcasts to find a TFTP server, and request an IOS image, using a Cisco-defined convention for IOS file names. (This process is sometimes called network boot or tftp boot.) (This IOS is also a fully functional IOS.)
3. If that fails, the router loads a limited-function IOS stored in ROM. (This IOS is sometimes called a fallback image, and the user interface of this OS is sometimes called RxBOOT mode.)
4. Although highly unlikely, if the limited-function IOS in ROM fails to load, the router loads an operating system (also stored in ROM), called ROMMON (short for ROM Monitor), which is not an IOS image at all. It can be used to recover and load an IOS image from a TFTP server.

Typically, a router loads the IOS in Flash, with TFTP being used for testing. The last two options in the preceding list are typically not needed for normal operation. However, in some cases, you might want to use the limited-function "fallback" IOS or the ROMMON OS. To do so, you can set the configuration register to a particular value. The last digit of the config register is called the boot field. If the boot field is set to hex 1, and the router is reloaded or powered off/on, the router immediately loads a limited-function IOS stored in ROM (option 3 in the preceding list). If the boot field has a value of hex 0, and the router is reloaded or powered off/on, the router immediately loads the ROMMON OS, and enters ROMMON mode (option 4 in the preceding list).

You can change the configuration register with the config-register IOS command. To change the boot field to 0 so that ROMMON loads next time the router reloads, you enter the following:

```
config-register 0x2100
```

NOTE The running-config file does not have to be copied to startup-config for the configuration register value to be saved—it is saved automatically.

The **show version** command lists the value of the configuration register in the last line of the output of the command.

The ROMMON OS and ROMMON mode play a key role in the password recovery process. For instance, assume you have forgotten all passwords on a router. In that case, you cannot

get into configuration mode to change the configuration register. As it turns out, password recovery requires that you change the configuration register—and the only other way to do that is to get into ROMMON mode. So to get into ROMMON mode and change the configuration register, you can use the following generic process:

Step 1 Turn off the router.

Step 2 Turn on the router.

Step 3 Send a break sequence from the terminal connected to the console—typically a **Ctrl-Break** from the keyboard. (Each terminal emulator might use a different key sequence for sending a break.) This puts the router in ROMMON mode.

Step 4 Set the configuration register to hex 2142. (Methods vary per router platform; the easiest is to use the confreg utility in ROMMON mode.)

Step 5 Continue the normal boot process, typically by using the **i** command, which uses the normal sequence to find the IOS to load into the router.

At this point, the router loads its normal IOS, but it ignores the configuration in NVRAM due to that particular config register setting. (The "4" in 0x2142 means that the "ignore startup configuration" bit has been set.) That means that no passwords are set, so you can log in from the console, and get into enable mode, with no passwords required. However, it also means that the router is using none of its other configuration, so it is not forwarding packets either. To complete the password-reset process and to get the router working again, follow these steps:

Step 1 From the console, get into enable mode.

Step 2 Issue a **copy startup-config running-config** command to load the original configuration.

Step 3 Get into configuration mode using the **configure terminal** command. (Remember, you're already in enable mode, so no other passwords are required.)

Step 4 Change the passwords using the appropriate commands listed in Chapter 7 and in this appendix.

Step 5 Change the **config-register** back to the original value (probably 0x2102), using the configuration-register command, so that the router will not ignore its NVRAM startup-config during the next reload.

Interestingly, if you go to Cisco.com, and search on "password recovery," you will find detailed password-recovery instructions on most every model of Cisco device. Also, if you want to see more details about how to use ROMMON to set the config register, as mentioned in Step 4, refer to the password recovery documents at Cisco.com.

Also note that when using this process, the interfaces should be in a shutdown state, so make sure to perform a **no shutdown** command on all the interfaces.

Subnetting (Chapter 12)

If you read this far, it is fair to assume that you intend to pass the INTRO exam. 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 INTRO exam, you should be able to answer the following style of questions in 60 seconds or less to lessen time pressure on the INTRO exam:

- Given a subnet number and mask, what router command would configure 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, how many subnets can the network be divided?
- Given a mask and a class (A, B, or C) for the network, into 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), what are the valid subnet masks that meet this requirement?

