

- RPVST+ adds an extra type-length value (TLV) to the BPDU that identifies the VLAN ID, while RSTP does not (because it does not need to, as RSTP ignores VLANs.)
- Both view the 16-bit priority as having a 12-bit System ID Extension, with RSTP setting the value to 0000.0000.0000, meaning “no VLAN,” while RPVST+ uses the VLAN ID.

In other words, standard RSTP behaves as if VLANs do not exist, while Cisco’s RPVST+ integrates VLAN information into the entire process.

**NOTE** Some documents refer to the feature of sending BPDUs over trunks with VLAN tags matching the same VLAN as BPDU tunneling.

## Other RSTP Configuration Options

This chapter does not attempt to work through all the configuration options available for RSTP. However, many of the configuration settings may be intuitive now that you know quite a bit about the protocol. This final topic in the first section of the chapter summarizes a few of the configuration concepts. As a reminder, for those interested in continuing on to CCNP Enterprise, you might be interested in reading more about RSTP configuration in the companion website’s Appendix O, “Spanning Tree Protocol Implementation.”

- **Switch Priority:** The global command **spanning-tree vlan *x* priority *y*** lets an engineer set the switch’s priority in that VLAN.
- **Primary and Secondary Root Switches:** The global command **spanning-tree vlan *x* root primary | secondary** also lets you set the priority, but the switch decides on a value to make that switch likely to be the primary root switch (the root) or the secondary root switch (the switch that becomes root if the primary fails).
- **Port Costs:** The interface subcommand **spanning-tree [vlan *x*] cost *y*** lets an engineer set the switch’s STP/RSTP cost on that port, either for all VLANs or for a specific VLAN on that port. Changing those costs then changes the root cost for some switches, which impacts the choice of root ports and designated ports.

That concludes this chapter’s examination of RSTP configuration—now on to Layer 2 EtherChannel!

## Configuring Layer 2 EtherChannel

As introduced in Chapter 9, two neighboring switches can treat multiple parallel links between each other as a single logical link called an *EtherChannel*. Without EtherChannel, a switch treats each physical port as an independent port, applying MAC learning, forwarding, and STP logic per physical port. With EtherChannel, the switch applies all those same processes to a group of physical ports as one entity: the EtherChannel. Without EtherChannel, with parallel links between two switches, STP/RSTP would block all links except one, but with EtherChannel, the switch can use all the links, load balancing the traffic over the links.

**NOTE** All references to EtherChannel in this chapter refer to Layer 2 EtherChannels, not to Layer 3 EtherChannels (as discussed in Chapter 17, “IP Routing in the LAN”). CCNA 200-301 exam topics include both Layer 2 and Layer 3 EtherChannels.

EtherChannel may be one of the most challenging switch features to make work. First, the configuration has several options, so you have to remember the details of which options work together. Second, the switches also require a variety of other interface settings to match among all the links in the channel, so you have to know those settings as well.

This section shows how to configure a Layer 2 EtherChannel, first through manual (static) configuration, and then by allowing dynamic protocols to create the channel. This section closes with some information about some common configuration issues that occur with Layer 2 EtherChannels.

## Configuring a Manual Layer 2 EtherChannel

To configure a Layer 2 EtherChannel so that all the ports always attempt to be part of the channel, simply add the correct **channel-group** configuration command to each physical interface, on each switch, all with the **on** keyword, and all with the same number. The **on** keyword tells the switches to place a physical interface into an EtherChannel, and the number identifies the PortChannel interface number that the interface should be a part of.

Before getting into the configuration and verification, however, you need to start using three terms as synonyms: *EtherChannel*, *PortChannel*, and *Channel-group*. Oddly, IOS uses the **channel-group** configuration command, but then to display its status, IOS uses the **show etherchannel** command. Then the output of this **show** command refers to neither an “EtherChannel” nor a “Channel-group,” instead using the term “PortChannel.” So, pay close attention to these three terms in the example.

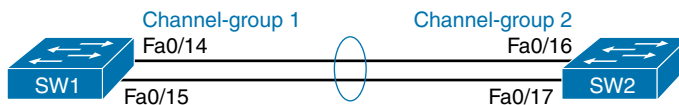
To configure an EtherChannel manually, follow these steps:

**Key  
Topic**

**Config  
Checklist**

- Step 1.** Add the **channel-group number mode on** command in interface configuration mode under each physical interface that should be in the channel to add it to the channel.
- Step 2.** Use the same number for all commands on the same switch, but the channel-group number on the neighboring switch can differ.

Example 10-4 shows a simple example, with two links between switches SW1 and SW2, as shown in Figure 10-6. The configuration shows SW1's two interfaces placed into channel-group 1, with two **show** commands to follow.



**Figure 10-6** Sample LAN Used in EtherChannel Example

### Example 10-4 Configuring and Monitoring EtherChannel

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface fa 0/14
SW1(config-if)# channel-group 1 mode on
SW1(config)# interface fa 0/15
```

```
SW1(config-if)# channel-group 1 mode on
SW1(config-if)# ^Z

SW1# show spanning-tree vlan 3

VLAN0003
Spanning tree enabled protocol ieee
Root ID    Priority    28675
           Address    0019.e859.5380
           Cost       12
           Port       72 (Port-channel1)
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority    28675 (priority 28672 sys-id-ext 3)
           Address    0019.e86a.6f80
           Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time  300

Interface                Role Sts Cost          Prio.Nbr Type
-----
Po1                        Root FWD 12           128.64   P2p Peer(STP)

SW1# show etherchannel 1 summary

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby  (LACP only)
        R - Layer3       S - Layer2
        U - in use       N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators: 1

Group  Port-channel  Protocol    Ports
-----+-----+-----+
1      Po1 (SU)      -           Fa0/14 (P) Fa0/15 (P)
```

Take a few moments to look at the output in the two **show** commands in the example, as well. First, the **show spanning-tree** command lists Po1, short for PortChannel1, as

an interface. This interface exists because of the **channel-group** commands using the **1** parameter. STP no longer operates on physical interfaces Fa0/14 and Fa0/15, instead operating on the PortChannel1 interface, so only that interface is listed in the output.

Next, note the output of the **show etherchannel 1 summary** command. It lists as a heading “Port-channel,” with Po1 below it. It also lists both Fa0/14 and Fa0/15 in the list of ports, with a (P) beside each. Per the legend, the *P* means that the ports are bundled in the port channel, which is a code that means these ports have passed all the configuration checks and are valid to be included in the channel.

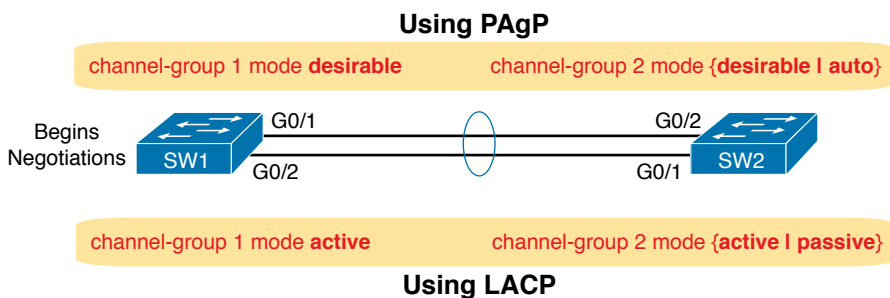
## Configuring Dynamic EtherChannels

In addition to manual configuration, Cisco switches also support two different configuration options that then use a dynamic protocol to negotiate whether a particular link becomes part of an EtherChannel or not. Basically, the configuration enables a protocol for a particular channel-group number. At that point, the switch can use the protocol to send messages to/from the neighboring switch and discover whether their configuration settings pass all checks. If a given physical link passes, the link is added to the EtherChannel and used; if not, it is placed in a down state, and not used, until the configuration inconsistency can be resolved.

Most Cisco Catalyst switches support the Cisco-proprietary Port Aggregation Protocol (PAgP) and the IEEE standard Link Aggregation Control Protocol (LACP), based on IEEE standard 802.3ad. Although differences exist between the two, to the depth discussed here, they both accomplish the same task: negotiate so that only links that pass the configuration checks are actually used in an EtherChannel.

One difference of note is that LACP does support more links in a channel—16—as compared to PAgP’s maximum of 8. With LACP, only 8 can be active at one time, with the others waiting to be used should any of the other links fail.

To configure either protocol, a switch uses the **channel-group** configuration commands on each switch, but with a keyword that either means “use this protocol and begin negotiations” or “use this protocol and wait for the other switch to begin negotiations.” As shown in Figure 10-7, the **desirable** and **auto** keywords enable PAgP, and the **active** and **passive** keywords enable LACP. With these options, at least one side has to begin the negotiations. In other words, with PAgP, at least one of the two sides must use **desirable**, and with LACP, at least one of the two sides must use **active**.



**Figure 10-7** Correct EtherChannel Configuration Combinations

**NOTE** Do not use the **on** parameter on one end, and either **auto** or **desirable** (or for LACP, **active** or **passive**) on the neighboring switch. The **on** option uses neither PAgP nor LACP, so a configuration that uses **on**, with PAgP or LACP options on the other end, would prevent the EtherChannel from working.

For example, in the design shown in Figure 10-7, imagine both physical interfaces on both switches were configured with the **channel-group 2 mode desirable** interface subcommand. As a result, the two switches would negotiate and create an EtherChannel. Example 10-5 shows the verification of that configuration, with the command **show etherchannel 1 port-channel**. This command confirms the protocol in use (PAgP, because the **desirable** keyword was configured), and the list of interfaces in the channel.

#### Example 10-5 EtherChannel Verification: PAgP Desirable Mode

```
SW1# show etherchannel 1 port-channel
      Port-channels in the group:
      -----

Port-channel: Po1
-----
Age of the Port-channel   = 0d:00h:04m:04s
Logical slot/port        = 16/1           Number of ports = 2
GC                       = 0x00020001     HotStandBy port = null
Port state               = Port-channel Ag-Inuse
Protocol                 = PAgP
Port security            = Disabled
Load share deferral      = Disabled

Ports in the Port-channel:

Index   Load   Port          EC state          No of bits
-----+-----+-----+-----+-----
  0      00     Gi0/1        Desirable-S1       0
  0      00     Gi0/2        Desirable-S1       0

Time since last port bundled: 0d:00h:03m:57s Gi0/2
```

## Physical Interface Configuration and EtherChannels

Even when the **channel-group** commands have all been configured correctly, other configuration settings can prevent a switch from using a physical port in an EtherChannel—even physical ports manually configured to be part of the channel. The next topic examines those reasons.

First, before using a physical port in an EtherChannel, the switch compares the new physical port's configuration to the existing ports in the channel. That new physical interface's settings must be the same as the existing ports' settings; otherwise, the switch does not add the new link to the list of approved and working interfaces in the channel. That is, the

physical interface remains configured as part of the PortChannel, but it is not used as part of the channel, often being placed into some nonworking state.

The list of items the switch checks includes the following:



- Speed
- Duplex
- Operational access or trunking state (all must be access, or all must be trunks)
- If an access port, the access VLAN
- If a trunk port, the allowed VLAN list (per the **switchport trunk allowed** command)
- If a trunk port, the native VLAN
- STP interface settings

In addition, switches check the settings on the neighboring switch. To do so, the switches either use PAgP or LACP (if already in use) or use Cisco Discovery Protocol (CDP) if using manual configuration. When checking neighbors, all settings except the STP settings must match.

As an example, SW1 and SW2 again use two links in one EtherChannel from Figure 10-7. Before configuring the EtherChannel, SW1's G0/2 was given a different RSTP port cost than G0/1. Example 10-6 picks up the story just after configuring the correct **channel-group** commands, when the switch is deciding whether to use G0/1 and G0/2 in this.

#### Example 10-6 Local Interfaces Fail in EtherChannel Because of Mismatched STP Cost

```
*Mar 1 23:18:56.132: %PM-4-ERR_DISABLE: channel-misconfig (STP) error detected on
Po1, putting Gi0/1 in err-disable state
*Mar 1 23:18:56.132: %PM-4-ERR_DISABLE: channel-misconfig (STP) error detected on
Po1, putting Gi0/2 in err-disable state
*Mar 1 23:18:56.132: %PM-4-ERR_DISABLE: channel-misconfig (STP) error detected on Po1,
putting Po1 in err-disable state
*Mar 1 23:18:58.120: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to
down
*Mar 1 23:18:58.137: %LINK-3-UPDOWN: Interface Port-channel1, changed state to down
*Mar 1 23:18:58.137: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to
down

SW1# show etherchannel summary

Flags: D - down          P - bundled in port-channel
       I - stand-alone s - suspended
       H - Hot-standby (LACP only)
       R - Layer3        S - Layer2
       U - in use        N - not in use, no aggregation
       f - failed to allocate aggregator

       M - not in use, minimum links not met
       m - not in use, port not aggregated due to minimum links not met
       u - unsuitable for bundling
       w - waiting to be aggregated
```

```
d - default port

A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators: 1

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
1      Po1 (SD)        -         Gi0/1 (D) Gi0/2 (D)
```

The messages at the top of the example specifically state what the switch does when determining whether the interface settings match. In this case, SW1 detects the different STP costs. SW1 does not use G0/1, does not use G0/2, and even places them into an err-disabled state. The switch also puts the PortChannel into err-disabled state. As a result, the PortChannel is not operational, and the physical interfaces are also not operational.

To solve this problem, you must reconfigure the physical interfaces to use the same STP settings. In addition, the PortChannel and physical interfaces must be **shutdown**, and then **no shutdown**, to recover from the err-disabled state. (Note that when a switch applies the **shutdown** and **no shutdown** commands to a PortChannel, it applies those same commands to the physical interfaces, as well; so, just do the **shutdown/no shutdown** on the PortChannel interface.)

EtherChannel Load Distribution

When using Layer 2 EtherChannels, a switch’s MAC learning process associates MAC addresses with the PortChannel interfaces and not the underlying physical ports. Later, when a switch makes a forwarding decision to send a frame out a PortChannel interface, the switch must do more work: to decide out which specific physical port to use to forward the frame. IOS documentation refers to those rules as *EtherChannel load distribution* or *load balancing*. Figure 10-8 shows the main idea.

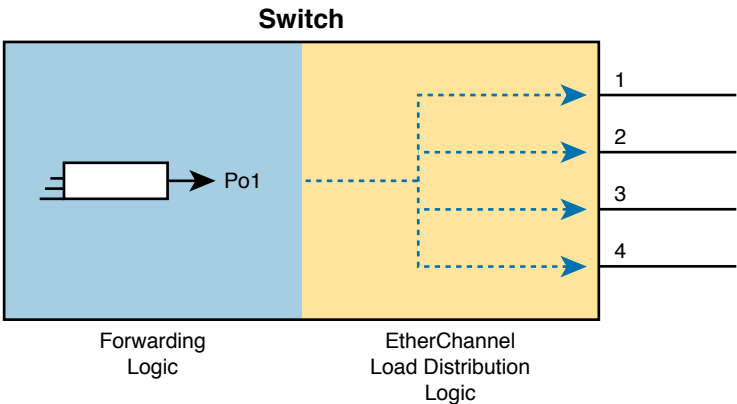


Figure 10-8 Correct EtherChannel Configuration Combinations

Configuration Options for EtherChannel Load Distribution

EtherChannel load distribution makes the choice for each frame based on various numeric values found in the Layer 2, 3, and 4 headers. The process uses one configurable setting as input: the load distribution method as defined with the **port-channel load-balance *method*** global command. The process then performs some match against the fields identified by the configured method.

Table 10-4 lists the most common methods. However, note that some switches may support only MAC-based methods, or only MAC- and IP-based methods, depending on the model and software version.

Table 10-4 EtherChannel Load Distribution Methods

Configuration Keyword	Math Uses...	Layer
src-mac	Source MAC address	2
dst-mac	Destination MAC address	2
src-dst-mac	Both source and destination MAC	2
src-ip	Source IP address	3
dst-ip	Destination IP address	3
src-dst-ip	Both source and destination IP	3
src-port	Source TCP or UDP port	4
dst-port	Destination TCP or UDP port	4
src-dst-port	Both source and destination TCP or UDP port	4

To appreciate why you might want to use different methods, you need to consider the results of how switches make their choice. (The discussion here focuses on the result, and not the logic, because the logic remains internal to the switch, and Cisco does not document how each switch model or IOS version works internally.) However, the various load distribution algorithms do share some common goals:

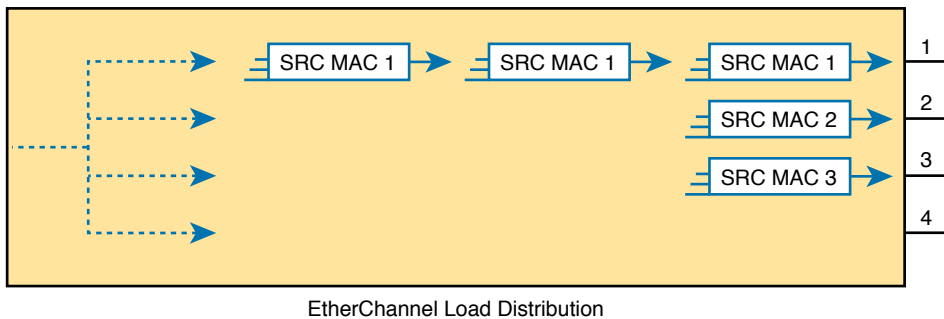
- To cause all messages in a single application flow to use the same link in the channel, rather than being sent over different links. Doing so means that the switch will not inadvertently reorder the messages sent in that application flow by sending one message over a busy link that has a queue of waiting messages, while immediately sending the next message out an unused link.
- To integrate the load distribution algorithm work into the hardware forwarding ASIC so that load distribution works just as quickly as the work to forward any other frame.
- To use all the active links in the EtherChannel, adjusting to the addition and removal of active links over time.
- Within the constraints of the other goals, balance the traffic across those active links.

In short, the algorithms first intend to avoid message reordering, make use of the switch forwarding ASICs, and use all the active links. However, the algorithm does not attempt to send the exact same number of bits over each link over time. The algorithm does try to balance the traffic, but always within the constraints of the other goals.



Whatever load distribution method you choose, the method identifies fields in the message headers. Any messages in the same application flow will then have the same values in the fields used by the load distribution algorithm and will always be forwarded over the same link. For example, when a user connects to a website, that web server may return thousands of packets to the client. Those thousands of packets should flow over the same link in the EtherChannel.

For instance, with the load distribution method of **src-mac** (meaning source MAC address), all frames with the same MAC address flow over one link. Figure 10-9 shows the idea with pseudo MAC addresses, with the load distribution sending frames with source MAC 1 over link 1, source MAC 2 over link 2, and source MAC 3 over link 3.



**Figure 10-9** *Distributing All Frames with Same Mac Out Same Interface*

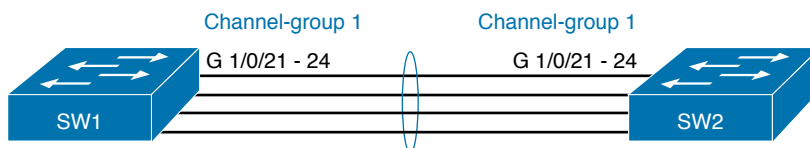
Cisco provides a variety of load distribution options so that the engineer can examine the flows in the network with the idea of finding which fields have the most variety in their values: source and destination MAC, or IP address, or transport layer port numbers. The more variety in the values in the fields, the better the balancing effects, and the lower the chance of sending disproportionate amounts of traffic over one link.

**NOTE** The algorithm focuses on the low-order bits in the fields in the headers because the low-order bits typically differ the most in real networks, while the high-order bits do not differ much. By focusing on the lower-order bits, the algorithm achieves better balancing of traffic over the links.

10

### The Effects of the EtherChannel Load Distribution Algorithm

Figure 10-10 details a new EtherChannel that will be used in two examples to show the effects of load distribution. The examples will focus on frames sent by switch SW1 in the figure, showing the use of the **test etherchannel load-balance EXEC** command. That command asks the switch to consider some addresses or ports and answer the question: which link would you use when forwarding a message with those address/port values?



**Figure 10-10** *Four-Link EtherChannel*

Example 10-7 shows how switch SW1 distributes traffic when using **src-mac** load distribution. The example lists the output from three of the **test etherchannel load-balance** commands, but note that all three commands use the same source MAC address. As a result, the answer from each command references the same interface (G1/0/22 in this case).

**Example 10-7** *Testing with Identical Source MACs When Using src-mac Balancing*

```
SW1# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
IPv4: Source MAC address
IPv6: Source MAC address

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1111
Would select Gi1/0/22 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1112
Would select Gi1/0/22 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1113
Would select Gi1/0/22 of Po1
```

Example 10-7 makes two important points:

- All three tests list the same outgoing physical interface because (1) the method uses only the source MAC address, and (2) all three tests use the same MAC addresses.
- All three tests use a different destination MAC address, with different low-order bits, but that had no impact on the choice because the method—**src-mac**—does not consider the destination MAC address.

In contrast on that first point, Example 10-8 repeats the **test** commands from Example 10-7. The switch still uses the **src-mac** balancing method, but now with different source MAC addresses in each test. Notice that the source MAC addresses used in the tests differ by just a few bit values in the low-order bits, so as a result, each test shows a different interface choice by SW1.

**Example 10-8** *Testing with Source MACs with Low-Order Bit Differences*

```
SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1111
Would select Gi1/0/22 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0002 0200.1111.1111
Would select Gi1/0/24 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0003 0200.1111.1111
Would select Gi1/0/23 of Po1
```

Example 10-9 shows yet a third variation, this time changing the load distribution method to **src-dst-mac**, which means that the switch will consider both source and destination MAC. The example repeats the exact same **test etherchannel** commands as Example 10-7, with the exact same MAC addresses: the source MAC addresses remain the same in all three tests, but the destination MAC addresses differ in the low-order bits. With the chosen destination MAC values differing slightly, switch SW1 happens to choose three different interfaces.

### Example 10-9 Evidence of Source and Destination MAC Load Distribution

```
SW1# config t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# port-channel load-balance src-dst-mac
SW1(config)# ^Z
SW1#
SW1# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-dst-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
IPv4: Source XOR Destination MAC address
IPv6: Source XOR Destination MAC address

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1111
Would select Gi1/0/22 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1112
Would select Gi1/0/24 of Po1

SW1# test etherchannel load-balance interface po1 mac 0200.0000.0001 0200.1111.1113
Would select Gi1/0/23 of Po1
```

## Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 10-5 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

**Table 10-5** Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review config checklists		Book, website
Review command tables		Book
Review memory tables		Website
Do labs		Blog

## Review All the Key Topics

**Table 10-6** Key Topics for Chapter 10

Key Topic Element	Description	Page Number
Figure 10-1	Typical design choice for which switches should be made to be root	241
Figure 10-2	Conceptual view of load-balancing benefits of PVST+	242
Table 10-2	STP Standards and Configuration Options	243
Figure 10-4	Shows the format of the system ID extension of the STP priority field	243
List	Facts about RPVST+'s methods versus RSTP	246
List	Steps to manually configure an EtherChannel	248
List	Items a switch compares in a new physical port's configuration to the existing ports in the channel	252

## Key Terms You Should Know

Rapid PVST+, PVST+, system ID extension, PAgP, LACP, PortChannel, Channel-group, EtherChannel, EtherChannel Load Distribution, primary root, secondary root

## Command References

Tables 10-7 and 10-8 list configuration and verification commands used in this chapter. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.

**Table 10-7** Chapter 10 Configuration Command Reference

Command	Description
<b>spanning-tree mode</b> {pvst   rapid-pvst   mst}	Global configuration command to set the STP mode.
<b>spanning-tree</b> [vlan <i>vlan-number</i> ] <b>root primary</b>	Global configuration command that changes this switch to the root switch. The switch's priority is changed to the lower of either 24,576 or 4096 less than the priority of the current root bridge when the command was issued.
<b>spanning-tree</b> [vlan <i>vlan-number</i> ] <b>root secondary</b>	Global configuration command that sets this switch's STP base priority to 28,672.
<b>spanning-tree</b> vlan <i>vlan-id</i> <b>priority</b> <i>priority</i>	Global configuration command that changes the bridge priority of this switch for the specified VLAN.
<b>spanning-tree</b> [vlan <i>vlan-number</i> ] <b>cost</b> <i>cost</i>	Interface subcommand that changes the STP cost to the configured value.
<b>spanning-tree</b> [vlan <i>vlan-number</i> ] <b>port-priority</b> <i>priority</i>	Interface subcommand that changes the STP port priority in that VLAN (0 to 240, in increments of 16).
<b>channel-group</b> <i>channel-group-number</i> <b>mode</b> {auto   desirable   active   passive   on}	Interface subcommand that enables EtherChannel on the interface.

**Table 10-8** Chapter 10 EXEC Command Reference

Command	Description
<b>show spanning-tree</b>	Lists details about the state of STP on the switch, including the state of each port.
<b>show spanning-tree</b> vlan <i>vlan-id</i>	Lists STP information for the specified VLAN.
<b>show etherchannel</b> [ <i>channel-group-number</i> ] {brief   detail   port   port-channel   summary}	Lists information about the state of EtherChannels on this switch.



# Part III Review

Keep track of your part review progress with the checklist shown in Table P3-1. Details on each task follow the table.

**Table P3-1** Part III Part Review Checklist

Activity	1st Date Completed	2nd Date Completed
Repeat All DIKTA Questions		
Answer Part Review Questions		
Review Key Topics		
Do Labs		
Review Appendices		
Videos		

## Repeat All DIKTA Questions

For this task, answer the “Do I Know This Already?” questions again for the chapters in this part of the book, using the PCPT software.

## Answer Part Review Questions

For this task, answer the Part Review questions for this part of the book, using the PTP software.

## Review Key Topics

Review all key topics in all chapters in this part, either by browsing the chapters or by using the Key Topics application on the companion website.

## Labs

Depending on your chosen lab tool, here are some suggestions for what to do in lab:

**Pearson Network Simulator:** If you use the full Pearson ICND1 or CCNA simulator, focus more on the configuration scenario and troubleshooting scenario labs associated with the topics in this part of the book. These types of labs include a larger set of topics and work well as Part Review activities. (See the Introduction for some details about how to find which labs are about topics in this part of the book.) Note that the Sim Lite that comes with this book also has a couple of labs about VLANs.

**Blog: Config Labs:** The author’s blog includes a series of configuration-focused labs that you can do on paper, each in 10–15 minutes. Review and perform the labs for this part of the book, as found at <http://blog.certskills.com>. Then navigate to the Hands-on Config labs.

**Other:** If using other lab tools, as a few suggestions: Make sure and experiment heavily with VLAN configuration and VLAN trunking configuration.

## Dig Deeper with Appendices on the Companion Website

The chapters in Part III of the book recommended the following appendices for extra reading. If you care to read further, consider:

- **Appendix K, “Analyzing Ethernet LAN Designs”:** A chapter from the previous edition that discusses design topologies and LAN design with two-tier and three-tier designs, including access and distribution switches.
- **Appendix O, “Spanning Tree Protocol Implementation”:** A chapter that works through the configuration and verification commands for STP and RSTP.
- **Appendix P, “LAN Troubleshooting”:** A chapter from the previous edition of the ICND2 Cert Guide. This chapter includes topics about VLANs, trunks, and STP and how to troubleshoot each.

## Watch Videos

Chapter 8 recommends two videos, one about VLANs and another about the VLAN allowed list on trunks. If you have not watched those videos yet, take a moment to scan back to Chapter 8 on the companion website and watch the videos.



The book makes a big transition at this point. Part I gave you a broad introduction to networking, and Parts II and III went into some detail about the dominant LAN technology today: Ethernet. Part IV transitions from Ethernet to the network layer details that sit above Ethernet and WAN technology, specifically IP Version 4 (IPv4).

Thinking about the network layer requires engineers to shift how they think about addressing. Ethernet allows the luxury of using universal MAC addresses, assigned by the manufacturers, with no need to plan or configure addresses. Although the network engineer needs to understand MAC addresses, MAC already exists on each Ethernet NIC, and switches learn the Ethernet MAC addresses dynamically without even needing to be configured to do so. As a result, most people operating the network can ignore the specific MAC address values for most tasks.

Conversely, IP addressing gives us flexibility and allows choice, but those features require planning, along with a much deeper understanding of the internal structure of the addresses. People operating the network must be more aware of the network layer addresses when doing many tasks. To better prepare you for these Layer 3 addressing details, this part breaks down the addressing details into four chapters, with an opportunity to learn more in preparation for the CCNP Enterprise certification.

Part IV examines most of the basic details of IPv4 addressing and subnetting, mostly from the perspective of operating an IP network. Chapter 11 takes a grand tour of IPv4 addressing as implemented inside a typical enterprise network. Chapters 12, 13, and 14 look at some of the specific questions people must ask themselves when operating an IPv4 network.



# Part IV

## IPv4 Addressing

**Chapter 11:** Perspectives on IPv4 Subnetting

**Chapter 12:** Analyzing Classful IPv4 Networks

**Chapter 13:** Analyzing Subnet Masks

**Chapter 14:** Analyzing Existing Subnets

**Part IV Review**

# Perspectives on IPv4 Subnetting

This chapter covers the following exam topics:

### 1.0 Network Fundamentals

1.6 Configure and verify IPv4 addressing and subnetting

1.7 Describe the need for private IPv4 addressing

Most entry-level networking jobs require you to operate and troubleshoot a network using a preexisting IP addressing and subnetting plan. The CCNA exam assesses your readiness to use preexisting IP addressing and subnetting information to perform typical operations tasks, such as monitoring the network, reacting to possible problems, configuring addresses for new parts of the network, and troubleshooting those problems.

However, you also need to understand how networks are designed and why. Anyone monitoring a network must continually ask the question, “Is the network working *as designed*?” If a problem exists, you must consider questions such as “What happens when the network works normally, and what is different right now?” Both questions require you to understand the intended design of the network, including details of the IP addressing and subnetting design.

This chapter provides some perspectives and answers for the bigger issues in IPv4 addressing. What addresses can be used so that they work properly? What addresses should be used? When told to use certain numbers, what does that tell you about the choices made by some other network engineer? How do these choices impact the practical job of configuring switches, routers, hosts, and operating the network on a daily basis? This chapter hopes to answer these questions while revealing details of how IPv4 addresses work.

## “Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

**Table 11-1** “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Analyze Requirements	1–3
Make Design Choices	4–7

1. Host A is a PC, connected to switch SW1 and assigned to VLAN 1. Which of the following are typically assigned an IP address in the same subnet as host A? (Choose two answers.)
  - a. The local router's WAN interface
  - b. The local router's LAN interface
  - c. All other hosts attached to the same switch
  - d. Other hosts attached to the same switch and also in VLAN 1
2. Why does the formula for the number of hosts per subnet ( $2^H - 2$ ) require the subtraction of two hosts?
  - a. To reserve two addresses for redundant default gateways (routers)
  - b. To reserve the two addresses required for DHCP operation
  - c. To reserve addresses for the subnet ID and default gateway (router)
  - d. To reserve addresses for the subnet broadcast address and subnet ID
3. A Class B network needs to be subnetted such that it supports 100 subnets and 100 hosts/subnet. Which of the following answers list a workable combination for the number of network, subnet, and host bits? (Choose two answers.)
  - a. Network = 16, subnet = 7, host = 7
  - b. Network = 16, subnet = 8, host = 8
  - c. Network = 16, subnet = 9, host = 7
  - d. Network = 8, subnet = 7, host = 17
4. Which of the following are private IP networks? (Choose two answers.)
  - a. 172.31.0.0
  - b. 172.32.0.0
  - c. 192.168.255.0
  - d. 192.1.168.0
  - e. 11.0.0.0
5. Which of the following are public IP networks? (Choose three answers.)
  - a. 9.0.0.0
  - b. 172.30.0.0
  - c. 192.168.255.0
  - d. 192.1.168.0
  - e. 1.0.0.0

6. Before Class B network 172.16.0.0 is subnetted by a network engineer, what parts of the structure of the IP addresses in this network already exist, with a specific size? (Choose two answers.)
- a. Network
  - b. Subnet
  - c. Host
  - d. Broadcast
7. A network engineer spends time thinking about the entire Class B network 172.16.0.0 and how to subnet that network. He then chooses how to subnet this Class B network and creates an addressing and subnetting plan, on paper, showing his choices. If you compare his thoughts about this network before subnetting the network to his thoughts about this network after mentally subnetting the network, which of the following occurred to the parts of the structure of addresses in this network?
- a. The subnet part got smaller.
  - b. The host part got smaller.
  - c. The network part got smaller.
  - d. The host part was removed.
  - e. The network part was removed.

## Foundation Topics

### Introduction to Subnetting

Say you just happened to be at the sandwich shop when it was selling the world's longest sandwich. You're pretty hungry, so you go for it. Now you have one sandwich, but because it's over 2 kilometers long, you realize it's a bit more than you need for lunch all by yourself. To make the sandwich more useful (and more portable), you chop the sandwich into meal-size pieces and give the pieces to other folks around you who are also ready for lunch.

Huh? Well, subnetting, at least the main concept, is similar to this sandwich story. You start with one network, but it is just one large network. As a single large entity, it might not be useful, and it is probably far too large. To make it useful, you chop it into smaller pieces, called subnets, and assign those subnets to be used in different parts of the enterprise internetwork.

This short first section of the chapter introduces IP subnetting. First, it shows the general ideas behind a completed subnet design that indeed chops (or subnets) one network into subnets. The rest of this section describes the many design steps that you would take to create just such a subnet design. By the end of this section, you should have the right context to then read through the subnetting design steps introduced throughout the rest of this chapter.

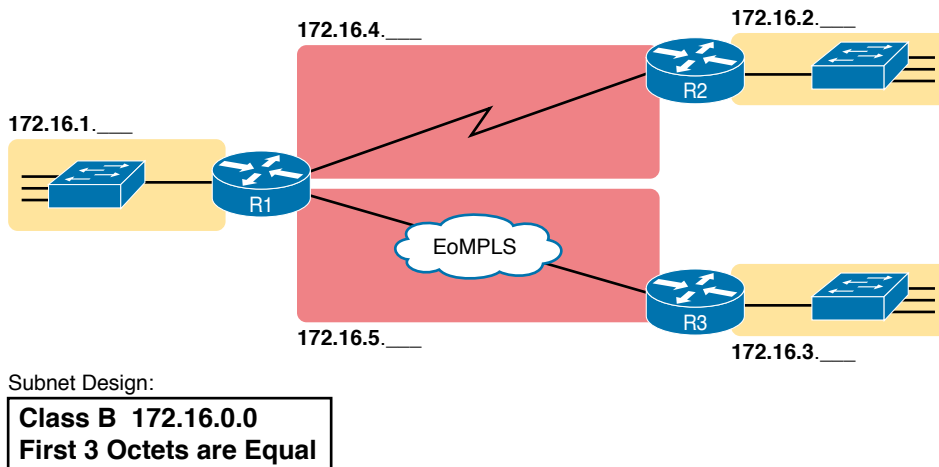
**NOTE** All the chapters from this chapter up until Chapter 22, "Fundamentals of IP Version 6," focus on IPv4 rather than IPv6. All references to *IP* refer to IPv4 unless otherwise stated.

## Subnetting Defined Through a Simple Example

An IP network—in other words, a Class A, B, or C network—is simply a set of consecutively numbered IP addresses that follows some preset rules. These Class A, B, and C rules define that for a given network, all the addresses in the network have the same value in some of the octets of the addresses. For example, Class B network 172.16.0.0 consists of all IP addresses that begin with 172.16: 172.16.0.0, 172.16.0.1, 172.16.0.2, and so on, through 172.16.255.255. Another example: Class A network 10.0.0.0 includes all addresses that begin with 10.

An IP subnet is simply a subset of a Class A, B, or C network. In fact, the word *subnet* is a shortened version of the phrase *subdivided network*. For example, one subnet of Class B network 172.16.0.0 could be the set of all IP addresses that begin with 172.16.1, and would include 172.16.1.0, 172.16.1.1, 172.16.1.2, and so on, up through 172.16.1.255. Another subnet of that same Class B network could be all addresses that begin with 172.16.2.

To give you a general idea, Figure 11-1 shows some basic documentation from a completed subnet design that could be used when an engineer subnets Class B network 172.16.0.0.



**Figure 11-1** Subnet Plan Document

The design shows five subnets—one for each of the three LANs and one each for the two WAN links. The small text note shows the rationale used by the engineer for the subnets: each subnet includes addresses that have the same value in the first three octets. For example, for the LAN on the left, the number shows 172.16.1., meaning “all addresses that begin with 172.16.1.” Also, note that the design, as shown, does not use all the addresses in Class B network 172.16.0.0, so the engineer has left plenty of room for growth.

## Operational View Versus Design View of Subnetting

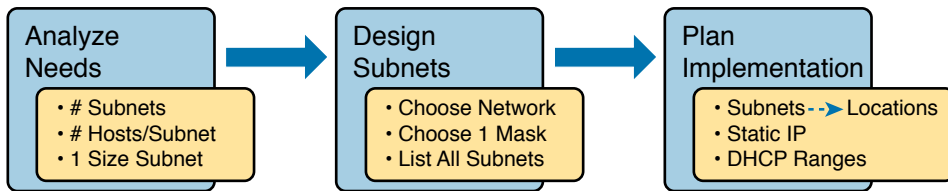
Most IT jobs require you to work with subnetting from an operational view. That is, someone else, before you got the job, designed how IP addressing and subnetting would work for that particular enterprise network. You need to interpret what someone else has already chosen.

To fully understand IP addressing and subnetting, you need to think about subnetting from both a design and operational perspective. For example, Figure 11-1 simply states that in all these subnets, the first three octets must be equal. Why was that convention chosen? What

alternatives exist? Would those alternatives be better for your internetwork today? All these questions relate more to subnetting design rather than to operation.

To help you see both perspectives, this chapter focuses more on design issues by moving through the entire design process for the purpose of introducing the bigger picture of IP subnetting. The next three chapters each take one topic from this chapter and examine it more closely but more from an operational perspective: how to use those ideas in real networks.

The remaining three main sections of this chapter examine each of the steps listed in Figure 11-2, in sequence.



**Figure 11-2** Subnet Planning, Design, and Implementation Tasks

## Analyze Subnetting and Addressing Needs

This section discusses the meaning of four basic questions that can be used to analyze the addressing and subnetting needs for any new or changing enterprise network:

1. Which hosts should be grouped together into a subnet?
2. How many subnets does this internetwork require?
3. How many host IP addresses does each subnet require?
4. Will we use a single subnet size for simplicity, or not?

### Rules About Which Hosts Are in Which Subnet

Every device that connects to an IP internetwork needs to have an IP address. These devices include computers used by end users, servers, mobile phones, laptops, IP phones, tablets, and networking devices like routers, switches, and firewalls. In short, any device that uses IP to send and receive packets needs an IP address.

**NOTE** In a discussion of IP addressing, the term *network* has specific meaning: a Class A, B, or C IP network. To avoid confusion with that use of the term *network*, this book uses the terms *internetwork* and *enterprise network* when referring to a collection of hosts, routers, switches, and so on.

Answers to the “Do I Know This Already?” quiz:

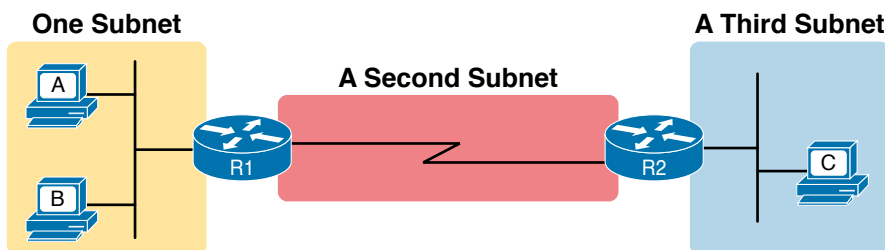
**1 B, D 2 D 3 B, C 4 A, C 5 A, D, E 6 A, C 7 B**

The IP addresses must be assigned according to some basic rules—and for good reasons. To make routing work efficiently, IP addressing rules group addresses into groups called subnets. The rules are as follows:

**Key  
Topic**

- Addresses in the same subnet are not separated by a router.
- Addresses in different subnets are separated by at least one router.

Figure 11-3 shows the general concept, with hosts A and B in one subnet and host C in another. In particular, note that hosts A and B are not separated from each other by any routers. However, host C, separated from A and B by at least one router, must be in a different subnet.



**Figure 11-3** PC A and B in One Subnet and PC C in a Different Subnet

The idea that hosts on the same link must be in the same subnet is much like the postal code concept. All mailing addresses in the same town use the same postal code (ZIP codes in the United States). Addresses in another town, whether relatively nearby or on the other side of the country, have a different postal code. The postal code gives the postal service a better ability to automatically sort the mail to deliver it to the right location. For the same general reasons, hosts on the same LAN are in the same subnet, and hosts in different LANs are in different subnets.

Note that the point-to-point WAN link in the figure also needs a subnet. Figure 11-3 shows Router R1 connected to the LAN subnet on the left and to a WAN subnet on the right. Router R2 connects to that same WAN subnet. To do so, both R1 and R2 will have IP addresses on their WAN interfaces, and the addresses will be in the same subnet. (An Ethernet WAN link has the same IP addressing needs, with each of the two routers having an IP address in the same subnet.)

The Ethernet LANs in Figure 11-3 also show a slightly different style of drawing, using simple lines with no Ethernet switch. Drawings of Ethernet LANs when the details of the LAN switches do not matter simply show each device connected to the same line, as shown in Figure 11-3. (This kind of drawing mimics the original Ethernet cabling before switches and hubs existed.)

Finally, because the routers' main job is to forward packets from one subnet to another, routers typically connect to multiple subnets. For example, in this case, Router R1 connects to one LAN subnet on the left and one WAN subnet on the right. To do so, R1 will be configured with two different IP addresses, one per interface. These addresses will be in different subnets because the interfaces connect the router to different subnets.

## Determining the Number of Subnets

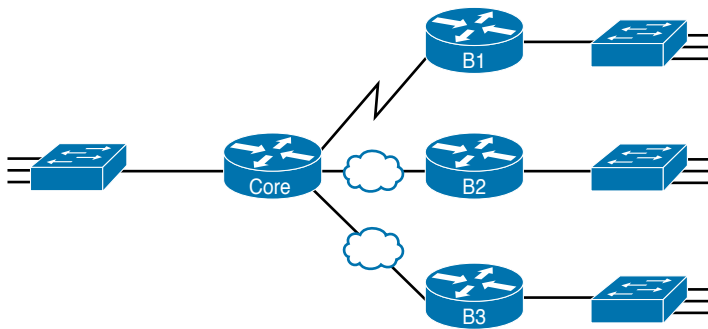
To determine the number of subnets required, the engineer must think about the internetwork as documented and count the locations that need a subnet. To do so, the engineer requires access to network diagrams, VLAN configuration details, and details about WAN links. For the types of links discussed in this book, you should plan for one subnet for every

### Key Topic

- VLAN
- Point-to-point serial link
- Ethernet WAN (Ethernet Line Service)

**NOTE** Other WAN technologies outside the scope of the CCNA exam topics allow subnetting options other than one subnet per pair of routers on the WAN (as shown here). However, this book only uses point-to-point WAN technologies—serial links and Ethernet WAN links—that have one subnet for each point-to-point WAN connection between two routers.

For example, imagine that the network planner has only Figure 11-4 on which to base the subnet design.



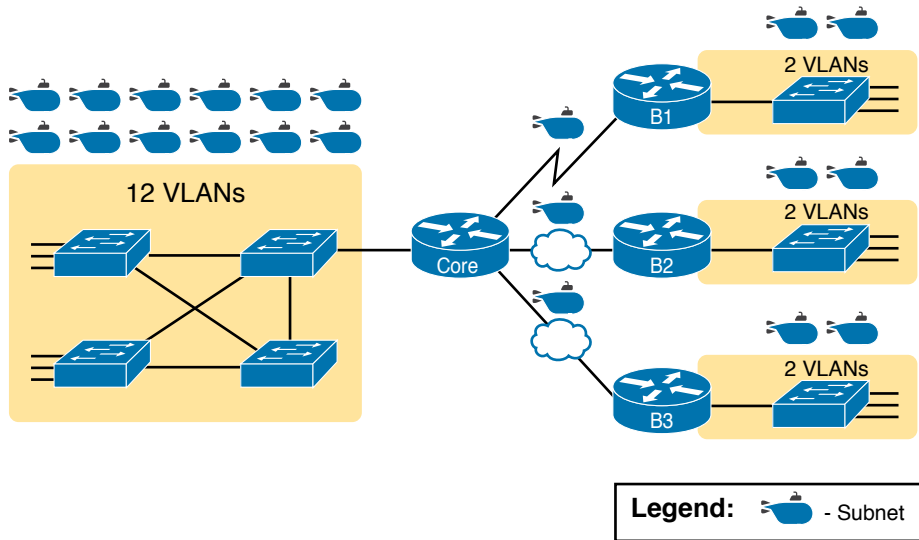
**Figure 11-4** *Four-Site Internetwork with Small Central Site*

The number of subnets required cannot be fully predicted with only this figure. Certainly, three subnets will be needed for the WAN links, one per link. However, each LAN switch can be configured with a single VLAN or with multiple VLANs. You can be certain that you need at least one subnet for the LAN at each site, but you might need more.