My suggestion of 60 seconds to answer these questions is indeed subjective. If such questions currently require you to take more than one to three minutes of thought and computation, however, I recommend that you spend some time improving the speed with which you find these types of answers. You can use the CD-only appendix that supplies 25 more subnetting practice questions to help you get faster.

For the INTRO exam, you need to be able to answer the following styles of questions within 60–90 seconds, again in my humble and subjective opinion:

- 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.

So, if your goal is to complete your CCNA, even if you are shooting for the INTRO exam to start, you will still benefit from getting really good and fast at the subnetting math. 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 12, "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 is 172.16.1.0, then 172.16.2.0, and so on, with the very last 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. For host 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, one of two formulas can be used, depending on a couple of factors. Choosing which formula to use can be tricky, and in part depends on the context and wording of the exam question. In short, if the question states or implies that classfull logic is needed, the formula should be $2^n - 2$, where n is the number of subnet bits. If the question implies the classless logic is needed, the formula is 2^n. Table D-4 lists the details.

Table D-4 *When to Use Which Formula for the Number of Subnets*

Use this formula…	If the question says…
$2^n - 2$	Classful
$2^n - 2$	RIP version 1 or IGRP
$2^n - 2$	SLSM (in other words, not VLSM)
2^n	Classless
2^n	OSPF, EIGRP, RIP version 2
2^n	VLSM

Cisco makes no public statement about which to use if there are no hints in the question. In my humble opinion, if the question implies nothing at all about which formula to use, use $2^n - 2$. (Table 14-2 in Chapter 14, "Introduction to Dynamic Routing Protocols," introduces the

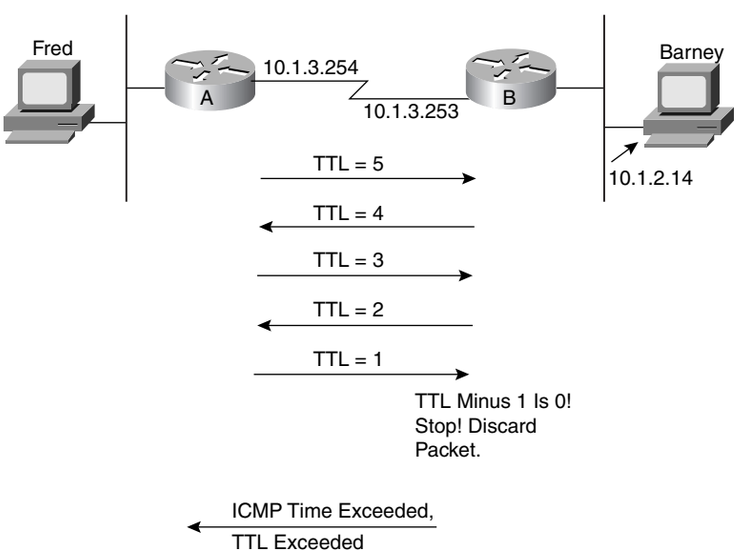
concept of classless and classfull routing protocols. The ICND Exam Certification Guide covers more details about the differences.)

The traceroute Command (Chapter 13)

IOS supports a very useful troubleshooting command called **traceroute**. If you also have a copy of the *CCNA ICND Exam Certification Guide*, turn to Chapter 8, "Advanced TCP/IP Topics," to read about how **traceroute** works. For those who do not have that book, this section of Appendix D repeats the coverage from the ICND book, with one specific point added at the end of this section as a Note.

The ICMP Time Exceeded message notifies a host when a packet that it sent has been discarded because it was "out of time." Packets are not actually timed, but to prevent packets from being forwarded forever when there is a routing loop, each IP header uses a Time to Live (TTL) field. Routers decrement TTL by one every time they forward a packet; if a router decrements TTL to zero, it throws away the packet. This prevents packets from rotating forever. Figure D-1 shows the basic process.