Next, consider the more detailed version of the same figure shown in Figure 11-5. In this case, the figure shows VLAN counts in addition to the same Layer 3 topology (the routers and the links connected to the routers). It also shows that the central site has many more switches, but the key fact on the left, regardless of how many switches exist, is that the central site has a total of 12 VLANs. Similarly, the figure lists each branch as having two VLANs. Along with the same three WAN subnets, this internetwork requires 21 subnets.

Finally, in a real job, you would consider the needs today as well as how much growth you expect in the internetwork over time. Any subnetting plan should include a reasonable estimate of the number of subnets you need to meet future needs.





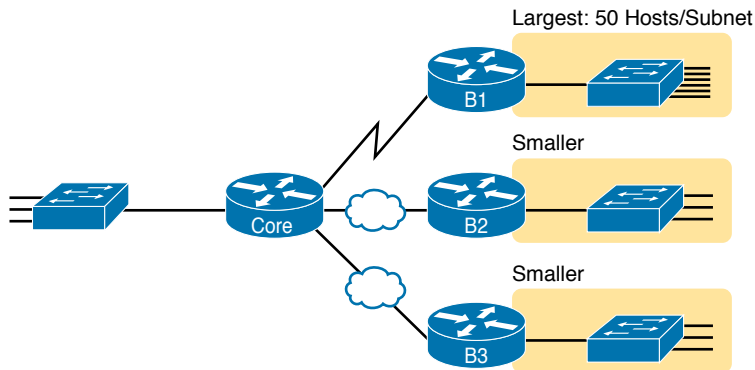
**Figure 11-5** *Four-Site Internetwork with Larger Central Site*

### Determining the Number of Hosts per Subnet

Determining the number of hosts per subnet requires knowing a few simple concepts and then doing a lot of research and questioning. Every device that connects to a subnet needs an IP address. For a totally new network, you can look at business plans—numbers of people at the site, devices on order, and so on—to get some idea of the possible devices. When expanding an existing network to add new sites, you can use existing sites as a point of comparison and then find out which sites will get bigger or smaller. And don't forget to count the router interface IP address in each subnet and the switch IP address used to remotely manage the switch.

Instead of gathering data for each and every site, planners often just use a few typical sites for planning purposes. For example, maybe you have some large sales offices and some small sales offices. You might dig in and learn a lot about only one large sales office and only one small sales office. Add that analysis to the fact that point-to-point links need a subnet with just two addresses, plus any analysis of more one-of-a-kind subnets, and you have enough information to plan the addressing and subnetting design.

For example, in Figure 11-6, the engineer has built a diagram that shows the number of hosts per LAN subnet in the largest branch, B1. For the two other branches, the engineer did not bother to dig to find out the number of required hosts. As long as the number of required IP addresses at sites B2 and B3 stays below the estimate of 50, based on larger site B1, the engineer can plan for 50 hosts in each branch LAN subnet and have plenty of addresses per subnet.



**Figure 11-6** Large Branch B1 with 50 Hosts/Subnet

## One Size Subnet Fits All—Or Not

The final choice in the initial planning step is to decide whether you will use a simpler design by using a one-size-subnet-fits-all philosophy. A subnet's size, or length, is simply the number of usable IP addresses in the subnet. A subnetting design can either use one size subnet or varied sizes of subnets, with pros and cons for each choice.

### Defining the Size of a Subnet

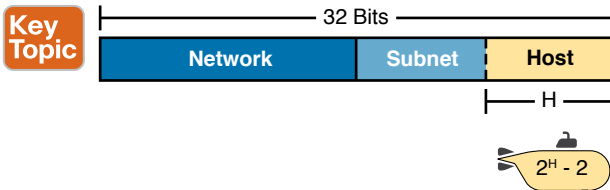
Before you finish this book, you will learn all the details of how to determine the size of the subnet. For now, you just need to know a few specific facts about the size of subnets. Chapter 12, “Analyzing Classful IPv4 Networks,” and Chapter 13, “Analyzing Subnet Masks,” give you a progressively deeper knowledge of the details.

The engineer assigns each subnet a *subnet mask*, and that mask, among other things, defines the size of that subnet. The mask sets aside a number of *host bits* whose purpose is to number different host IP addresses in that subnet. Because you can number  $2^x$  things with  $x$  bits, if the mask defines  $H$  host bits, the subnet contains  $2^H$  unique numeric values.

However, the subnet's size is not  $2^H$ . It's  $2^H - 2$  because two numbers in each subnet are reserved for other purposes. Each subnet reserves the numerically lowest value for the *subnet number* and the numerically highest value as the *subnet broadcast address*. As a result, the number of usable IP addresses per subnet is  $2^H - 2$ .

**NOTE** The terms *subnet number*, *subnet ID*, and *subnet address* all refer to the number that represents or identifies a subnet.

Figure 11-7 shows the general concept behind the three-part structure of an IP address, focusing on the host part and the resulting subnet size.



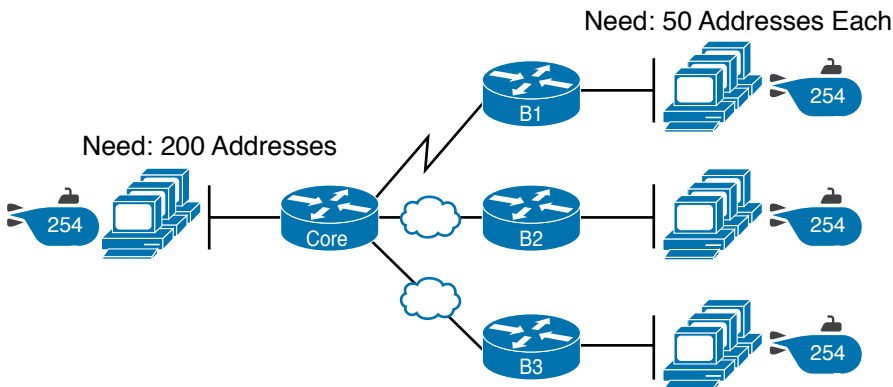
**Figure 11-7** Subnet Size Concepts

### One Size Subnet Fits All

To choose to use a single-size subnet in an enterprise network, you must use the same mask for all subnets because the mask defines the size of the subnet. But which mask?

One requirement to consider when choosing that one mask is this: that one mask must provide enough host IP addresses to support the largest subnet. To do so, the number of host bits ( $H$ ) defined by the mask must be large enough so that  $2^H - 2$  is larger than (or equal to) the number of host IP addresses required in the largest subnet.

For example, consider Figure 11-8. It shows the required number of hosts per LAN subnet. (The figure ignores the subnets on the WAN links, which require only two IP addresses each.) The branch LAN subnets require only 50 host addresses, but the main site LAN subnet requires 200 host addresses. To accommodate the largest subnet, you need at least 8 host bits. Seven host bits would not be enough because  $2^7 - 2 = 126$ . Eight host bits would be enough because  $2^8 - 2 = 254$ , which is more than enough to support 200 hosts in a subnet.



**Figure 11-8** Network Using One Subnet Size

What's the big advantage when using a single-size subnet? Operational simplicity. In other words, keeping it simple. Everyone on the IT staff who has to work with networking can get used to working with one mask—and one mask only. Staff members will be able to answer all subnetting questions more easily because everyone gets used to doing subnetting math with that one mask.

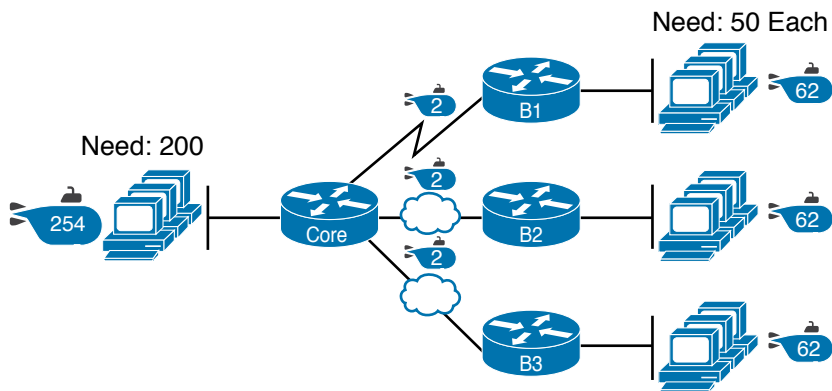
The big disadvantage for using a single-size subnet is that it wastes IP addresses. For example, in Figure 11-8, all the branch LAN subnets support 254 addresses, while the largest branch subnet needs only 50 addresses. The WAN subnets only need two IP addresses, but each supports 254 addresses, again wasting more IP addresses.

The wasted IP addresses do not actually cause a problem in most cases, however. Most organizations use private IP networks in their enterprise internetworks, and a single Class A or Class B private network can supply plenty of IP addresses, even with the waste.

### Multiple Subnet Sizes (Variable-Length Subnet Masks)

To create multiple sizes of subnets in one Class A, B, or C network, the engineer must create some subnets using one mask, some with another, and so on. Different masks mean different numbers of host bits, and a different number of hosts in some subnets based on the  $2^H - 2$  formula.

For example, consider the requirements listed earlier in Figure 11-8. It showed one LAN subnet on the left that needs 200 host addresses, three branch subnets that need 50 addresses, and three WAN links that need two addresses. To meet those needs, but waste fewer IP addresses, three subnet masks could be used, creating subnets of three different sizes, as shown in Figure 11-9.



**Figure 11-9** *Three Masks, Three Subnet Sizes*

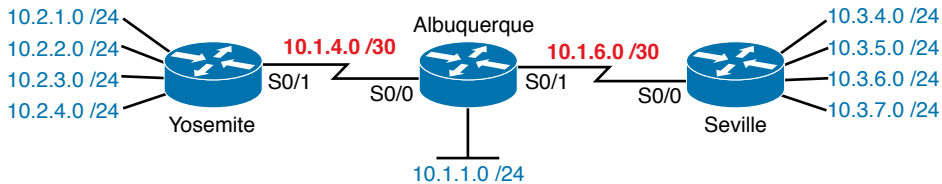
The smaller subnets now waste fewer IP addresses compared to the design shown earlier in Figure 11-8. The subnets on the right that need 50 IP addresses have subnets with 6 host bits, for  $2^6 - 2 = 62$  available addresses per subnet. The WAN links use masks with 2 host bits, for  $2^2 - 2 = 2$  available addresses per subnet.

However, some are still wasted because you cannot set the size of the subnet as some arbitrary size. All subnets will be a size based on the  $2^H - 2$  formula, with H being the number of host bits defined by the mask for each subnet.

### One Mask for All Subnets, or More Than One

For the most part, this book explains subnetting using designs that use a single mask, creating a single subnet size for all subnets. Why? First, it makes the process of learning subnetting easier. Second, some types of analysis that you can do about a network—specifically, calculating the number of subnets in the classful network—only make sense when a single mask is used.

However, you still need to be ready to work with designs that use more than one mask in different subnets of the same Class A, B, or C network. In fact, a design that does just that is said to be using *variable-length subnet masks (VLSM)*. For example, the internetwork in Figure 11-10 shows 11 subnets, two with a mask of /30, and nine with a mask of /24. By using more than one mask among all the subnets of one Class A network (10.0.0.0), the design uses VLSM.



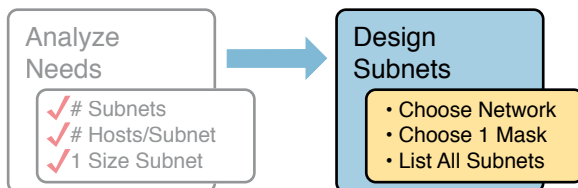
**Figure 11-10** Internetwork with VLSM: Network 10.0.0.0, >1 Mask

For the current CCNA 200-301 exam, using VLSM causes no issues, although it does cause problems with some older routing protocols. The only routing protocol included in the CCNA blueprint (OSPF) works the same regardless of whether the design uses VLSM. Just be aware of the term and what it means and that it should not impact the features included in the current CCNA exam.

**NOTE** VLSM has been featured in the CCNA exam topics in the past. If you want to read a little more about VLSM, check out Appendix N, “Variable-Length Subnet Masks,” on the companion website for this book.

## Make Design Choices

Now that you know how to analyze the IP addressing and subnetting needs, the next major step examines how to apply the rules of IP addressing and subnetting to those needs and make some choices. In other words, now that you know how many subnets you need and how many host addresses you need in the largest subnet, how do you create a useful subnetting design that meets those requirements? The short answer is that you need to do the three tasks shown on the right side of Figure 11-11.



**Figure 11-11** Input to the Design Phase, and Design Questions to Answer

## Choose a Classful Network

In the original design for what we know of today as the Internet, companies used registered *public classful IP networks* when implementing TCP/IP inside the company. By the

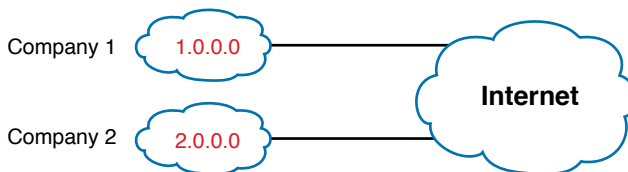
mid-1990s, an alternative became more popular: *private IP networks*. This section discusses the background behind these two choices because it impacts the choice of what IP network a company will then subnet and implement in its enterprise internetwork.

## Public IP Networks

The original design of the Internet required that any company that connected to the Internet had to use a *registered public IP network*. To do so, the company would complete some paperwork, describing the enterprise's internetwork and the number of hosts existing, plus plans for growth. After submitting the paperwork, the company would receive an assignment of either a Class A, B, or C network.

Public IP networks—and the administrative processes surrounding them—ensure that all the companies that connect to the Internet all use unique IP addresses. In particular, after a public IP network is assigned to a company, only that company should use the addresses in that network. That guarantee of uniqueness means that Internet routing can work well because there are no duplicate public IP addresses.

For example, consider the example shown in Figure 11-12. Company 1 has been assigned public Class A network 1.0.0.0, and company 2 has been assigned public Class A network 2.0.0.0. Per the original intent for public addressing in the Internet, after these public network assignments have been made, no other companies can use addresses in Class A networks 1.0.0.0 or 2.0.0.0.



**Figure 11-12** Two Companies with Unique Public IP Networks

This original address assignment process ensured unique IP addresses across the entire planet. The idea is much like the fact that your telephone number should be unique in the universe, your postal mailing address should also be unique, and your email address should also be unique. If someone calls you, your phone rings, but no one else's phone rings. Similarly, if company 1 is assigned Class A network 1.0.0.0, and the engineers at Company 1 assign address 1.1.1.1 to a particular PC, that address should be unique in the universe. A packet sent through the Internet to destination 1.1.1.1 should only arrive at this one PC inside company 1, instead of being delivered to some other host.

## Growth Exhausts the Public IP Address Space

By the early 1990s, the world was running out of public IP networks that could be assigned. During most of the 1990s, the number of hosts newly connected to the Internet was growing at a double-digit pace *per month*. Companies kept following the rules, asking for public IP networks, and it was clear that the current address-assignment scheme could not continue without some changes. Simply put, the number of Class A, B, and C networks supported by the 32-bit address in IP version 4 (IPv4) was not enough to support one public classful network per organization, while also providing enough IP addresses in each company.

**NOTE** The universe has run out of public IPv4 addresses in a couple of significant ways. IANA, which assigns public IPv4 address blocks to the five Regional Internet Registries (RIR) around the globe, assigned the last of the IPv4 address spaces in early 2011. By 2015, ARIN, the RIR for North America, exhausted its supply of IPv4 addresses, so companies must return unused public IPv4 addresses to ARIN before they have more to assign to new companies. Try an online search for “ARIN depletion” to see pages about the current status of available IPv4 address space for just one RIR example.

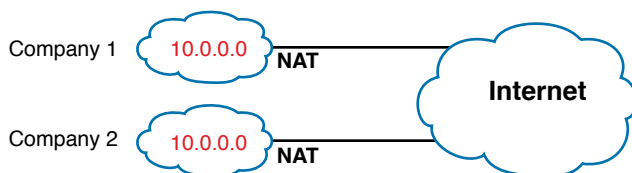
The Internet community worked hard during the 1990s to solve this problem, coming up with several solutions, including the following:

**Key  
Topic**

- A new version of IP (IPv6), with much larger addresses (128 bit)
- Assigning a subset of a public IP network to each company, instead of an entire public IP network, to reduce waste, using a feature called “Classless Interdomain Routing” (CIDR)
- Network Address Translation (NAT), which allows the use of private IP networks

These three solutions matter to real networks today. However, to stay focused on the topic of subnet design, this chapter focuses on the third option, and in particular, the private IP networks that can be used by an enterprise when also using NAT. (Be aware that Chapter 10, “Network Address Translation” in *CCNA 200-301 Official Cert Guide, Volume 2*, gives more detail about the last two bullets in the list, while Part VII of this book discusses the first bullet item (IPv6) in more depth.

Focusing on the third item in the bullet list, NAT allows multiple companies to use the exact same *private IP network*, using the same IP addresses as other companies while still connecting to the Internet. For example, Figure 11-13 shows the same two companies connecting to the Internet as in Figure 11-12, but now with both using the same private Class A network 10.0.0.0.



**Figure 11-13** Reusing the Same Private Network 10.0.0.0 with NAT

Both companies use the same classful IP network (10.0.0.0). Both companies can implement their subnet design internal to their respective enterprise internetworks, without discussing their plans. The two companies can even use the exact same IP addresses inside network 10.0.0.0. And amazingly, at the same time, both companies can even communicate with each other through the Internet.

The technology called Network Address Translation makes it possible for companies to reuse the same IP networks, as shown in Figure 11-13. NAT does this by translating the IP addresses inside the packets as they go from the enterprise to the Internet, using a small number of public IP addresses to support tens of thousands of private IP addresses. That one bit of information is not enough to understand how NAT works; however, to keep the focus

on subnetting, the book defers the discussion of how NAT works until *CCNA 200-301 Official Cert Guide, Volume 2*. For now, accept that most companies use NAT, and therefore, they can use private IP networks for their internetworks.

Private IP Networks

When using NAT—and almost every organization that connects to the Internet uses NAT—the company can simply pick one or more of the private IP networks from the list of reserved private IP network numbers. RFC 1918 defines the list of available private IP networks, which is summarized in Table 11-2.

Table 11-2 RFC 1918 Private Address Space

Class of Networks	Private IP Networks	Number of Networks
A	10.0.0.0	1
B	172.16.0.0 through 172.31.0.0	16
C	192.168.0.0 through 192.168.255.0	256

**NOTE** According to an informal survey I ran on my blog a few years back, about half of the respondents said that their networks use private Class A network 10.0.0.0, as opposed to other private networks or public networks.

From the perspective of making IPv4 work for the entire world, private IP networks have helped preserve and extend IPv4 and its use in every enterprise and throughout the Internet. In particular, private networks have improved IPv4’s implementation worldwide by



- **Avoiding Using Another Organization’s Public Address Range for Private Networks:** Some organizations have a part of their networks that need zero Internet access. The hosts in that part of their network need IP addresses. RFC 1918 suggests that truly private networks—that is, networks with no need for Internet connectivity—use addresses from the RFC 1918 list of private networks.
- **Avoiding/Delaying IPv4 Address Exhaustion:** To delay the day in which all public IPv4 addresses were assigned to organizations as public addresses, RFC 1918 calls for the use of NAT along with private networks for the addresses internal to an organization.
- **Reducing Internet Routers’ Routing Table Size:** Using private networks also helps reduce the size of the IP routing tables in Internet routers. For instance, routers in the Internet do not need routes for the private IP networks used inside organizations (in fact, ISPs filter those routes).

Choosing an IP Network During the Design Phase

Today, some organizations use private IP networks along with NAT, and some use public IP networks. Most new enterprise internetworks use private IP addresses throughout the network, along with NAT, as part of the connection to the Internet. Those organizations that already have registered public IP networks—often obtained before the addresses started



running short in the early 1990s—can continue to use those public addresses throughout their enterprise networks.

After the choice to use a private IP network has been made, just pick one that has enough IP addresses. You can have a small internetwork and still choose to use private Class A network 10.0.0.0. It might seem wasteful to choose a Class A network that has over 16 million IP addresses, especially if you need only a few hundred. However, there's no penalty or problem with using a private network that is too large for your current or future needs.

For the purposes of this book, most examples use private IP network numbers. For the design step to choose a network number, just choose a private Class A, B, or C network from the list of RFC 1918 private networks.

Regardless, from a math and concept perspective, the methods to subnet a public IP network versus a private IP network are the same.

## Choose the Mask

If a design engineer followed the topics in this chapter so far, in order, he would know the following:

- The number of subnets required
- The number of hosts/subnet required
- That a choice was made to use only one mask for all subnets so that all subnets are the same size (same number of hosts/subnet)
- The classful IP network number that will be subnetted

This section completes the design process, at least the parts described in this chapter, by discussing how to choose that one mask to use for all subnets. First, this section examines default masks, used when a network is not subnetted, as a point of comparison. Next, the concept of borrowing host bits to create subnet bits is explored. Finally, this section ends with an example of how to create a subnet mask based on the analysis of the requirements.

## Classful IP Networks Before Subnetting

Before an engineer subnets a classful network, the network is a single group of addresses. In other words, the engineer has not yet subdivided the network into many smaller subsets called *subnets*.

When thinking about an unsubnetted classful network, the addresses in a network have only two parts: the network part and host part. Comparing any two addresses in the classful network:

- The addresses have the same value in the network part.
- The addresses have different values in the host part.

The actual sizes of the network and host part of the addresses in a network can be easily predicted, as shown in Figure 11-14.

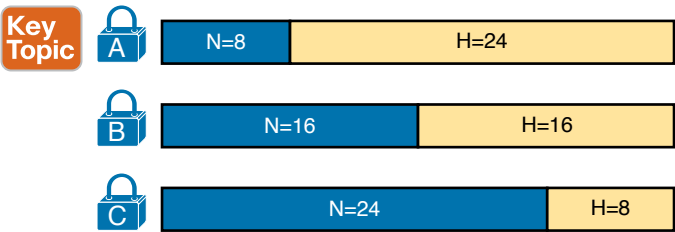


Figure 11-14 Format of Unsubnetted Class A, B, and C Networks

In Figure 11-14, N and H represent the number of network and host bits, respectively. Class rules define the number of network octets (1, 2, or 3) for Classes A, B, and C, respectively; the figure shows these values as a number of bits. The number of host octets is 3, 2, or 1, respectively.

Continuing the analysis of classful network before subnetting, the number of addresses in one classful IP network can be calculated with the same  $2^H - 2$  formula previously discussed. In particular, the size of an unsubnetted Class A, B, or C network is as follows:

- Class A:  $2^{24} - 2 = 16,777,214$
- Class B:  $2^{16} - 2 = 65,534$
- Class C:  $2^8 - 2 = 254$

Borrowing Host Bits to Create Subnet Bits

To subnet a network, the designer thinks about the network and host parts, as shown in Figure 11-15, and then the engineer adds a third part in the middle: the subnet part. However, the designer cannot change the size of the network part or the size of the entire address (32 bits). To create a subnet part of the address structure, the engineer borrows bits from the host part. Figure 11-15 shows the general idea.

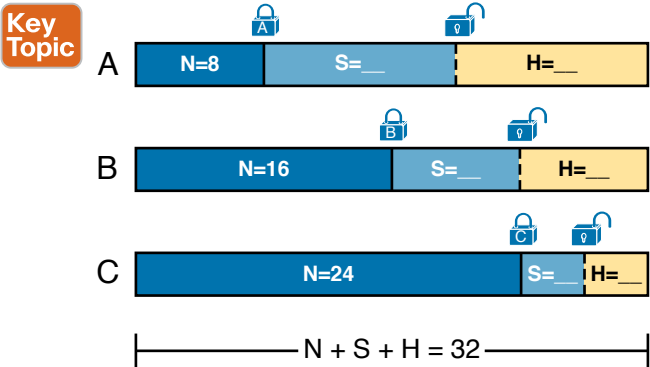


Figure 11-15 Concept of Borrowing Host Bits

Figure 11-15 shows a rectangle that represents the subnet mask. N, representing the number of network bits, remains locked at 8, 16, or 24, depending on the class. Conceptually, the designer moves a (dashed) dividing line into the host field, with subnet bits (S) between the

network and host parts, and the remaining host bits (H) on the right. The three parts must add up to 32 because IPv4 addresses consist of 32 bits.

## Choosing Enough Subnet and Host Bits

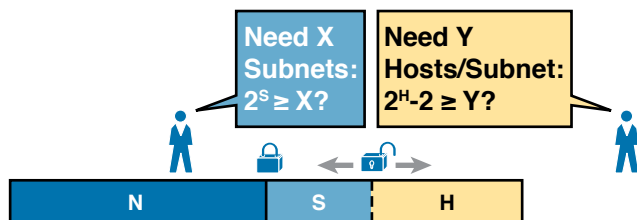
The design process requires a choice of where to place the dashed line shown in Figure 11-15. But what is the right choice? How many subnet and host bits should the designer choose? The answers hinge on the requirements gathered in the early stages of the planning process:

- Number of subnets required
- Number of hosts/subnet

The bits in the subnet part create a way to uniquely number the different subnets that the design engineer wants to create. With 1 subnet bit, you can number  $2^1$  or 2 subnets. With 2 bits,  $2^2$  or 4 subnets, with 3 bits,  $2^3$  or 8 subnets, and so on. The number of subnet bits must be large enough to uniquely number all the subnets, as determined during the planning process.

At the same time, the remaining number of host bits must also be large enough to number the host IP addresses in the largest subnet. Remember, in this chapter, we assume the use of a single mask for all subnets. This single mask must support both the required number of subnets and the required number of hosts in the largest subnet. Figure 11-16 shows the concept.

### Key Topic



**Figure 11-16** *Borrowing Enough Subnet and Host Bits*

Figure 11-16 shows the idea of the designer choosing a number of subnet (S) and host (H) bits and then checking the math.  $2^S$  must be more than the number of required subnets, or the mask will not supply enough subnets in this IP network. Also,  $2^H - 2$  must be more than the required number of hosts/subnet.

**NOTE** The idea of calculating the number of subnets as  $2^S$  applies only in cases where a single mask is used for all subnets of a single classful network, as is being assumed in this chapter.

To effectively design masks, or to interpret masks that were chosen by someone else, you need a good working memory of the powers of 2. Appendix A, “Numeric Reference Tables,” lists a table with powers of 2 up through  $2^{32}$  for your reference.

Example Design: 172.16.0.0, 200 Subnets, 200 Hosts

To help make sense of the theoretical discussion so far, consider an example that focuses on the design choice for the subnet mask. In this case, the planning and design choices so far tell us the following:

- Use a single mask for all subnets.
- Plan for 200 subnets.
- Plan for 200 host IP addresses per subnet.
- Use private Class B network 172.16.0.0.

To choose the mask, the designer asks this question:

How many subnet (S) bits do I need to number 200 subnets?

You can see that  $S = 7$  is not large enough ( $2^7 = 128$ ), but  $S = 8$  is enough ( $2^8 = 256$ ). So, you need *at least* 8 subnet bits.

Next, the designer asks a similar question, based on the number of hosts per subnet:

How many host (H) bits do I need to number 200 hosts per subnet?

The math is basically the same, but the formula subtracts 2 when counting the number of hosts/subnet. You can see that  $H = 7$  is not large enough ( $2^7 - 2 = 126$ ), but  $H = 8$  is enough ( $2^8 - 2 = 254$ ).

Only one possible mask meets all the requirements in this case. First, the number of network bits (N) must be 16 because the design uses a Class B network. The requirements tell us that the mask needs at least 8 subnet bits and at least 8 host bits. The mask only has 32 bits in it; Figure 11-17 shows the resulting mask.

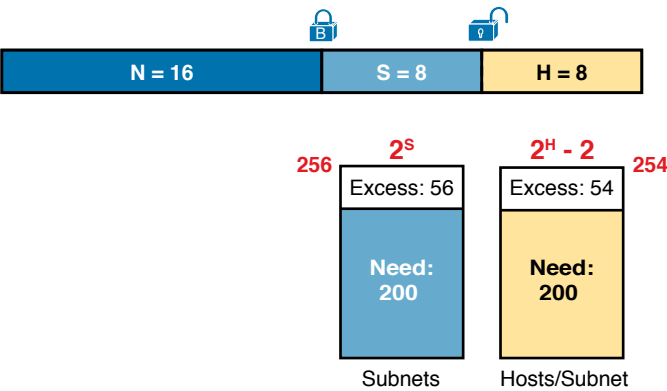


Figure 11-17 Example Mask Choice,  $N = 16$ ,  $S = 8$ ,  $H = 8$

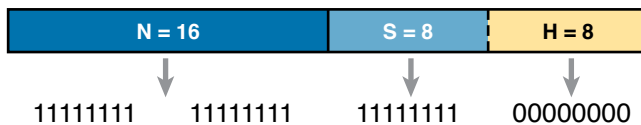
Masks and Mask Formats

Although engineers think about IP addresses in three parts when making design choices (network, subnet, and host), the subnet mask gives the engineer a way to communicate those design choices to all the devices in the subnet.

The subnet mask is a 32-bit binary number with a number of binary 1s on the left and with binary 0s on the right. By definition, the number of binary 0s equals the number of host bits; in fact, that is exactly how the mask communicates the idea of the size of the host part of the addresses in a subnet. The beginning bits in the mask equal binary 1, with those bit positions representing the combined network and subnet parts of the addresses in the subnet.

Because the network part always comes first, then the subnet part, and then the host part, the subnet mask, in binary form, cannot have interleaved 1s and 0s. Each subnet mask has one unbroken string of binary 1s on the left, with the rest of the bits as binary 0s.

After the engineer chooses the classful network and the number of subnet and host bits in a subnet, creating the binary subnet mask is easy. Just write down N 1s, S 1s, and then H 0s (assuming that N, S, and H represent the number of network, subnet, and host bits). Figure 11-18 shows the mask based on the previous example, which subnets a Class B network by creating 8 subnet bits, leaving 8 host bits.



**Figure 11-18** *Creating the Subnet Mask—Binary—Class B Network*

In addition to the binary mask shown in Figure 11-18, masks can also be written in two other formats: the familiar *dotted-decimal notation* (DDN) seen in IP addresses and an even briefer *prefix* notation. Chapter 13, “Analyzing Subnet Masks,” discusses these formats and how to convert between the different formats.

## Build a List of All Subnets

Building a list of all subnets, the final task of the subnet design step, determines the actual subnets that can be used, based on all the earlier choices. The earlier design work determined the Class A, B, or C network to use, and the (one) subnet mask to use that supplies enough subnets and enough host IP addresses per subnet. But what are those subnets? How do you identify or describe a subnet? This section answers these questions.

A subnet consists of a group of consecutive numbers. Most of these numbers can be used as IP addresses by hosts. However, each subnet reserves the first and last numbers in the group, and these two numbers cannot be used as IP addresses. In particular, each subnet contains the following:

### Key Topic

- **Subnet number:** Also called the *subnet ID* or *subnet address*, this number identifies the subnet. It is the numerically smallest number in the subnet. It cannot be used as an IP address by a host.
- **Subnet broadcast:** Also called the *subnet broadcast address* or *directed broadcast address*, this is the last (numerically highest) number in the subnet. It also cannot be used as an IP address by a host.
- **IP addresses:** All the numbers between the subnet ID and the subnet broadcast address can be used as a host IP address.

For example, consider the earlier case in which the design results were as follows:

Network     172.16.0.0 (Class B)  
Mask        255.255.255.0 (for all subnets)

With some math, the facts about each subnet that exists in this Class B network can be calculated. In this case, Table 11-3 shows the first 10 such subnets. It then skips many subnets and shows the last two (numerically largest) subnets.

**Table 11-3** First 10 Subnets, Plus the Last Few, from 172.16.0.0, 255.255.255.0

Subnet Number	IP Addresses	Broadcast Address
172.16.0.0	172.16.0.1 – 172.16.0.254	172.16.0.255
172.16.1.0	172.16.1.1 – 172.16.1.254	172.16.1.255
172.16.2.0	172.16.2.1 – 172.16.2.254	172.16.2.255
172.16.3.0	172.16.3.1 – 172.16.3.254	172.16.3.255
172.16.4.0	172.16.4.1 – 172.16.4.254	172.16.4.255
172.16.5.0	172.16.5.1 – 172.16.5.254	172.16.5.255
172.16.6.0	172.16.6.1 – 172.16.6.254	172.16.6.255
172.16.7.0	172.16.7.1 – 172.16.7.254	172.16.7.255
172.16.8.0	172.16.8.1 – 172.16.8.254	172.16.8.255
172.16.9.0	172.16.9.1 – 172.16.9.254	172.16.9.255
Skipping many...		
172.16.254.0	172.16.254.1 – 172.16.254.254	172.16.254.255
172.16.255.0	172.16.255.1 – 172.16.255.254	172.16.255.255

After you have the network number and the mask, calculating the subnet IDs and other details for all subnets requires some math. In real life, most people use subnet calculators or subnet-planning tools. For the CCNA exam, you need to be ready to find this kind of information.

If you want to dig a little deeper in preparation for CCNP Enterprise or other studies related to IP routing, consider using Appendix L, “Subnet Design,” on the book’s companion website, which shows you how to find all the subnets of a given network.

## Plan the Implementation

The next step, planning the implementation, is the last step before actually configuring the devices to create a subnet. The engineer first needs to choose where to use each subnet. For example, at a branch office in a particular city, which subnet from the subnet planning chart (Table 11-3) should be used for each VLAN at that site? Also, for any interfaces that require static IP addresses, which addresses should be used in each case? Finally, what range of IP addresses from inside each subnet should be configured in the DHCP server, to be

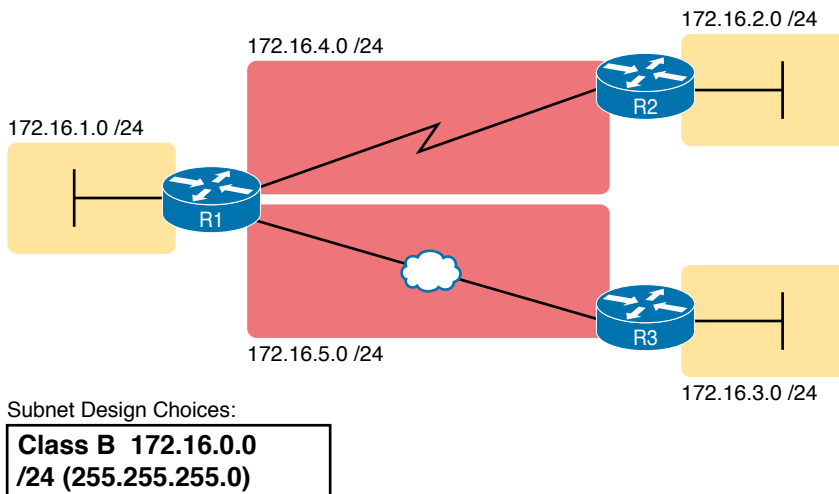
dynamically leased to hosts for use as their IP address? Figure 11-19 summarizes the list of implementation planning tasks.



**Figure 11-19** *Facts Supplied to the Plan Implementation Step*

## Assigning Subnets to Different Locations

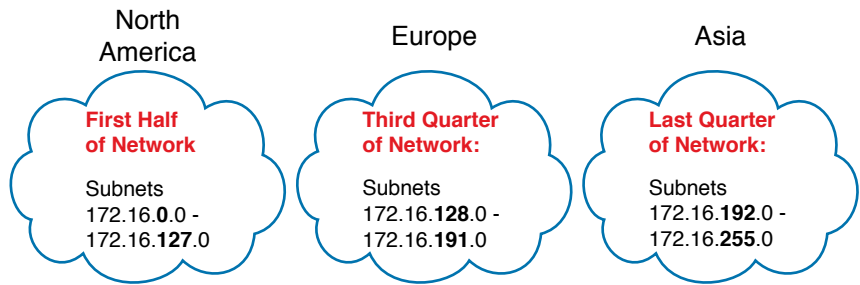
The job is simple: Look at your network diagram, identify each location that needs a subnet, and pick one from the table you made of all the possible subnets. Then, track it so that you know which ones you use where, using a spreadsheet or some other purpose-built subnet-planning tool. That's it! Figure 11-20 shows a sample of a completed design using Table 11-3, which happens to match the initial design sample shown way back in Figure 11-1.



**Figure 11-20** *Example of Subnets Assigned to Different Locations*

Although this design could have used any five subnets from Table 11-3, in real networks, engineers usually give more thought to some strategy for assigning subnets. For example, you might assign all LAN subnets lower numbers and WAN subnets higher numbers. Or you might slice off large ranges of subnets for different divisions of the company. Or you might follow that same strategy but ignore organizational divisions in the company, paying more attention to geographies.

For example, for a U.S.-based company with a smaller presence in both Europe and Asia, you might plan to reserve ranges of subnets based on continent. This kind of choice is particularly useful when later trying to use a feature called route summarization. Figure 11-21 shows the general benefit of placing addressing in the network for easier route summarization, using the same subnets from Table 11-3 again.



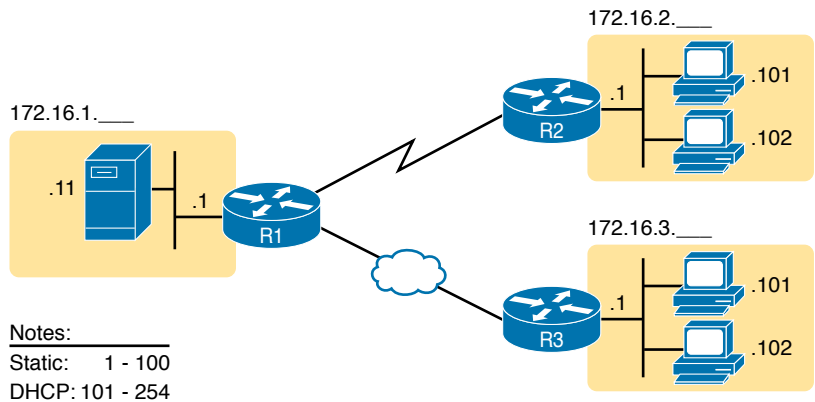
**Figure 11-21** *Reserving 50 Percent of Subnets for North America and 25 Percent Each for Europe and Asia*

**Choose Static and Dynamic Ranges per Subnet**

Devices receive their IP address and mask assignment in one of two ways: dynamically by using Dynamic Host Configuration Protocol (DHCP) or statically through configuration. For DHCP to work, the network engineer must tell the DHCP server the subnets for which it must assign IP addresses. In addition, that configuration limits the DHCP server to only a subset of the addresses in the subnet. For static addresses, you simply configure the device to tell it what IP address and mask to use.

To keep things as simple as possible, most shops use a strategy to separate the static IP addresses on one end of each subnet, and the DHCP-assigned dynamic addresses on the other. It does not really matter whether the static addresses sit on the low end of the range of addresses or the high end.

For example, imagine that the engineer decides that, for the LAN subnets in Figure 11-20, the DHCP pool comes from the high end of the range, namely, addresses that end in .101 through .254. (The address that ends in .255 is, of course, reserved.) The engineer also assigns static addresses from the lower end, with addresses ending in .1 through .100. Figure 11-22 shows the idea.



**Figure 11-22** *Static from the Low End and DHCP from the High End*

Figure 11-22 shows all three routers with statically assigned IP addresses that end in .1. The only other static IP address in the figure is assigned to the server on the left, with address 172.16.1.11 (abbreviated simply as .11 in the figure).



On the right, each LAN has two PCs that use DHCP to dynamically lease their IP addresses. DHCP servers often begin by leasing the addresses at the bottom of the range of addresses, so in each LAN, the hosts have leased addresses that end in .101 and .102, which are at the low end of the range chosen by design.

## Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 11-4 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

**Table 11-4** Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review memory tables		Website

## Review All the Key Topics

### Key Topic

**Table 11-5** Key Topics for Chapter 11

Key Topic Element	Description	Page Number
List	Key facts about subnets	269
List	Rules about what places in a network topology need a subnet	270
Figure 11-7	Locations of the network, subnet, and host parts of an IPv4 address	273
List	Features that extended the life of IPv4	277
List	Motivations for using private IP networks	278
Figure 11-14	Formats of Class A, B, and C addresses when not subnetted	280
Figure 11-15	Formats of Class A, B, and C addresses when subnetted	280
Figure 11-16	General logic when choosing the size of the subnet and host parts of addresses in a subnet	281
List	Items that together define a subnet	283

## Key Terms You Should Know

subnet, network, classful IP network, variable-length subnet masks (VLSM), network part, subnet part, host part, public IP network, private IP network, subnet mask



# CHAPTER 12

# Analyzing Classful IPv4 Networks

This chapter covers the following exam topics:

## 1.0 Network Fundamentals

### 1.6 Configure and verify IPv4 addressing and subnetting

When operating a network, you often start investigating a problem based on an IP address and mask. Based on the IP address alone, you should be able to determine several facts about the Class A, B, or C network in which the IP address resides.

This chapter lists the key facts about classful IP networks and explains how to discover these facts. Following that, this chapter lists some practice problems. Before moving to the next chapter, you should practice until you can consistently determine all these facts, quickly and confidently, based on an IP address.

## “Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

**Table 12-1** “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Classful Network Concepts	1–5

- Which of the following are not valid Class A network IDs? (Choose two answers.)
  - 1.0.0.0
  - 130.0.0.0
  - 127.0.0.0
  - 9.0.0.0
- Which of the following are not valid Class B network IDs?
  - 130.0.0.0
  - 191.255.0.0
  - 128.0.0.0
  - 150.255.0.0
  - All are valid Class B network IDs.

3. Which of the following are true about IP address 172.16.99.45's IP network? (Choose two answers.)
  - a. The network ID is 172.0.0.0.
  - b. The network is a Class B network.
  - c. The default mask for the network is 255.255.255.0.
  - d. The number of host bits in the unsubnetted network is 16.
4. Which of the following are true about IP address 192.168.6.7's IP network? (Choose two answers.)
  - a. The network ID is 192.168.6.0.
  - b. The network is a Class B network.
  - c. The default mask for the network is 255.255.255.0.
  - d. The number of host bits in the unsubnetted network is 16.
5. Which of the following is a network broadcast address?
  - a. 10.1.255.255
  - b. 192.168.255.1
  - c. 224.1.1.255
  - d. 172.30.255.255

## Foundation Topics

### Classful Network Concepts

Imagine that you have a job interview for your first IT job. As part of the interview, you're given an IPv4 address and mask: 10.4.5.99, 255.255.255.0. What can you tell the interviewer about the classful network (in this case, the Class A network) in which the IP address resides?

This section, the first of two major sections in this chapter, reviews the concepts of *classful IP networks* (in other words, Class A, B, and C networks). In particular, this chapter examines how to begin with a single IP address and then determine the following facts:

- Class (A, B, or C)
- Default mask
- Number of network octets/bits
- Number of host octets/bits
- Number of host addresses in the network
- Network ID
- Network broadcast address
- First and last usable address in the network

IPv4 Network Classes and Related Facts

IP version 4 (IPv4) defines five address classes. Three of the classes, Classes A, B, and C, consist of unicast IP addresses. Unicast addresses identify a single host or interface so that the address uniquely identifies the device. Class D addresses serve as multicast addresses, so that one packet sent to a Class D multicast IPv4 address can actually be delivered to multiple hosts. Finally, Class E addresses were originally intended for experimentation but were changed to simply be reserved for future use. The class can be identified based on the value of the first octet of the address, as shown in Table 12-2.

**Key Topic** **Table 12-2** IPv4 Address Classes Based on First Octet Values

Class	First Octet Values	Purpose
A	1–126	Unicast (large networks)
B	128–191	Unicast (medium-sized networks)
C	192–223	Unicast (small networks)
D	224–239	Multicast
E	240–255	Reserved (formerly experimental)

After you identify the class of a unicast address as either A, B, or C, many other related facts can be derived just through memorization. Table 12-3 lists that information for reference and later study; each of these concepts is described in this chapter.

**Key Topic** **Table 12-3** Key Facts for Classes A, B, and C

	Class A	Class B	Class C
First octet range	1–126	128–191	192–223
Valid network numbers	1.0.0.0–126.0.0.0	128.0.0.0–191.255.0.0	192.0.0.0–223.255.255.0
Total networks	$2^7 - 2 = 126$	$2^{14} = 16,384$	$2^{21} = 2,097,152$
Hosts per network	$2^{24} - 2$	$2^{16} - 2$	$2^8 - 2$
Octets (bits) in network part	1 (8)	2 (16)	3 (24)
Octets (bits) in host part	3 (24)	2 (16)	1 (8)
Default mask	255.0.0.0	255.255.0.0	255.255.255.0

Note that the address ranges of all addresses that begin with 0 and all addresses that begin with 127 are reserved. Had they not been reserved since the creation of Class A networks, as listed in RFC 791 (published in 1981), then they might have been known as class A networks 0.0.0.0 and 127.0.0.0. Because they are reserved, however, the address space has 126 class A networks, and not 128. Also, note that there are no similar reserved ranges to begin/end the class B and C ranges.

Answers to the “Do I Know This Already?” quiz:  
**1 B, C 2 E 3 B, D 4 A, C 5 D**

In addition to the reservation of what would be class A networks 0.0.0.0 and 127.0.0.0 for other purposes, other newer RFCs have also reserved small pieces of the Class A, B, and C address space. So, tables like Table 12-3, with the count of the numbers of Class A, B, and C networks, are a good place to get a sense of the size of the number; however, the number of reserved networks does change slightly over time (albeit slowly) based on these other reserved address ranges.

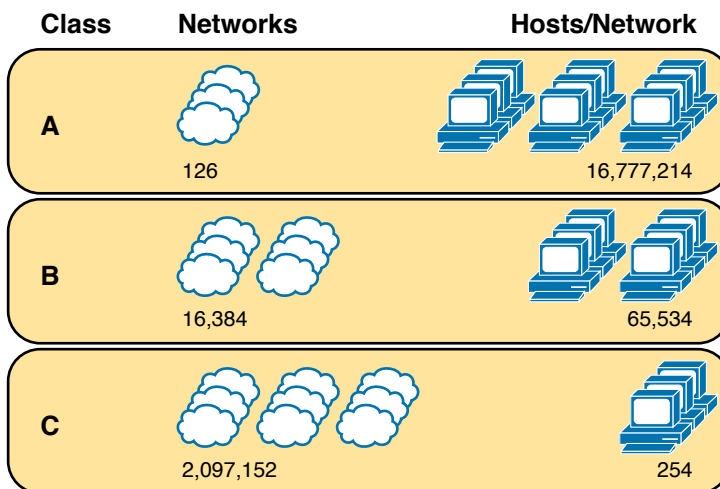
**NOTE** If you are interested in seeing all the reserved IPv4 address ranges, just do an Internet search on “IANA IPv4 special-purpose address registry.”

### The Number and Size of the Class A, B, and C Networks

Table 12-3 lists the range of Class A, B, and C network numbers; however, some key points can be lost just referencing a table of information. This section examines the Class A, B, and C network numbers, focusing on the more important points and the exceptions and unusual cases.

First, the number of networks from each class significantly differs. Only 126 Class A networks exist: network 1.0.0.0, 2.0.0.0, 3.0.0.0, and so on, up through network 126.0.0.0. However, 16,384 Class B networks exist, with more than 2 million Class C networks.

Next, note that the size of networks from each class also significantly differs. Each Class A network is relatively large—over 16 million host IP addresses per network—so they were originally intended to be used by the largest companies and organizations. Class B networks are smaller, with over 65,000 hosts per network. Finally, Class C networks, intended for small organizations, have 254 hosts in each network. Figure 12-1 summarizes those facts.



**Figure 12-1** Numbers and Sizes of Class A, B, and C Networks

### Address Formats

In some cases, an engineer might need to think about a Class A, B, or C network as if the network has not been subdivided through the subnetting process. In such a case, the

addresses in the classful network have a structure with two parts: the *network part* (sometimes called the *prefix*) and the *host part*. Then, comparing any two IP addresses in one network, the following observations can be made:

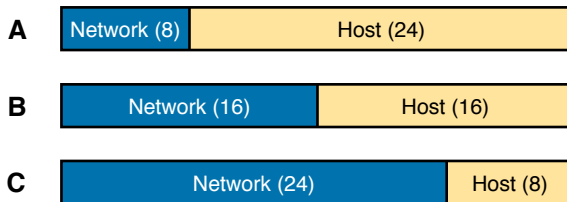
### Key Topic

The addresses in the same network have the same values in the network part.

The addresses in the same network have different values in the host part.

For example, in Class A network 10.0.0.0, by definition, the network part consists of the first octet. As a result, all addresses have an equal value in the network part, namely a 10 in the first octet. If you then compare any two addresses in the network, the addresses have a different value in the last three octets (the host octets). For example, IP addresses 10.1.1.1 and 10.1.1.2 have the same value (10) in the network part, but different values in the host part.

Figure 12-2 shows the format and sizes (in number of bits) of the network and host parts of IP addresses in Class A, B, and C networks, before any subnetting has been applied.



**Figure 12-2** *Sizes (Bits) of the Network and Host Parts of Unsubnetted Classful Networks*

### Default Masks

Although we humans can easily understand the concepts behind Figure 12-2, computers prefer numbers. To communicate those same ideas to computers, each network class has an associated *default mask* that defines the size of the network and host parts of an unsubnetted Class A, B, and C network. To do so, the mask lists binary 1s for the bits considered to be in the network part and binary 0s for the bits considered to be in the host part.

For example, Class A network 10.0.0.0 has a network part of the first single octet (8 bits) and a host part of the last three octets (24 bits). As a result, the Class A default mask is 255.0.0.0, which in binary is