Figure D-1 *TTL Decremented to 0*



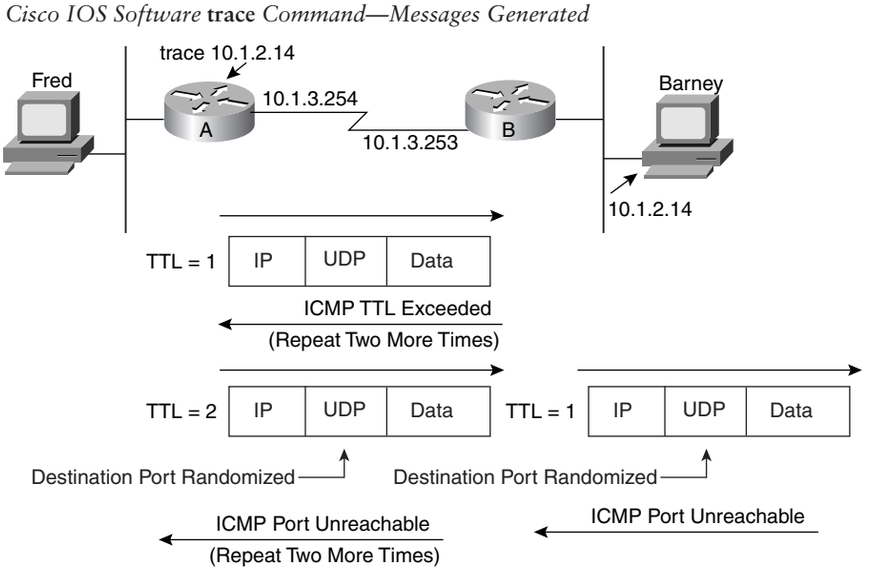
As you see in Figure D-1, the router that discards the packet also sends an ICMP Time Exceeded message, with a Code field of "time exceeded." That way, the sender knows that the packet was not delivered. Getting a Time Exceeded message can also help you when troubleshooting a network. Hopefully, you do not get too many of these; otherwise, you have routing problems.

The IOS **traceroute** command uses the Time Exceeded message and the IP TTL field to its advantage. By purposefully sending IP packets (with a UDP transport layer) with the TTL set to one, an ICMP Time Exceeded message is returned by the first router in the route. That's because that router decrements TTL to zero, causing it to discard the packet, and also sends the Time Exceeded message.

The **traceroute** command actually sends three successive packets with TTL = 1, so if the same router sends back the Time Exceeded message all three times, the **traceroute** command can be confident that router is the first router in the route.

Next, the **traceroute** command sends another set of three IP packets, this time with TTL = 2. These messages make it through the first router but are discarded by the second router because the TTL is decremented to zero. The original packets sent by the **traceroute** command use a destination port number that is very unlikely to be used, so that the destination host will return the Port Unreachable message. The ICMP Port Unreachable message signifies that the packets reached the true destination host without having time exceeded, but there was nothing listening on that port. So the **traceroute** command knows that the packets are getting to the true endpoint. Figure D-2 outlines the process; Router A is using the **traceroute** command trying to find this route to Barney. Example D-5 shows this **traceroute** command on Router A, with debug messages from Router B, showing the resulting Time Exceeded messages.

Figure D-2 *Cisco IOS Software* **trace** *Command—Messages Generated*



Example D-5 *ICMP* **debug** *on Router B When Running* **trace** *Command on Router A*

```
RouterA#traceroute 10.1.2.14

Type escape sequence to abort.
Tracing the route to 10.1.2.14

  1 10.1.3.253 8 msec 4 msec 4 msec
  2 10.1.2.14 12 msec 8 msec 4 msec
RouterA#
!
! Moving to Router B now
!
RouterB#debug ip icmp
RouterB#
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
```

NOTE Many operating systems have some form of IP **traceroute** command. However, the commands differ in some cases with regard to the messages they use. The Cisco IOS **traceroute** command sends IP packets, with UDP headers in them, as described here. Other implementations of the **traceroute** command, such as the Microsoft **tracert** command, use ICMP Echo messages instead.

Miscellaneous Commands (Various)

There are several router commands that are mentioned in new versions of the Cisco INTRO course that are either not currently in the INTRO book or are just briefly mentioned. This section explains those commands more fully.

The show version Command

The **show version** command lists information about the version of IOS in the switch or router—no surprise there. This command also lists information about the amount of RAM (64 MB in this case), Flash (16 MB in this case), and the value of the configuration register (0x2102 in this case). Example D-6 shows an example, with those portions highlighted.

Example D-6 *The* **show version** *Command: Details*

```
Switch>enable
Router4#show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-JK9O3S-M), Version 12.2(16a), RELEASE SOFTWARE (fc2)
Copyright  1986-2003 by cisco Systems, Inc.
Compiled Fri 18-Apr-03 19:25 by pwade
```

continues

Example D-6 *The* **show version** *Command: Details (Continued)*

```
Image text-base: 0x8000808C, data-base: 0x815F0848

ROM: System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)

Router3 uptime is 0 minutes
System returned to ROM by power-on
System image file is "flash:c2600-jk9o3s-mz.122-16a.bin"

cisco 2610 (MPC860) processor (revision 0x203) with 59392K/6144K bytes of memory
.
Processor board ID JAB033902ZB (3349316226)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

History Buffer Commands

When you type commands from the command-line interface (CLI), the last several commands are saved in the history buffer. As mentioned back in Chapter 7, you can use the up-arrow key, or **Ctrl-P**, to move back in the history buffer stack to retrieve a command you typed a few commands ago. This feature makes it very easy and fast to use a set of commands repeatedly. Table D-5 lists some of the key commands related to the history buffer.

Table D-5 *Commands Related to the History Buffer*

Command	Meaning
show history	Lists the commands currently held in the history buffer.
history size *x*	From global configuration mode, sets the default number of commands saved in the history buffer for every one in the router or switch.
terminal history size *x*	From EXEC mode, this command allows a single user to set, just for this one connection, the size of their history buffer.

The logging synchronous and exec-timeout Commands

The console automatically receives copies of all unsolicited messages on a router; that feature cannot be disabled. Normally the router puts the message on the screen at any time—including right in the middle of a command you are typing, or in the middle of the output of

a **show** command. However, with the **logging synchronous** console subcommand, the router waits until the **show** command output is complete before displaying the log messages.

You can also define an inactivity timeout on the console using the **exec-timeout** *hours minutes* command. After that much time passes with no typing at the console, the console session is closed. Also, by setting the timeout to 0 hours and 0 minutes, the router never times out the console connection. Example D-7 shows the syntax for these two commands.

Example D-7 *Defining Console Inactivity Timeouts and When to Display Log Messages*

```
line console 0
 login
 password cisco
 exec-timeout 0 0
 logging synchronous
```

Additional Definitions (Various)

This section contains definitions of terms added to the book after initial publication in June 2003.

I/G bit I/G stands for Individual/Group. This bit is the most significant bit in the most significant byte of an Ethernet MAC address; its value implies that the address is a unicast MAC address (binary 0) or not (binary 1).

U/L bit U/L stands for Universal/Local. This bit is the second most significant bit in the most significant byte of an Ethernet MAC address; a value of binary 0 implies that the address is a Universally Administered Address (UAA) (also known as Burned-In Address [BIA]); with binary 1, implies that the MAC address is a locally configured address.

Network Access Layer Another term for the TCP/IP network interface layer. This layer is the lowest in the TCP/IP architectural model, and most closely equates to OSI Layers 1 and 2.

Microsegmentation A switch creates a separate collision domain per interface. The device(s) off each switch port can be considered to be on the same Ethernet segment. Microsegmentation is a term used to compare hubs, which create a single collision domain and segment, to switches, which create many collision domains and segments. The term micro simply refers to the fact that each segment is small, often with a single attached device.

ICMP Source Quench A seldom-used ICMP message type that is meant to tell the sender of a packet to slow down the rate at which the sender is sending data. The idea is for routers to react to congestion in the network by getting the sender, or source, of the packets causing the congestion to slow down, or quench, the rate at which they send packets.