```
11111111 00000000 00000000 00000000
```

Figure 12-3 shows default masks for each network class, both in binary and dotted-decimal format.

**NOTE** Decimal 255 converts to the binary value 11111111. Decimal 0, converted to 8-bit binary, is 00000000. See Appendix A, “Numeric Reference Tables,” for a conversion table.

Key  
Topic

A

Decimal	255	.	0	.	0	.	0
Binary	11111111		00000000		00000000		00000000
Concept	Network (8)		Host (24)				

B

Decimal	255	.	255	.	0	.	0
Binary	11111111		11111111		00000000		00000000
Concept	Network (16)				Host (16)		

C

Decimal	255	.	255	.	255	.	0
Binary	11111111		11111111		11111111		00000000
Concept	Network (24)						Host (8)

Figure 12-3 Default Masks for Classes A, B, and C

## Number of Hosts per Network

Calculating the number of hosts per network requires some basic binary math. First, consider a case where you have a single binary digit. How many unique values are there? There are, of course, two values: 0 and 1. With 2 bits, you can make four combinations: 00, 01, 10, and 11. As it turns out, the total combination of unique values you can make with N bits is  $2^N$ .

Host addresses—the IP addresses assigned to hosts—must be unique. The host bits exist for the purpose of giving each host a unique IP address by virtue of having a different value in the host part of the addresses. So, with H host bits,  $2^H$  unique combinations exist.

However, the number of hosts in a network is not  $2^H$ ; instead, it is  $2^H - 2$ . Each network reserves two numbers that would have otherwise been useful as host addresses but have instead been reserved for special use: one for the network ID and one for the network broadcast address. As a result, the formula to calculate the number of host addresses per Class A, B, or C network is

Key  
Topic

$$2^H - 2$$

where H is the number of host bits.

## Deriving the Network ID and Related Numbers

Each classful network has four key numbers that describe the network. You can derive these four numbers if you start with just one IP address in the network. The numbers are as follows:

- Network number
- First (numerically lowest) usable address
- Last (numerically highest) usable address
- Network broadcast address

First, consider both the network number and first usable IP address. The *network number*, also called the *network ID* or *network address*, identifies the network. By definition, the network number is the numerically lowest number in the network. However, to prevent any ambiguity, the people that made up IP addressing added the restriction that the network number cannot be assigned as an IP address. So, the lowest number in the network is the network ID. Then, the first (numerically lowest) host IP address is *one larger than* the network number.

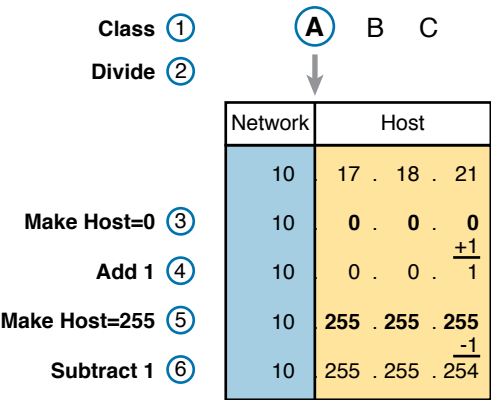
Next, consider the network broadcast address along with the last (numerically highest) usable IP address. The TCP/IP RFCs define a network broadcast address as a special address in each network. This broadcast address could be used as the destination address in a packet, and the routers would forward a copy of that one packet to all hosts in that classful network. Numerically, a network broadcast address is always the highest (last) number in the network. As a result, the highest (last) number usable as an IP address is the address that is *one less than* the network broadcast address.

Simply put, if you can find the network number and network broadcast address, finding the first and last usable IP addresses in the network is easy. For the exam, you should be able to find all four values with ease; the process is as follows:



- Step 1.** Determine the class (A, B, or C) based on the first octet.
- Step 2.** Mentally divide the network and host octets based on the class.
- Step 3.** To find the network number, change the IP address's host octets to 0.
- Step 4.** To find the first address, add 1 to the fourth octet of the network ID.
- Step 5.** To find the broadcast address, change the network ID's host octets to 255.
- Step 6.** To find the last address, subtract 1 from the fourth octet of the network broadcast address.

The written process actually looks harder than it is. Figure 12-4 shows an example of the process, using Class A IP address 10.17.18.21, with the circled numbers matching the process.



**Figure 12-4** Example of Deriving the Network ID and Other Values from 10.17.18.21



Figure 12-4 shows the identification of the class as Class A (Step 1) and the number of network/host octets as 1 and 3, respectively. So, to find the network ID at Step 3, the figure copies only the first octet, setting the last three (host) octets to 0. At Step 4, just copy the network ID and add 1 to the fourth octet. Similarly, to find the broadcast address at Step 5, copy the network octets, but set the host octets to 255. Then, at Step 6, subtract 1 from the fourth octet to find the last (numerically highest) usable IP address.

Just to show an alternative example, consider IP address 172.16.8.9. Figure 12-5 shows the process applied to this IP address.

Class ①	A	<b>B</b>	C
Divide ②		↓	
	Network	Host	
	172 . 16	8 . 9	
Make Host=0 ③	172 . 16	0 . 0	
Add 1 ④	172 . 16	0 . $\begin{smallmatrix} +1 \\ 1 \end{smallmatrix}$	
Make Host=255 ⑤	172 . 16	255 . 255	
Subtract 1 ⑥	172 . 16	255 . $\begin{smallmatrix} -1 \\ 254 \end{smallmatrix}$	

**Figure 12-5** Example Deriving the Network ID and Other Values from 172.16.8.9

Figure 12-5 shows the identification of the class as Class B (Step 1) and the number of network/host octets as 2 and 2, respectively. So, to find the network ID at Step 3, the figure copies only the first two octets, setting the last two (host) octets to 0. Similarly, Step 5 shows the same action, but with the last two (host) octets being set to 255.

## Unusual Network IDs and Network Broadcast Addresses

Some of the more unusual numbers in and around the range of Class A, B, and C network numbers can cause some confusion. This section lists some examples of numbers that make many people make the wrong assumptions about the meaning of the number.

For Class A, the first odd fact is that the range of values in the first octet omits the numbers 0 and 127. As it turns out, what would be Class A network 0.0.0.0 was originally reserved for some broadcasting requirements, so all addresses that begin with 0 in the first octet are reserved. What would be Class A network 127.0.0.0 is still reserved because of a special address used in software testing, called the loopback address (127.0.0.1).

For Class B (and C), some of the network numbers can look odd, particularly if you fall into a habit of thinking that 0s at the end means the number is a network ID, and 255s at the end means it's a network broadcast address. First, Class B network numbers range from 128.0.0.0 to 191.255.0.0, for a total of  $2^{14}$  networks. However, even the very first (lowest number) Class B network number (128.0.0.0) looks a little like a Class A network number because it ends with three 0s. However, the first octet is 128, making it a Class B network with a two-octet network part (128.0).

For another Class B example, the high end of the Class B range also might look strange at first glance (191.255.0.0), but this is indeed the numerically highest of the valid Class B network numbers. This network’s broadcast address, 191.255.255.255, might look a little like a Class A broadcast address because of the three 255s at the end, but it is indeed the broadcast address of a Class B network.

Similarly to Class B networks, some of the valid Class C network numbers do look strange. For example, Class C network 192.0.0.0 looks a little like a Class A network because of the last three octets being 0, but because it is a Class C network, it consists of all addresses that begin with three octets equal to 192.0.0. Similarly, 223.255.255.0, another valid Class C network, consists of all addresses that begin with 223.255.255.

### Practice with Classful Networks

As with all areas of IP addressing and subnetting, you need to practice to be ready for the CCNA exam. You should practice some while reading this chapter to make sure that you understand the processes. At that point, you can use your notes and this book as a reference, with a goal of understanding the process. After that, keep practicing this and all the other subnetting processes. Before you take the exam, you should be able to always get the right answer, and with speed. Table 12-4 summarizes the key concepts and suggestions for this two-phase approach.

**Table 12-4** Keep-Reading and Take-Exam Goals for This Chapter’s Topics

	After Reading This Chapter	Before Taking the Exam
Focus on...	Learning how	Being correct and fast
Tools Allowed	All	Your brain and a notepad
Goal: Accuracy	90% correct	100% correct
Goal: Speed	Any speed	10 seconds

### Practice Deriving Key Facts Based on an IP Address

Practice finding the various facts that can be derived from an IP address, as discussed throughout this chapter. To do so, complete Table 12-5.

**Table 12-5** Practice Problems: Find the Network ID and Network Broadcast

	IP Address	Class	Network Octets	Host Octets	Network ID	Network Broadcast Address
1	1.1.1.1					
2	128.1.6.5					
3	200.1.2.3					
4	192.192.1.1					
5	126.5.4.3					

	IP Address	Class	Network Octets	Host Octets	Network ID	Network Broadcast Address
6	200.1.9.8					
7	192.0.0.1					
8	191.255.1.47					
9	223.223.0.1					

The answers are listed in the section “Answers to Earlier Practice Problems,” later in this chapter.

### Practice Remembering the Details of Address Classes

Tables 12-2 and 12-3, shown earlier in this chapter, summarized some key information about IPv4 address classes. Tables 12-6 and 12-7 show sparse versions of these same tables. To practice recalling those key facts, particularly the range of values in the first octet that identifies the address class, complete these tables. Then, refer to Tables 12-2 and 12-3 to check your answers. Repeat this process until you can recall all the information in the tables.

**Table 12-6** Sparse Study Table Version of Table 12-2

Class	First Octet Values	Purpose
A		
B		
C		
D		
E		

**Table 12-7** Sparse Study Table Version of Table 12-3

	Class A	Class B	Class C
First octet range			
Valid network numbers			
Total networks			
Hosts per network			
Octets (bits) in network part			
Octets (bits) in host part			
Default mask			

## Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter’s material using either the tools in the book or interactive tools for the same material found on the book’s companion website. Refer to the “Your Study Plan” element for more details. Table 12-8 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

**Table 12-8** Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review memory tables		Website
Practice analyzing classful IPv4 networks		Website, Appendix D

## Review All the Key Topics



**Table 12-9** Key Topics for Chapter 12

Key Topic Elements	Description	Page Number
Table 12-2	Address classes	290
Table 12-3	Key facts about Class A, B, and C networks	290
List	Comparisons of network and host parts of addresses in the same classful network	292
Figure 12-3	Default masks	293
Paragraph	Function to calculate the number of hosts per network	294
List	Steps to find information about a classful network	294

## Key Terms You Should Know

network, classful IP network, network number, network ID, network address, network broadcast address, network part, host part, default mask

## Additional Practice for This Chapter's Processes

For additional practice with analyzing classful networks, you may do a set of practice problems using your choice of tools:

**Application:** Use the Analyzing Classful IPv4 Networks application on the companion website.

**PDF:** Alternatively, practice the same problems using companion website Appendix D, “Practice for Chapter 12: Analyzing Classful IPv4 Networks.”

## Answers to Earlier Practice Problems

Table 12-5, shown earlier, listed several practice problems. Table 12-10 lists the answers.

**Table 12-10** Practice Problems: Find the Network ID and Network Broadcast

	IP Address	Class	Network Octets	Host Octets	Network ID	Network Broadcast
1	1.1.1.1	A	1	3	1.0.0.0	1.255.255.255
2	128.1.6.5	B	2	2	128.1.0.0	128.1.255.255
3	200.1.2.3	C	3	1	200.1.2.0	200.1.2.255
4	192.192.1.1	C	3	1	192.192.1.0	192.192.1.255
5	126.5.4.3	A	1	3	126.0.0.0	126.255.255.255
6	200.1.9.8	C	3	1	200.1.9.0	200.1.9.255
7	192.0.0.1	C	3	1	192.0.0.0	192.0.0.255
8	191.255.1.47	B	2	2	191.255.0.0	191.255.255.255
9	223.223.0.1	C	3	1	223.223.0.0	223.223.0.255

The class, number of network octets, and number of host octets all require you to look at the first octet of the IP address to determine the class. If a value is between 1 and 126, inclusive, the address is a Class A address, with one network and three host octets. If a value is between 128 and 191 inclusive, the address is a Class B address, with two network and two host octets. If a value is between 192 and 223, inclusive, it is a Class C address, with three network octets and one host octet.

The last two columns can be found based on Table 12-3, specifically the number of network and host octets along with the IP address. To find the network ID, copy the IP address, but change the host octets to 0. Similarly, to find the network broadcast address, copy the IP address, but change the host octets to 255.

The last three problems can be confusing and were included on purpose so that you could see an example of these unusual cases, as follows.

**Answers to Practice Problem 7 (from Table 12-5)**

Consider IP address 192.0.0.1. First, 192 is on the lower edge of the first octet range for Class C; as such, this address has three network and one host octet. To find the network ID, copy the address, but change the single host octet (the fourth octet) to 0, for a network ID of 192.0.0.0. It looks strange, but it is indeed the network ID.

The network broadcast address choice for problem 7 can also look strange. To find the broadcast address, copy the IP address (192.0.0.1), but change the last octet (the only host octet) to 255, for a broadcast address of 192.0.0.255. In particular, if you decide that the broadcast should be 192.255.255.255, you might have fallen into the trap of logic, like “Change all 0s in the network ID to 255s,” which is not the correct logic. Instead, change all host octets in the IP address (or network ID) to 255s.

**Answers to Practice Problem 8 (from Table 12-5)**

The first octet of problem 8 (191.255.1.47) sits on the upper edge of the Class B range for the first octet (128–191). As such, to find the network ID, change the last two octets (host octets) to 0, for a network ID of 191.255.0.0. This value sometimes gives people problems because they are used to thinking that 255 somehow means the number is a broadcast address.

The broadcast address, found by changing the two host octets to 255, means that the broadcast address is 191.255.255.255. It looks more like a broadcast address for a Class A network, but it is actually the broadcast address for Class B network 191.255.0.0.

**Answers to Practice Problem 9 (from Table 12-5)**

Problem 9, with IP address 223.223.0.1, is near the high end of the Class C range. As a result, only the last (host) octet is changed to 0 to form the network ID 223.223.0.0. It looks a little like a Class B network number at first glance because it ends in two octets of 0. However, it is indeed a Class C network ID (based on the value in the first octet).

*This page intentionally left blank*

# Analyzing Subnet Masks

This chapter covers the following exam topics:

### 1.0 Network Fundamentals

#### 1.6 Configure and verify IPv4 addressing and subnetting

The subnet mask used in one or many subnets in an IP internetwork says a lot about the intent of the subnet design. First, the mask divides addresses into two parts: *prefix* and *host*, with the host part defining the size of the subnet. Then, the class (A, B, or C) further divides the structure of addresses in a subnet, breaking the prefix part into the *network* and *subnet* parts. The subnet part defines the number of subnets that could exist inside one classful IP network, assuming that one mask is used throughout the classful network.

The subnet mask holds the key to understanding several important subnetting design points. However, to analyze a subnet mask, you first need some basic math skills with masks. The math converts masks between the three different formats used to represent a mask:

- Binary
- Dotted-decimal notation (DDN)
- Prefix (also called classless interdomain routing [CIDR])

This chapter has two major sections. The first focuses on the mask formats and the math used to convert between the three formats. The second section explains how to take an IP address and its subnet mask and analyze those values. In particular, it shows how to determine the three-part format of the IPv4 address and describes the facts about the subnetting design that are implied by the mask.

## “Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

**Table 13-1** “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Subnet Mask Conversion	1–3
Defining the Format of IPv4 Addresses	4–7



1. Which of the following answers lists the prefix (CIDR) format equivalent of 255.255.254.0?
  - a. /19
  - b. /20
  - c. /23
  - d. /24
  - e. /25
2. Which of the following answers lists the prefix (CIDR) format equivalent of 255.255.255.240?
  - a. /26
  - b. /28
  - c. /27
  - d. /30
  - e. /29
3. Which of the following answers lists the dotted-decimal notation (DDN) equivalent of /30?
  - a. 255.255.255.192
  - b. 255.255.255.252
  - c. 255.255.255.240
  - d. 255.255.254.0
  - e. 255.255.255.0
4. Working at the help desk, you receive a call and learn a user's PC IP address and mask (10.55.66.77, mask 255.255.255.0). When thinking about this using classful logic, you determine the number of network (N), subnet (S), and host (H) bits. Which of the following is true in this case?
  - a. N=12
  - b. S=12
  - c. H=8
  - d. S=8
  - e. N=24

5. Working at the help desk, you receive a call and learn a user's PC IP address and mask (192.168.9.1/27). When thinking about this using classful logic, you determine the number of network (N), subnet (S), and host (H) bits. Which of the following is true in this case?
  - a. N=24
  - b. S=24
  - c. H=8
  - d. H=7
6. Which of the following statements is true about classless IP addressing concepts?
  - a. Uses a 128-bit IP address
  - b. Applies only for Class A and B networks
  - c. Separates IP addresses into network, subnet, and host parts
  - d. Ignores Class A, B, and C network rules
7. Which of the following masks, when used as the only mask within a Class B network, would supply enough subnet bits to support 100 subnets? (Choose two.)
  - a. /24
  - b. 255.255.255.252
  - c. /20
  - d. 255.255.252.0

## Foundation Topics

### Subnet Mask Conversion

This section describes how to convert between different formats for the subnet mask. You can then use these processes when you practice. If you already know how to convert from one format to the other, go ahead and move to the section “Practice Converting Subnet Masks,” later in this chapter.

#### Three Mask Formats

Subnet masks can be written as 32-bit binary numbers, but not just any binary number. In particular, the binary subnet mask must follow these rules:



- The value must not interleave 1s and 0s.
- If 1s exist, they are on the left.
- If 0s exist, they are on the right.

For example, the following values would be illegal. The first is illegal because the value interleaves 0s and 1s, and the second is illegal because it lists 0s on the left and 1s on the right:

```
10101010 01010101 11110000 00001111
00000000 00000000 00000000 11111111
```

The following two binary values meet the requirements, in that they have all 1s on the left, followed by all 0s, with no interleaving of 1s and 0s:

```
11111111 00000000 00000000 00000000
11111111 11111111 11111111 00000000
```

Two alternative subnet mask formats exist so that we humans do not have to work with 32-bit binary numbers. One format, dotted-decimal notation (DDN), converts each set of 8 bits into the decimal equivalent. For example, the two previous binary masks would convert to the following DDN subnet masks because binary 11111111 converts to decimal 255, and binary 00000000 converts to decimal 0:

255.0.0.0

255.255.255.0

Although the DDN format has been around since the beginning of IPv4 addressing, the third mask format was added later, in the early 1990s: the *prefix* format. This format takes advantage of the rule that the subnet mask starts with some number of 1s, and then the rest of the digits are 0s. Prefix format lists a slash (/) followed by the number of binary 1s in the binary mask. Using the same two examples as earlier in this section, the prefix format equivalent masks are as follows:

/8

/24

Note that although the terms *prefix* or *prefix mask* can be used, the terms *CIDR mask* or *slash mask* can also be used. This newer prefix style mask was created around the same time as the classless interdomain routing (CIDR) specification back in the early 1990s, and the acronym CIDR grew to be used for anything related to CIDR, including prefix-style masks. In addition, the term *slash mask* is sometimes used because the value includes a slash mark (/).

You need to get comfortable working with masks in different formats. The rest of this section examines how to convert between the three formats.

## Converting Between Binary and Prefix Masks

Converting between binary and prefix masks should be relatively intuitive after you know that the prefix value is simply the number of binary 1s in the binary mask. For the sake of completeness, the processes to convert in each direction are



**Binary to prefix:** Count the number of binary 1s in the binary mask, and write the total, in decimal, after a /.

**Prefix to binary:** Write P binary 1s, where P is the prefix value, followed by as many binary 0s as required to create a 32-bit number.

Tables 13-2 and 13-3 show some examples.

**Table 13-2** Example Conversions: Binary to Prefix

Binary Mask	Logic	Prefix Mask
11111111 11111111 11000000 00000000	Count 8 + 8 + 2 = 18 binary 1s	/18
11111111 11111111 11111111 11110000	Count 8 + 8 + 8 + 4 = 28 binary 1s	/28
11111111 11111000 00000000 00000000	Count 8 + 5 = 13 binary 1s	/13

**Table 13-3** Example Conversions: Prefix to Binary

Prefix Mask	Logic	Binary Mask
/18	Write 18 1s, then 14 0s, total 32	11111111 11111111 11000000 00000000
/28	Write 28 1s, then 4 0s, total 32	11111111 11111111 11111111 11110000
/13	Write 13 1s, then 19 0s, total 32	11111111 11111000 00000000 00000000

**Converting Between Binary and DDN Masks**

By definition, a dotted-decimal number (DDN) used with IPv4 addressing contains four decimal numbers, separated by dots. Each decimal number represents 8 bits. So, a single DDN shows four decimal numbers that together represent some 32-bit binary number.

Conversion from a DDN mask to the binary equivalent is relatively simple to describe but can be laborious to perform. First, to do the conversion, the process is as follows:

For each octet, perform a decimal-to-binary conversion.

However, depending on your comfort level with doing decimal-to-binary conversions, that process can be difficult or time-consuming. If you want to think about masks in binary for the exam, consider picking one of the following methods to do the conversion and practicing until you can do it quickly and accurately:

- Do the decimal-binary conversions, but practice your decimal-binary conversions to become faster. If you choose this path, consider the Cisco Binary Game, which you can find by searching its name at the Cisco Learning Network (CLN) (<http://learningnetwork.cisco.com>).
- Use the decimal-binary conversion chart in Appendix A, “Numeric Reference Tables.” This lets you find the answer more quickly now, but you cannot use the chart on exam day.
- Memorize the nine possible decimal values that can be in a decimal mask, and practice using a reference table with those values.

The third method, which is the method recommended in this book, takes advantage of the fact that any and every DDN mask octet must be one of only nine values. Why? Well, remember how a binary mask cannot interleave 1s and 0s, and the 0s must be on the right? It turns out that only nine different 8-bit binary numbers conform to these rules. Table 13-4 lists the values, along with other relevant information.

Answers to the “Do I Know This Already?” quiz:

1 C 2 B 3 B 4 C 5 A 6 D 7 A, B

# Key Topic

**Table 13-4** Nine Possible Values in One Octet of a Subnet Mask

Binary Mask Octet	Decimal Equivalent	Number of Binary 1s
00000000	0	0
10000000	128	1
11000000	192	2
11100000	224	3
11110000	240	4
11111000	248	5
11111100	252	6
11111110	254	7
11111111	255	8

Many subnetting processes can be done with or without binary math. Some of those processes—mask conversion included—use the information in Table 13-4. You should plan to memorize the information in the table. I recommend making a copy of the table to keep handy while you practice. (You will likely memorize the contents of this table simply by practicing the conversion process enough to get both good and fast at the conversion.)

Using the table, the conversion processes in each direction with binary and decimal masks are as follows:

# Key Topic

**Binary to decimal:** Organize the bits into four sets of eight. For each octet, find the binary value in the table and write down the corresponding decimal value.

**Decimal to binary:** For each octet, find the decimal value in the table and write down the corresponding 8-bit binary value.

Tables 13-5 and 13-6 show some examples.

**Table 13-5** Conversion Example: Binary to Decimal

Binary Mask	Logic	Decimal Mask
11111111 11111111 11000000 00000000	11111111 maps to 255 11000000 maps to 192 00000000 maps to 0	255.255.192.0
11111111 11111111 11111111 11110000	11111111 maps to 255 11110000 maps to 240	255.255.255.240
11111111 11111000 00000000 00000000	11111111 maps to 255 11111000 maps to 248 00000000 maps to 0	255.248.0.0

**Table 13-6** Conversion Examples: Decimal to Binary

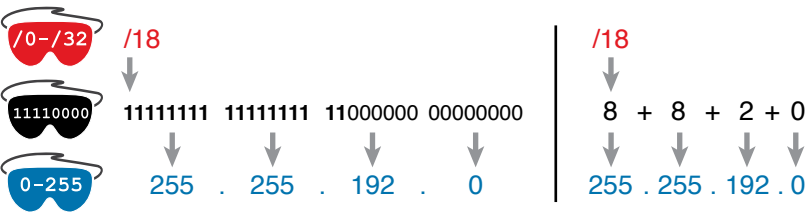
Decimal Mask	Logic	Binary Mask
255.255.192.0	255 maps to 11111111 192 maps to 11000000 0 maps to 00000000	11111111 11111111 11000000 00000000
255.255.255.240	255 maps to 11111111 240 maps to 11111000	11111111 11111111 11111111 11110000
255.248.0.0	255 maps to 11111111 248 maps to 11111000 0 maps to 00000000	11111111 11111000 00000000 00000000

**Converting Between Prefix and DDN Masks**

When you are learning, the best way to convert between the prefix and decimal formats is to first convert to binary. For example, to move from decimal to prefix, first convert decimal to binary and then from binary to prefix.

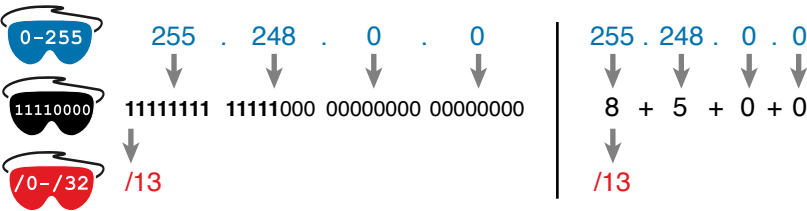
For the exams, set a goal to master these conversions doing the math in your head. While learning, you will likely want to use paper. To train yourself to do all this without writing it down, instead of writing each octet of binary, just write the number of binary 1s in that octet.

Figure 13-1 shows an example with a prefix-to-decimal conversion. The left side shows the conversion to binary as an interim step. For comparison, the right side shows the binary interim step in shorthand that just lists the number of binary 1s in each octet of the binary mask.



**Figure 13-1** Conversion from Prefix to Decimal: Full Binary Versus Shorthand

Similarly, when converting from decimal to prefix, mentally convert to binary along the way, and as you improve, just think of the binary as the number of 1s in each octet. Figure 13-2 shows an example of such a conversion.



**Figure 13-2** Conversion from Decimal to Prefix: Full Binary Versus Shorthand

Note that Appendix A has a table that lists all 33 legal subnet masks, with all three formats shown.

### Practice Converting Subnet Masks

Before moving to the second half of this chapter, and thinking about what these subnet masks mean, first do some practice. Practice the processes discussed in this chapter until you get the right answer most of the time. Later, before taking the exam, practice more until you master the topics in this chapter and can move pretty fast, as outlined in the right column of Table 13-7.

**Table 13-7** Keep-Reading and Take-Exam Goals for This Chapter’s Topics

	Before Moving to the Next Section	Before Taking the Exam
Focus On...	Learning how	Being correct and fast
Tools Allowed	All	Your brain and a notepad
Goal: Accuracy	90% correct	100% correct
Goal: Speed	Any speed	10 seconds

Table 13-8 lists eight practice problems. The table has three columns, one for each mask format. Each row lists one mask, in one format. Your job is to find the mask’s value in the other two formats for each row. Table 13-12, located in the section “Answers to Earlier Practice Problems,” later in this chapter, lists the answers.

**Table 13-8** Practice Problems: Find the Mask Values in the Other Two Formats

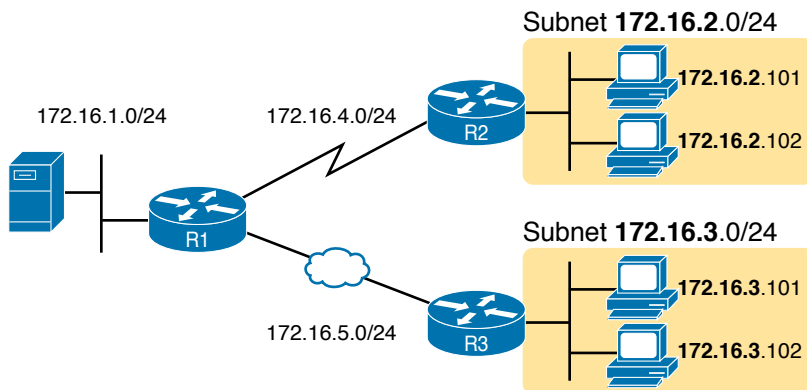
Prefix	Binary Mask	Decimal
	11111111 11111111 11000000 00000000	
		255.255.255.252
/25		
/16		
		255.0.0.0
	11111111 11111111 11111100 00000000	
		255.254.0.0
/27		

### Identifying Subnet Design Choices Using Masks

Subnet masks have many purposes. In fact, if ten experienced network engineers were independently asked, “What is the purpose of a subnet mask?” the engineers would likely give a variety of true answers. The subnet mask plays several roles.

This chapter focuses on one particular use of a subnet mask: defining the prefix part of the IP addresses in a subnet. The prefix part must be the same value for all addresses in a subnet. In fact, a single subnet can be defined as all IPv4 addresses that have the same value in the prefix part of their IPv4 addresses.

While the previous paragraph might sound a bit formal, the idea is relatively basic, as shown in Figure 13-3. The figure shows a network diagram, focusing on two subnets: a subnet of all addresses that begin with 172.16.2 and another subnet made of all addresses that begin with 172.16.3. In this example, the prefix—the part that has the same value in all the addresses in the subnet—is the first three octets.



**Figure 13-3** *Simple Subnet Design, with Mask /24*

While people can sit around a conference table and talk about how a prefix is three octets long, computers communicate that same concept using a subnet mask. In this case, the subnets use a subnet mask of /24, which means that the prefix part of the addresses is 24 bits (3 octets) long.

This section explains more about how to use a subnet mask to understand this concept of a prefix part of an IPv4 address, along with these other uses for a subnet mask. Note that this section discusses the first five items in the list.

**Key Topic**

- Defines the size of the prefix (combined network and subnet) part of the addresses in a subnet
- Defines the size of the host part of the addresses in the subnet
- Can be used to calculate the number of hosts in the subnet
- Provides a means for the network designer to communicate the design details—the number of subnet and host bits—to the devices in the network
- Under certain assumptions, can be used to calculate the number of subnets in the entire classful network
- Can be used in binary calculations of both the subnet ID and the subnet broadcast address



# Masks Divide the Subnet's Addresses into Two Parts

The subnet mask subdivides the IP addresses in a subnet into two parts: the *prefix*, or *subnet part*, and the *host part*.

The prefix part identifies the addresses that reside in the same subnet because all IP addresses in the same subnet have the same value in the prefix part of their addresses. The idea is much like the postal code (ZIP codes in the United States) in mailing addresses. All mailing addresses in the same town have the same postal code. Likewise, all IP addresses in the same subnet have identical values in the prefix part of their addresses.

The host part of an address identifies the host uniquely inside the subnet. If you compare any two IP addresses in the same subnet, their host parts will differ, even though the prefix parts of their addresses have the same value. To summarize these key comparisons:

## Key Topic

**Prefix (subnet) part:** Equal in all addresses in the same subnet.

**Host part:** Different in all addresses in the same subnet.

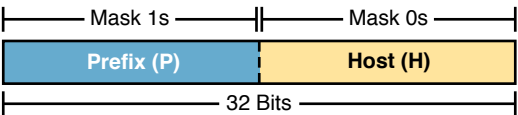
For example, imagine a subnet that, in concept, includes all addresses whose first three octets are 10.1.1. So, the following list shows several addresses in this subnet:

10.1.1.1  
10.1.1.2  
10.1.1.3

In this list, the prefix or subnet part (the first three octets of 10.1.1) are equal. The host part (the last octet [in bold]) is different. So, the prefix or subnet part of the address identifies the group, and the host part identifies the specific member of the group.

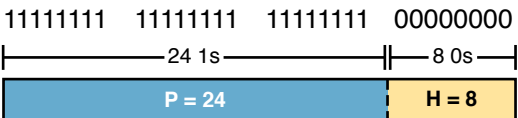
The subnet mask defines the dividing line between the prefix and the host part. To do so, the mask creates a conceptual line between the binary 1s in the binary mask and the binary 0s in the mask. In short, if a mask has P binary 1s, the prefix part is P bits long and the rest of the bits are host bits. Figure 13-4 shows the general concept.

## Key Topic



**Figure 13-4** Prefix (Subnet) and Host Parts Defined by Mask 1s and 0s

The next figure, Figure 13-5, shows a specific example using mask 255.255.255.0. Mask 255.255.255.0 (/24) has 24 binary 1s, for a prefix length of 24 bits.



**Figure 13-5** Mask 255.255.255.0: P=24, H=8

### Masks and Class Divide Addresses into Three Parts

In addition to the two-part view of IPv4 addresses, you can also think about IPv4 addresses as having three parts. To do so, just apply Class A, B, and C rules to the address format to define the network part at the beginning of the address. This added logic divides the prefix into two parts: the *network* part and the *subnet* part. The class defines the length of the network part, with the subnet part simply being the rest of the prefix. Figure 13-6 shows the idea.

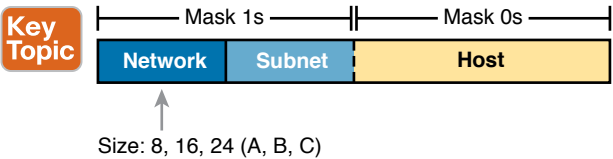


Figure 13-6 Class Concepts Applied to Create Three Parts

The combined network and subnet parts act like the prefix because all addresses in the same subnet must have identical values in the network and subnet parts. The size of the host part remains unchanged, whether viewing the addresses as having two parts or three parts.

To be complete, Figure 13-7 shows the same example as in the previous section, with the subnet of “all addresses that begin with 10.1.1.” In that example, the subnet uses mask 255.255.255.0, and the addresses are all in Class A network 10.0.0.0. The class defines 8 network bits, and the mask defines 24 prefix bits, meaning that  $24 - 8 = 16$  subnet bits exist. The host part remains as 8 bits per the mask.

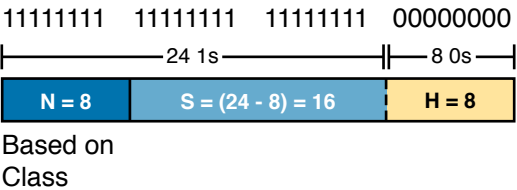


Figure 13-7 Subnet 10.1.1.0, Mask 255.255.255.0: N=8, S=16, H=8

### Classless and Classful Addressing

The terms *classless addressing* and *classful addressing* refer to the two different ways to think about IPv4 addresses as described so far in this chapter. Classful addressing means that you think about Class A, B, and C rules, so the prefix is separated into the network and subnet parts, as shown in Figures 13-6 and 13-7. Classless addressing means that you ignore the Class A, B, and C rules and treat the prefix part as one part, as shown in Figures 13-4 and 13-5. The following more formal definitions are listed for reference and study:

- Key Topic

**Classless addressing:** The concept that an IPv4 address has two parts—the prefix part plus the host part—as defined by the mask, with *no consideration of the class* (A, B, or C).

**Classful addressing:** The concept that an IPv4 address has three parts—network, subnet, and host—as defined by the mask *and Class A, B, and C rules*.

**NOTE** Unfortunately, the networking world uses the terms *classless* and *classful* in a couple of different ways. In addition to the classless and classful addressing described here, each routing protocol can be categorized as either a *classless routing protocol* or a *classful routing protocol*. In another use, the terms *classless routing* and *classful routing* refer to some details of how Cisco routers forward (route) packets using the default route in some cases. As a result, these terms can be easily confused and misused. So, when you see the words *classless* and *classful*, be careful to note the context: addressing, routing, or routing protocols.

## Calculations Based on the IPv4 Address Format

After you know how to break an address down using both classless and classful addressing rules, you can easily calculate a couple of important facts using some basic math formulas.

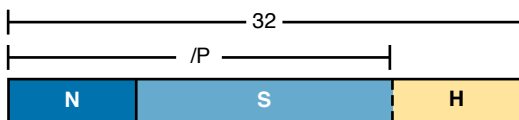
First, for any subnet, after you know the number of host bits, you can calculate the number of host IP addresses in the subnet. Next, if you know the number of subnet bits (using classful addressing concepts) and you know that only one subnet mask is used throughout the network, you can also calculate the number of subnets in the network. The formulas just require that you know the powers of 2:

**Hosts in the subnet:**  $2^H - 2$ , where H is the number of host bits.

**Subnets in the network:**  $2^S$ , where S is the number of subnet bits. Only use this formula if only one mask is used throughout the network.

**NOTE** The section “Choose the Mask” in Chapter 11, “Perspectives on IPv4 Subnetting,” details many concepts related to masks, including comments about this assumption of one mask throughout a single Class A, B, or C network.

The sizes of the parts of IPv4 addresses can also be calculated. The math is basic, but the concepts are important. Keeping in mind that IPv4 addresses are 32 bits long, the two parts with classless addressing must add up to 32 ( $P + H = 32$ ), and with classful addressing, the three parts must add up to 32 ( $N + S + H = 32$ ). Figure 13-8 shows the relationships.



Class:

A: N = 8

B: N = 16

C: N = 24

**Figure 13-8** Relationship Between /P, N, S, and H

You often begin with an IP address and mask, both when answering questions on the CCNA exam and when examining problems that occur in real networks. Based on the information in this chapter and earlier chapters, you should be able to find all the information in Figure 13-8 and then calculate the number of hosts/subnet and the number of subnets in the network.

For reference, the following process spells out the steps:

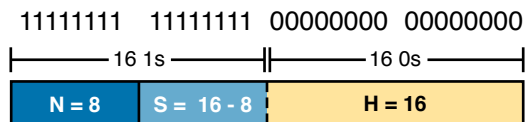


- Step 1.** Convert the mask to prefix format (/P) as needed. (See the earlier section “Practice Converting Subnet Masks” for review.)
- Step 2.** Determine N based on the class. (See Chapter 12, “Analyzing Classful IPv4 Networks,” for review.)
- Step 3.** Calculate  $S = P - N$ .
- Step 4.** Calculate  $H = 32 - P$ .
- Step 5.** Calculate hosts/subnet:  $2^H - 2$ .
- Step 6.** Calculate number of subnets:  $2^S$ .

For example, consider the case of IP address 8.1.4.5 with mask 255.255.0.0 by following this process:

- Step 1.** 255.255.0.0 = /16, so  $P=16$ .
- Step 2.** 8.1.4.5 is in the range 1–126 in the first octet, so it is Class A; so  $N=8$ .
- Step 3.**  $S = P - N = 16 - 8 = 8$ .
- Step 4.**  $H = 32 - P = 32 - 16 = 16$ .
- Step 5.**  $2^{16} - 2 = 65,534$  hosts/subnet.
- Step 6.**  $2^8 = 256$  subnets.

Figure 13-9 shows a visual analysis of the same problem.



**Figure 13-9** Visual Representation of Problem: 8.1.4.5, 255.255.0.0

For another example, consider address 200.1.1.1, mask 255.255.255.252 by following this process:

- Step 1.** 255.255.255.252 = /30, so  $P=30$ .
- Step 2.** 200.1.1.1 is in the range 192–223 in the first octet, so it is Class C; so  $N=24$ .
- Step 3.**  $S = P - N = 30 - 24 = 6$ .
- Step 4.**  $H = 32 - P = 32 - 30 = 2$ .
- Step 5.**  $2^2 - 2 = 2$  hosts/subnet.
- Step 6.**  $2^6 = 64$  subnets.

This example uses a popular mask for serial links because serial links only require two host addresses, and the mask supports only two host addresses.

### Practice Analyzing Subnet Masks

As with the other subnetting math in this book, using a two-phase approach may help. Take time now to practice until you feel as though you understand the process. Then, before the exam, make sure you master the math. Table 13-9 summarizes the key concepts and suggestions for this two-phase approach.

**Table 13-9** Keep-Reading and Take-Exam Goals for This Chapter’s Topics

	Before Moving to the Next Chapter	Before Taking the Exam
Focus On...	Learning how	Being correct and fast
Tools Allowed	All	Your brain and a notepad
Goal: Accuracy	90% correct	100% correct
Goal: Speed	Any speed	15 seconds

On a piece of scratch paper, answer the following questions. In each case:

- Determine the structure of the addresses in each subnet based on the class and mask, using classful IP addressing concepts. In other words, find the size of the network, subnet, and host parts of the addresses.
  - Calculate the number of hosts in the subnet.
  - Calculate the number of subnets in the network, assuming that the same mask is used throughout.
1. 8.1.4.5, 255.255.254.0
  2. 130.4.102.1, 255.255.255.0
  3. 199.1.1.100, 255.255.255.0
  4. 130.4.102.1, 255.255.252.0
  5. 199.1.1.100, 255.255.255.224

The answers are listed in the section “Answers to Earlier Practice Problems,” later in this chapter.

## Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter’s material using either the tools in the book or interactive tools for the same material found on the book’s companion website. Refer to the “Your Study Plan” element for more details. Table 13-10 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

**Table 13-10** Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review memory tables		Website
Practice analyzing subnet masks		Website, Appendix E

## Review All the Key Topics

### Key Topic

**Table 13-11** Key Topics for Chapter 13

Key Topic Element	Description	Page Number
List	Rules for binary subnet mask values	304
List	Rules to convert between binary and prefix masks	305
Table 13-4	Nine possible values in a decimal subnet mask	307
List	Rules to convert between binary and DDN masks	307
List	Some functions of a subnet mask	310
List	Comparisons of IP addresses in the same subnet	311
Figure 13-4	Two-part classless view of an IP address	311
Figure 13-6	Three-part classful view of an IP address	312
List	Definitions of classful addressing and classless addressing	312
List	Formal steps to analyze masks and calculate values	314

## Key Terms You Should Know

binary mask, dotted-decimal notation (DDN), decimal mask, prefix mask, CIDR mask, classful addressing, classless addressing

## Additional Practice for This Chapter's Processes

You can do more practice with the processes in this chapter with a pair of practice sets. One focuses on interpreting existing masks, while the other gives you practice with converting between mask formats. You may do each practice set using the following tools:

**Application:** Use the “Analyzing Subnet Masks” and “Converting Masks” applications on the companion website, listed under the Chapter Review for this chapter.

**PDF:** Alternatively, practice the same problems found in both these apps using companion website Appendix E, “Practice for Chapter 13: Analyzing Subnet Masks.”

## Answers to Earlier Practice Problems

Table 13-8, shown earlier, listed several practice problems for converting subnet masks; Table 13-12 lists the answers.

**Table 13-12** Answers to Problems in Table 13-8

Prefix	Binary Mask	Decimal
/18	11111111 11111111 11000000 00000000	255.255.192.0
/30	11111111 11111111 11111111 11111100	255.255.255.252
/25	11111111 11111111 11111111 10000000	255.255.255.128
/16	11111111 11111111 00000000 00000000	255.255.0.0
/8	11111111 00000000 00000000 00000000	255.0.0.0
/22	11111111 11111111 11111100 00000000	255.255.252.0
/15	11111111 11111110 00000000 00000000	255.254.0.0
/27	11111111 11111111 11111111 11100000	255.255.255.224

Table 13-13 lists the answers to the practice problems from the earlier section “Practice Analyzing Subnet Masks.”

**Table 13-13** Answers to Problems from Earlier in the Chapter

	Problem	/P	Class	N	S	H	2 <sup>S</sup>	2 <sup>H</sup> – 2
1	8.1.4.5 255.255.254.0	23	A	8	15	9	32,768	510
2	130.4.102.1 255.255.255.0	24	B	16	8	8	256	254
3	199.1.1.100 255.255.255.0	24	C	24	0	8	N/A	254
4	130.4.102.1 255.255.252.0	22	B	16	6	10	64	1022
5	199.1.1.100 255.255.255.224	27	C	24	3	5	8	30

The following list reviews the problems:

- For 8.1.4.5, the first octet (8) is in the 1–126 range, so it is a Class A address, with 8 network bits. Mask 255.255.254.0 converts to /23, so  $P - N = 15$ , for 15 subnet bits. H can be found by subtracting /P (23) from 32, for 9 host bits.
- 130.4.102.1 is in the 128–191 range in the first octet, making it a Class B address, with  $N = 16$  bits. 255.255.255.0 converts to /24, so the number of subnet bits is  $24 - 16 = 8$ . With 24 prefix bits, the number of host bits is  $32 - 24 = 8$ .
- The third problem purposely shows a case where the mask does not create a subnet part of the address. The address, 199.1.1.100, has a first octet between 192 and 223, making it a Class C address with 24 network bits. The prefix version of the mask is /24, so the number of subnet bits is  $24 - 24 = 0$ . The number of host bits is 32 minus

the prefix length (24), for a total of 8 host bits. So in this case, the mask shows that the network engineer is using the default mask, which creates no subnet bits and no subnets.

4. With the same address as the second problem, 130.4.102.1 is a Class B address with  $N = 16$  bits. This problem uses a different mask, 255.255.252.0, which converts to /22. This makes the number of subnet bits  $22 - 16 = 6$ . With 22 prefix bits, the number of host bits is  $32 - 22 = 10$ .
5. With the same address as the third problem, 199.1.1.100 is a Class C address with  $N = 24$  bits. This problem uses a different mask, 255.255.255.224, which converts to /27. This makes the number of subnet bits  $27 - 24 = 3$ . With 27 prefix bits, the number of host bits is  $32 - 27 = 5$ .



*This page intentionally left blank*

# Analyzing Existing Subnets

This chapter covers the following exam topics:

## 1.0 Network Fundamentals

### 1.6 Configure and verify IPv4 addressing and subnetting

Often, a networking task begins with the discovery of the IP address and mask used by some host. Then, to understand how the internetwork routes packets to that host, you must find key pieces of information about the subnet, specifically the following:

- Subnet ID
- Subnet broadcast address
- Subnet's range of usable unicast IP addresses

This chapter discusses the concepts and math to take a known IP address and mask, and then fully describe a subnet by finding the values in this list. These specific tasks might well be the most important IP skills in the entire IP addressing and subnetting topics in this book because these tasks might be the most commonly used tasks when operating and troubleshooting real networks.

## “Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

**Table 14-1** “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Defining a Subnet	1
Analyzing Existing Subnets: Binary	2
Analyzing Existing Subnets: Decimal	3–6

1. When you think about an IP address using classful addressing rules, an address can have three parts: network, subnet, and host. If you examined all the addresses in one subnet, in binary, which of the following answers correctly states which of the three parts of the addresses will be equal among all addresses? (Choose the best answer.)
  - a. Network part only
  - b. Subnet part only
  - c. Host part only
  - d. Network and subnet parts
  - e. Subnet and host parts

2. Which of the following statements are true regarding the binary subnet ID, subnet broadcast address, and host IP address values in any single subnet? (Choose two answers.)
- a. The host part of the broadcast address is all binary 0s.
  - b. The host part of the subnet ID is all binary 0s.
  - c. The host part of a usable IP address can have all binary 1s.
  - d. The host part of any usable IP address must not be all binary 0s.
3. Which of the following is the resident subnet ID for IP address 10.799.133/24?
- a. 10.0.0.0
  - b. 10.70.0
  - c. 10.799.0
  - d. 10.799.128
4. Which of the following is the resident subnet for IP address 192.168.44.97/30?
- a. 192.168.44.0
  - b. 192.168.44.64
  - c. 192.168.44.96
  - d. 192.168.44.128
5. Which of the following is the subnet broadcast address for the subnet in which IP address 172.31.77.201/27 resides?
- a. 172.31.201.255
  - b. 172.31.255.255
  - c. 172.31.77.223
  - d. 172.31.77.207
6. A fellow engineer tells you to configure the DHCP server to lease the last 100 usable IP addresses in subnet 10.1.4.0/23. Which of the following IP addresses could be leased as a result of your new configuration?
- a. 10.1.4.156
  - b. 10.1.4.254
  - c. 10.1.5.200
  - d. 10.1.7.200
  - e. 10.1.255.200

## Foundation Topics

### Defining a Subnet

An IP subnet is a subset of a classful network, created by choice of some network engineer. However, that engineer cannot pick just any arbitrary subset of addresses; instead, the engineer must follow certain rules, such as the following:

#### Key Topic

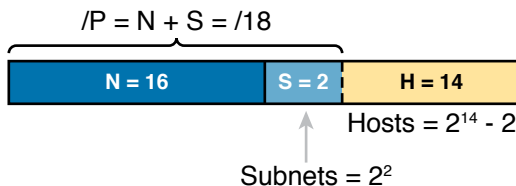
- The subnet contains a set of consecutive numbers.
- The subnet holds  $2^H$  numbers, where H is the number of host bits defined by the subnet mask.
- Two special numbers in the range cannot be used as IP addresses:
  - The first (lowest) number acts as an identifier for the subnet (*subnet ID*).
  - The last (highest) number acts as a *subnet broadcast address*.
- The remaining addresses, whose values sit between the subnet ID and subnet broadcast address, are used as *unicast IP addresses*.

This section reviews and expands the basic concepts of the subnet ID, subnet broadcast address, and range of addresses in a subnet.

### An Example with Network 172.16.0.0 and Four Subnets

Imagine that you work at the customer support center, where you receive all initial calls from users who have problems with their computer. You coach the user through finding her IP address and mask: 172.16.150.41, mask 255.255.192.0. One of the first and most common tasks you will do based on that information is to find the subnet ID of the subnet in which that address resides. (In fact, this subnet ID is sometimes called the *resident subnet* because the IP address exists in or resides in that subnet.)

Before getting into the math, examine the mask (255.255.192.0) and classful network (172.16.0.0) for a moment. From the mask, based on what you learned in Chapter 13, “Analyzing Subnet Masks,” you can find the structure of the addresses in the subnet, including the number of host and subnet bits. That analysis tells you that two subnet bits exist, meaning that there should be four ( $2^2$ ) subnets. Figure 14-1 shows the idea.



**Figure 14-1** Address Structure: Class B Network, /18 Mask

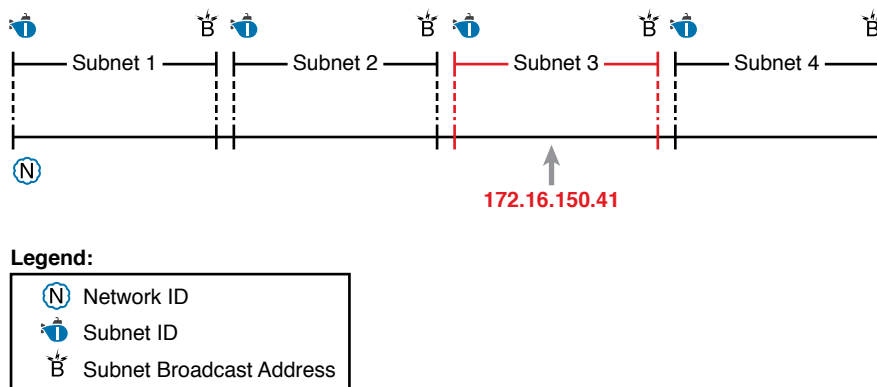
Answers to the “Do I Know This Already?” quiz:

1 D 2 B, D 3 C 4 C 5 C 6 C

**NOTE** This chapter, like the others in this part of the book, assumes that one mask is used throughout an entire classful network.

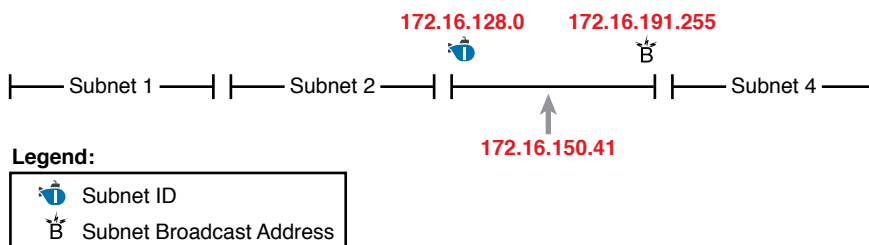
Because each subnet uses a single mask, all subnets of this single IP network must be the same size, because all subnets have the same structure. In this example, all four subnets will have the structure shown in the figure, so all four subnets will have  $2^{14} - 2$  host addresses.

Next, consider the big picture of what happens with this example subnet design: the one Class B network now has four subnets of equal size. Conceptually, if you represent the entire Class B network as a number line, each subnet consumes one-fourth of the number line, as shown in Figure 14-2. Each subnet has a subnet ID—the numerically lowest number in the subnet—so it sits on the left of the subnet. And each subnet has a subnet broadcast address—the numerically highest number in the subnet—so it sits on the right side of the subnet.



**Figure 14-2** Network 172.16.0.0, Divided into Four Equal Subnets

The rest of this chapter focuses on how to take one IP address and mask and discover the details about that one subnet in which the address resides. In other words, you see how to find the resident subnet of an IP address. Again, using IP address 172.16.150.41 and mask 255.255.192.0 as an example, Figure 14-3 shows the resident subnet, along with the subnet ID and subnet broadcast address that bracket the subnet.



**Figure 14-3** Resident Subnet for 172.16.150.41, 255.255.192.0

## Subnet ID Concepts

A subnet ID is simply a number used to succinctly represent a subnet. When listed along with its matching subnet mask, the subnet ID identifies the subnet and can be used to derive the subnet broadcast address and range of addresses in the subnet. Rather than having to write down all these details about a subnet, you simply need to write down the subnet ID and mask, and you have enough information to fully describe the subnet.

The subnet ID appears in many places, but it is seen most often in IP routing tables. For example, when an engineer configures a router with its IP address and mask, the router calculates the subnet ID and puts a route into its routing table for that subnet. The router typically then advertises the subnet ID/mask combination to neighboring routers with some IP routing protocol. Eventually, all the routers in an enterprise learn about the subnet—again using the subnet ID and subnet mask combination—and display it in their routing tables. (You can display the contents of a router’s IP routing table using the **show ip route** command.)

Unfortunately, the terminology related to subnets can sometimes cause problems. First, the terms *subnet ID*, *subnet number*, and *subnet address* are synonyms. In addition, people sometimes simply say *subnet* when referring to both the idea of a subnet and the number that is used as the subnet ID. When talking about routing, people sometimes use the term *prefix* instead of *subnet*. The term *prefix* refers to the same idea as *subnet*; it just uses terminology from the classless addressing way to describe IP addresses, as discussed in Chapter 13’s section “Classless and Classful Addressing.”

The biggest terminology confusion arises between the terms *network* and *subnet*. In the real world, people often use these terms synonymously, and that is perfectly reasonable in some cases. In other cases, the specific meaning of these terms, and their differences, matter to what is being discussed.

For example, people often might say, “What is the network ID?” when they really want to know the subnet ID. In another case, they might want to know the Class A, B, or C network ID. So, when one engineer asks something like, “What’s the net ID for 172.16.150.41 slash 18?” use the context to figure out whether he wants the literal classful network ID (172.16.0.0, in this case) or the literal subnet ID (172.16.128.0, in this case).

For the exams, be ready to notice when the terms *subnet* and *network* are used, and then use the context to figure out the specific meaning of the term in that case.

Table 14-2 summarizes the key facts about the subnet ID, along with the possible synonyms, for easier review and study.



**Table 14-2** Summary of Subnet ID Key Facts

Definition	Number that represents the subnet
Numeric Value	First (smallest) number in the subnet
Literal Synonyms	Subnet number, subnet address, prefix, resident subnet
Common-Use Synonyms	Network, network ID, network number, network address
Typically Seen In...	Routing tables, documentation

## Subnet Broadcast Address

The subnet broadcast address has two main roles: to be used as a destination IP address for the purpose of sending packets to all hosts in the subnet, and as a means to find the high end of the range of addresses in a subnet.

The original purpose for the subnet broadcast address was to give hosts a way to send one packet to all hosts in a subnet and to do so efficiently. For example, a host in subnet A could send a packet with a destination address of subnet B's subnet broadcast address. The routers would forward this one packet just like a packet sent to a host in subnet B. After the packet arrives at the router connected to subnet B, that last router would then forward the packet to all hosts in subnet B, typically by encapsulating the packet in a data-link layer broadcast frame. As a result, all hosts in host B's subnet would receive a copy of the packet.

The subnet broadcast address also helps you find the range of addresses in a subnet because the broadcast address is the last (highest) number in a subnet's range of addresses. To find the low end of the range, calculate the subnet ID; to find the high end of the range, calculate the subnet broadcast address.

Table 14-3 summarizes the key facts about the subnet broadcast address, along with the possible synonyms, for easier review and study.



**Table 14-3 Summary of Subnet Broadcast Address Key Facts**

<b>Definition</b>	A reserved number in each subnet that, when used as the destination address of a packet, causes the device to forward the packet to all hosts in that subnet
<b>Numeric Value</b>	Last (highest) number in the subnet
<b>Literal Synonyms</b>	Directed broadcast address
<b>Broader-Use Synonyms</b>	Network broadcast
<b>Typically Seen In...</b>	In calculations of the range of addresses in a subnet

## Range of Usable Addresses

The engineers implementing an IP internetwork need to know the range of unicast IP addresses in each subnet. Before you can plan which addresses to use as statically assigned IP addresses, which to configure to be leased by the DHCP server, and which to reserve for later use, you need to know the range of usable addresses.

To find the range of usable IP addresses in a subnet, first find the subnet ID and the subnet broadcast address. Then, just add 1 to the fourth octet of the subnet ID to get the first (lowest) usable address, and subtract 1 from the fourth octet of the subnet broadcast address to get the last (highest) usable address in the subnet.

For example, Figure 14-3 showed subnet ID 172.16.128.0, mask /18. The first usable address is simply one more than the subnet ID (in this case, 172.16.128.1). That same figure showed a subnet broadcast address of 172.16.191.255, so the last usable address is one less, or 172.16.191.254.

Now that this section has described the concepts behind the numbers that collectively define a subnet, the rest of this chapter focuses on the math used to find these values.

## Analyzing Existing Subnets: Binary

What does it mean to “analyze a subnet”? For this book, it means that you should be able to start with an IP address and mask and then define key facts about the subnet in which that address resides. Specifically, that means discovering the subnet ID, subnet broadcast address, and range of addresses. The analysis can also include the calculation of the number of addresses in the subnet as discussed in Chapter 13, but this chapter does not review those concepts.

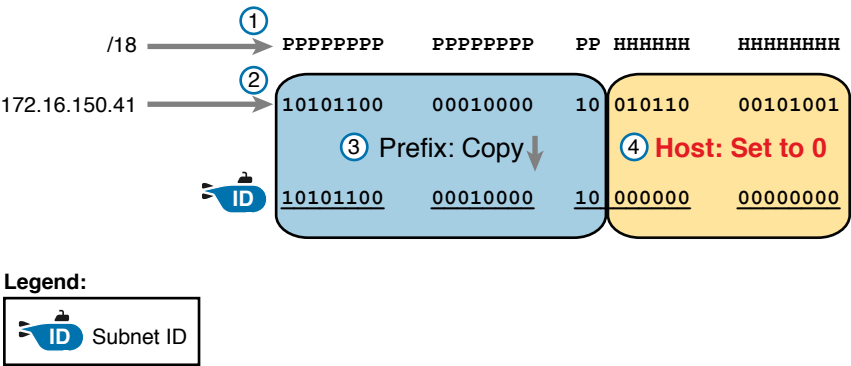
Many methods exist to calculate the details about a subnet based on the address/mask. This section begins by discussing some calculations that use binary math, with the next section showing alternatives that use only decimal math. Although many people prefer the decimal method for going fast on the exams, the binary calculations ultimately give you a better understanding of IPv4 addressing. In particular, if you plan to move on to attain Cisco certifications beyond CCNA, you should take the time to understand the binary methods discussed in this section, even if you use the decimal methods for the exams.

### Finding the Subnet ID: Binary

The two following statements summarize the logic behind the binary value of any subnet ID:

All numbers in the subnet (subnet ID, subnet broadcast address, and all usable IP addresses) have the same value in the prefix part of the numbers.

The subnet ID is the lowest numeric value in the subnet, so its host part, in binary, is all 0s. To find the subnet ID in binary, you take the IP address in binary and change all host bits to binary 0. To do so, you need to convert the IP address to binary. You also need to identify the prefix and host bits, which can be easily done by converting the mask (as needed) to prefix format. (Note that Appendix A, “Numeric Reference Tables,” includes a decimal-binary conversion table.) Figure 14-4 shows the idea, using the same address/mask as in the earlier examples in this chapter: 172.16.150.41, mask /18.



**Figure 14-4** Binary Concept: Convert the IP Address to the Subnet ID

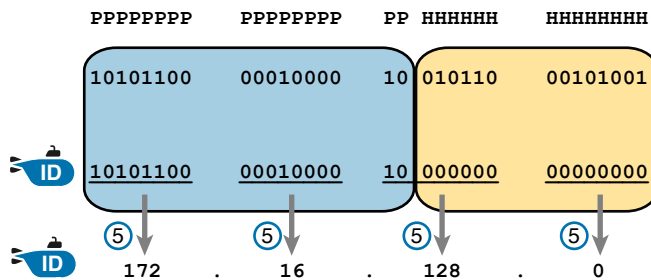


Starting at the top of Figure 14-4, the format of the IP address is represented with 18 prefix (P) and 14 host (H) bits in the mask (Step 1). The second row (Step 2) shows the binary version of the IP address, converted from the dotted-decimal notation (DDN) value 172.16.150.41. (If you have not yet used the conversion table in Appendix A, it might be useful to double-check the conversion of all four octets based on the table.)

The next two steps show the action to copy the IP address's prefix bits (Step 3) and give the host bits a value of binary 0 (Step 4). This resulting number is the subnet ID (in binary).

The last step, not shown in Figure 14-4, is to convert the subnet ID from binary to decimal. This book shows that conversion as a separate step, in Figure 14-5, mainly because many people make a mistake at this step in the process. When converting a 32-bit number (like an IP address or IP subnet ID) back to an IPv4 DDN, you must follow this rule:

Convert 8 bits at a time from binary to decimal, regardless of the line between the prefix and host parts of the number.



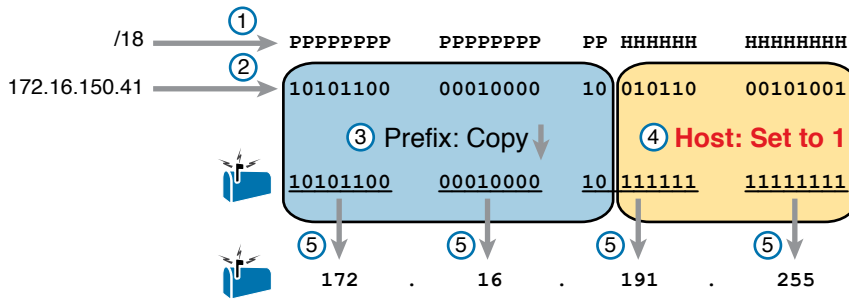
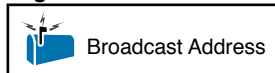
**Figure 14-5** Converting the Subnet ID from Binary to DDN

Figure 14-5 shows this final step. Note that the third octet (the third set of 8 bits) has 2 bits in the prefix and 6 bits in the host part of the number, but the conversion occurs for all 8 bits.

**NOTE** You can do the numeric conversions in Figures 14-4 and 14-5 by relying on the conversion table in Appendix A. To convert from DDN to binary, for each octet, find the decimal value in the table and then write down the 8-bit binary equivalent. To convert from binary back to DDN, for each octet of 8 bits, find the matching binary entry in the table and write down the corresponding decimal value. For example, 172 converts to binary 10101100, and 00010000 converts to decimal 16.

## Finding the Subnet Broadcast Address: Binary

Finding the subnet broadcast address uses a similar process. To find the subnet broadcast address, use the same binary process used to find the subnet ID, but instead of setting all the host bits to the lowest value (all binary 0s), set the host part to the highest value (all binary 1s). Figure 14-6 shows the concept.

**Legend:****Figure 14-6** Finding a Subnet Broadcast Address: Binary

The process in Figure 14-6 demonstrates the same first three steps shown in Figure 14-4. Specifically, it shows the identification of the prefix and host bits (Step 1), the results of converting the IP address 172.16.150.41 to binary (Step 2), and the copying of the prefix bits (first 18 bits, in this case). The difference occurs in the host bits on the right, changing all host bits (the last 14, in this case) to the largest possible value (all binary 1s). The final step converts the 32-bit subnet broadcast address to DDN format. Also, remember that with any conversion from DDN to binary or vice versa, the process always converts using 8 bits at a time. In particular, in this case, the entire third octet of binary 10111111 is converted back to decimal 191.

**Binary Practice Problems**

Figures 14-4 and 14-5 demonstrate a process to find the subnet ID using binary math. The following process summarizes those steps in written form for easier reference and practice:

**Key Topic**

- Step 1.** Convert the mask to prefix format to find the length of the prefix (/P) and the length of the host part (32 – P).
- Step 2.** Convert the IP address to its 32-bit binary equivalent.
- Step 3.** Copy the prefix bits of the IP address.
- Step 4.** Write down 0s for the host bits.
- Step 5.** Convert the resulting 32-bit number, 8 bits at a time, back to decimal.

The process to find the subnet broadcast address is exactly the same, except in Step 4, you set the bits to 1s, as shown in Figure 14-6.

Take a few moments and run through the following five practice problems on scratch paper. In each case, find both the subnet ID and subnet broadcast address. Also, record the prefix style mask:

1. 8.1.4.5, 255.255.0.0
2. 130.4.102.1, 255.255.255.0
3. 199.1.1.100, 255.255.255.0

4. 130.4.102.1, 255.255.252.0
5. 199.1.1.100, 255.255.255.224

Tables 14-4 through 14-8 show the results for the five different examples. The tables show the host bits in bold, and they include the binary version of the address and mask and the binary version of the subnet ID and subnet broadcast address.

**Table 14-4** Subnet Analysis for Subnet with Address 8.1.4.5, Mask 255.255.0.0

Prefix Length	/16	11111111 11111111 00000000 00000000
Address	8.1.4.5	00001000 00000001 00000100 00000101
Subnet ID	8.1.0.0	00001000 00000001 00000000 00000000
Broadcast Address	8.1.255.255	00001000 00000001 11111111 11111111

**Table 14-5** Subnet Analysis for Subnet with Address 130.4.102.1, Mask 255.255.255.0

Prefix Length	/24	11111111 11111111 11111111 00000000
Address	130.4.102.1	10000010 00000100 01100110 00000001
Subnet ID	130.4.102.0	10000010 00000100 01100110 00000000
Broadcast Address	130.4.102.255	10000010 00000100 01100110 11111111

**Table 14-6** Subnet Analysis for Subnet with Address 199.1.1.100, Mask 255.255.255.0

Prefix Length	/24	11111111 11111111 11111111 00000000
Address	199.1.1.100	11000111 00000001 00000001 01100100
Subnet ID	199.1.1.0	11000111 00000001 00000001 00000000
Broadcast Address	199.1.1.255	11000111 00000001 00000001 11111111

**Table 14-7** Subnet Analysis for Subnet with Address 130.4.102.1, Mask 255.255.252.0

Prefix Length	/22	11111111 11111111 11111100 00000000
Address	130.4.102.1	10000010 00000100 01100110 00000001
Subnet ID	130.4.100.0	10000010 00000100 01100100 00000000
Broadcast Address	130.4.103.255	10000010 00000100 01100111 11111111

**Table 14-8** Subnet Analysis for Subnet with Address 199.1.1.100, Mask 255.255.255.224

Prefix Length	/27	11111111 11111111 11111111 11100000
Address	199.1.1.100	11000111 00000001 00000001 01100100
Subnet ID	199.1.1.96	11000111 00000001 00000001 01100000
Broadcast Address	199.1.1.127	11000111 00000001 00000001 01111111

## Shortcut for the Binary Process

The binary process described in this section so far requires that all four octets be converted to binary and then back to decimal. However, you can easily predict the results in at least three of the four octets, based on the DDN mask. You can then avoid the binary math in all but one octet and reduce the number of binary conversions you need to do.

First, consider an octet, and that octet only, whose DDN mask value is 255. The mask value of 255 converts to binary 11111111, which means that all 8 bits are prefix bits. Thinking through the steps in the process, at Step 2, you convert the address to some number. At Step 3, you copy the number. At Step 4, you convert the same 8-bit number back to decimal. All you did in those three steps, in this one octet, is convert from decimal to binary and convert the same number back to the same decimal value!

In short, the subnet ID (and subnet broadcast address) are equal to the IP address in octets for which the mask is 255.

For example, the resident subnet ID for 172.16.150.41, mask 255.255.192.0 is 172.16.128.0. The first two mask octets are 255. Rather than think about the binary math, you could just start by copying the address's value in those two octets: 172.16.

Another shortcut exists for octets whose DDN mask value is decimal 0, or binary 00000000. With a decimal mask value of 0, the math always results in a decimal 0 for the subnet ID, no matter the beginning value in the IP address. Specifically, just look at Steps 4 and 5 in this case: At Step 4, you would write down 8 binary 0s, and at Step 5, you would convert 00000000 back to decimal 0.

The following revised process steps take these two shortcuts into account. However, when the mask is neither 0 nor 255, the process requires the same conversions. At most, you have to do only one octet of the conversions. To find the subnet ID, apply the logic in these steps for each of the four octets:



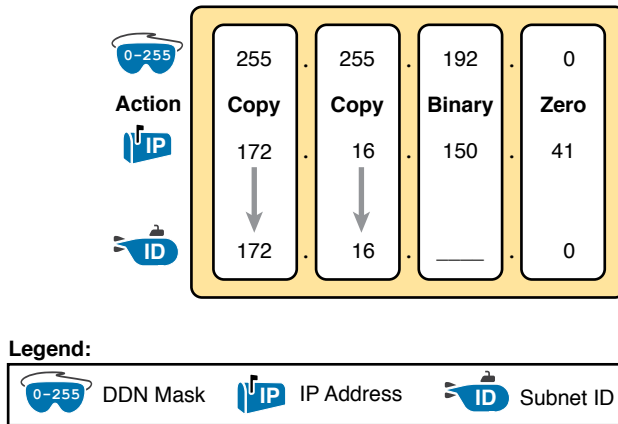
- Step 1.** If the mask = 255, copy the decimal IP address for that octet.
- Step 2.** If the mask = 0, write down a decimal 0 for that octet.
- Step 3.** If the mask is neither 0 nor 255 in this octet, use the same binary logic as shown in the section “Finding the Subnet ID: Binary,” earlier in this chapter.

Figure 14-7 shows an example of this process, again using 172.16.150.41, 255.255.192.0.

To find the subnet broadcast address, you can use a decimal shortcut similar to the one used to find the subnet ID: for DDN mask octets equal to decimal 0, set the decimal subnet broadcast address value to 255 instead of 0, as noted in the following list:



- Step 1.** If the mask = 255, copy the decimal IP address for that octet.
- Step 2.** If the mask = 0, write down a decimal 255 for that octet.
- Step 3.** If the mask is neither 0 nor 255 in this octet, use the same binary logic as shown in the section “Finding the Subnet Broadcast Address: Binary,” earlier in this chapter.



**Figure 14-7** Binary Shortcut Example

## Brief Note About Boolean Math

So far, this chapter has described how humans can use binary math to find the subnet ID and subnet broadcast address. However, computers typically use an entirely different binary process to find the same values, using a branch of mathematics called *Boolean algebra*. Computers already store the IP address and mask in binary form, so they do not have to do any conversions to and from decimal. Then, certain Boolean operations allow the computers to calculate the subnet ID and subnet broadcast address with just a few CPU instructions.

You do not need to know Boolean math to have a good understanding of IP subnetting. However, in case you are interested, computers use the following Boolean logic to find the subnet ID and subnet broadcast address, respectively:

Perform a *Boolean AND* of the IP address and mask. This process converts all host bits to binary 0.

Invert the mask, and then perform a *Boolean OR* of the IP address and inverted subnet mask. This process converts all host bits to binary 1s.

## Finding the Range of Addresses

Finding the range of usable addresses in a subnet, after you know the subnet ID and subnet broadcast address, requires only simple addition and subtraction. To find the first (lowest) usable IP address in the subnet, simply add 1 to the fourth octet of the subnet ID. To find the last (highest) usable IP address, simply subtract 1 from the fourth octet of the subnet broadcast address.

## Analyzing Existing Subnets: Decimal

Analyzing existing subnets using the binary process works well. However, some of the math takes time for most people, particularly the decimal-binary conversions. And you need to do the math quickly for the Cisco CCNA exam. For the exam, you really should be able to take an IP address and mask, and calculate the subnet ID and range of usable addresses within about 15 seconds. When using binary methods, most people require a lot of practice to be able to find these answers, even when using the abbreviated binary process.

This section discusses how to find the subnet ID and subnet broadcast address using only decimal math. Most people can find the answers more quickly using this process, at least after a little practice, as compared with the binary process. However, the decimal process does not tell you anything about the meaning behind the math. So, if you have not read the earlier section “Analyzing Existing Subnets: Binary,” it is worthwhile to read it for the sake of understanding subnetting. This section focuses on getting the right answer using a method that, after you have practiced, should be faster.

## Analysis with Easy Masks

With three easy subnet masks in particular, finding the subnet ID and subnet broadcast address requires only easy logic and literally no math. Three easy masks exist:

255.0.0.0

255.255.0.0

255.255.255.0

These easy masks have only 255 and 0 in decimal. In comparison, difficult masks have one octet that has neither a 255 nor a 0 in the mask, which makes the logic more challenging.

**NOTE** The terms *easy mask* and *difficult mask* are terms created for use in this book to describe the masks and the level of difficulty when working with each.

When the problem uses an easy mask, you can quickly find the subnet ID based on the IP address and mask in DDN format. Just use the following process for each of the four octets to find the subnet ID:

**Step 1.** If the mask octet = 255, copy the decimal IP address.

**Step 2.** If the mask octet = 0, write a decimal 0.

A similar simple process exists to find the subnet broadcast address, as follows:

**Step 1.** If the mask octet = 255, copy the decimal IP address.

**Step 2.** If the mask octet = 0, write a decimal 255.

Before moving to the next section, take some time to fill in the blanks in Table 14-9. Check your answers against Table 14-15 in the section “Answers to Earlier Practice Problems,” later in this chapter. Complete the table by listing the subnet ID and subnet broadcast address.

**Table 14-9** Practice Problems: Find Subnet ID and Broadcast Address, Easy Masks

	IP Address	Mask	Subnet ID	Broadcast Address
1	10.77.55.3	255.255.255.0		
2	172.30.99.4	255.255.255.0		
3	192.168.6.54	255.255.255.0		
4	10.77.3.14	255.255.0.0		
5	172.22.55.77	255.255.0.0		
6	1.99.53.76	255.0.0.0		

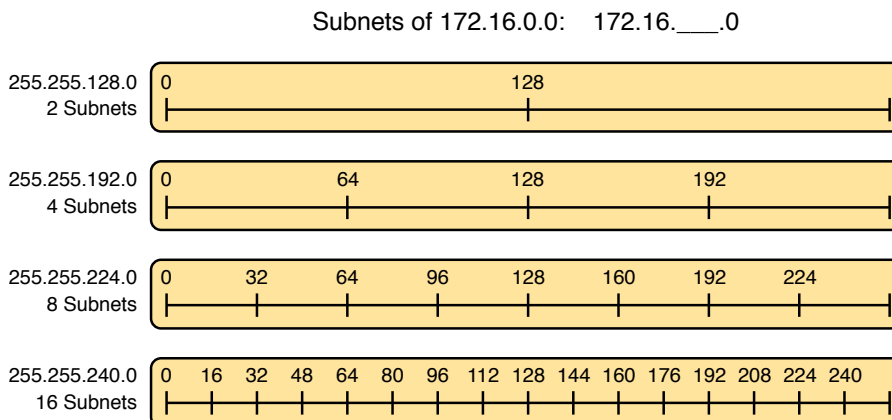
### Predictability in the Interesting Octet

Although three masks are easier to work with (255.0.0.0, 255.255.0.0, and 255.255.255.0), the rest make the decimal math a little more difficult, so we call these masks difficult masks. With difficult masks, one octet is neither a 0 nor a 255. The math in the other three octets is easy and boring, so this book calls the one octet with the more difficult math the *interesting octet*.

If you take some time to think about different problems and focus on the interesting octet, you will begin to see a pattern. This section takes you through that examination so that you can learn how to predict the pattern, in decimal, and find the subnet ID.

First, the subnet ID value has a predictable decimal value because of the assumption that a single subnet mask is used for all subnets of a single classful network. The chapters in this part of the book assume that, for a given classful network, the design engineer chooses to use a single subnet mask for all subnets. (See the section “One Size Subnet Fits All—Or Not” in Chapter 11, “Perspectives on IPv4 Subnetting,” for more details.)

To see that predictability, consider some planning information written down by a network engineer, as shown in Figure 14-8. The figure shows four different masks the engineer is considering using in an IPv4 network, along with Class B network 172.16.0.0. The figure shows the third-octet values for the subnet IDs that would be created when using mask 255.255.128.0, 255.255.192.0, 255.255.224.0, and 255.255.240.0, from top to bottom in the figure.

**Figure 14-8** Numeric Patterns in the Interesting Octet

First, to explain the figure further, look at the top row of the figure. If the engineer uses 255.255.128.0 as the mask, the mask creates two subnets, with subnet IDs 172.16.0.0 and 172.16.128.0. If the engineer uses mask 255.255.192.0, the mask creates four subnets, with subnet IDs 172.16.0.0, 172.16.64.0, 172.16.128.0, and 172.16.192.0.

If you take the time to look at the figure, the patterns become obvious. In this case:

Mask: 255.255.128.0      Pattern: Multiples of 128

Mask: 255.255.192.0      Pattern: Multiples of 64

Mask: 255.255.224.0      Pattern: Multiples of 32

Mask: 255.255.240.0      Pattern: Multiples of 16

To find the subnet ID, you just need a way to figure out what the pattern is. If you start with an IP address and mask, just find the subnet ID closest to the IP address, without going over, as discussed in the next section.

## Finding the Subnet ID: Difficult Masks

The following written process lists all the steps to find the subnet ID, using only decimal math. This process adds to the earlier process used with easy masks. For each octet:



**Step 1.** If the mask octet = 255, copy the decimal IP address.

**Step 2.** If the mask octet = 0, write a decimal 0.

**Step 3.** If the mask is neither, refer to this octet as the *interesting octet*:

**A.** Calculate the *magic number* as  $256 - \text{mask}$ .

**B.** Set the subnet ID's value to the multiple of the magic number that is closest to the IP address without going over.

The process uses two new terms created for this book: *magic number* and *interesting octet*. The term *interesting octet* refers to the octet identified at Step 3 in the process; in other words, it is the octet with the mask that is neither 255 nor 0. Step 3A then uses the term *magic number*, which is derived from the DDN mask. Conceptually, the magic number is the number you add to one subnet ID to get the next subnet ID in order, as shown in Figure 14-8. Numerically, it can be found by subtracting the DDN mask's value, in the interesting octet, from 256, as mentioned in Step 3A.

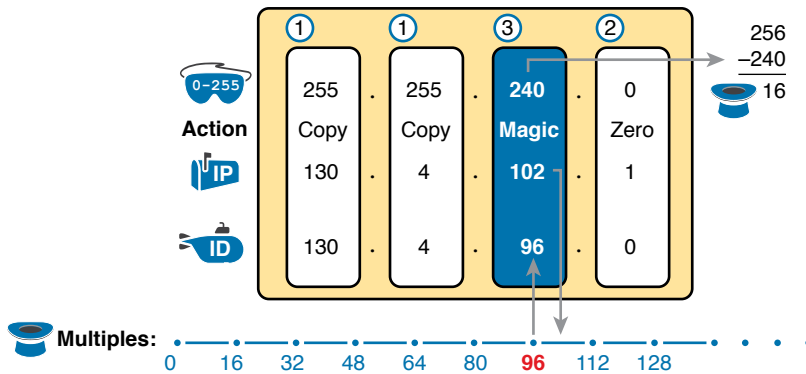
The best way to learn this process is to see it happen. In fact, if you can, stop reading now, use the companion website for this book, and watch the videos about finding the subnet ID with a difficult mask. These videos demonstrate this process. You can also use the examples on the next few pages that show the process being used on paper. Then follow the practice opportunities outlined in the section "Practice Analyzing Existing Subnets," later in this chapter.

## Resident Subnet Example 1

For example, consider the requirement to find the resident subnet for IP address 130.4.102.1, mask 255.255.240.0. The process does not require you to think about prefix bits versus host bits, convert the mask, think about the mask in binary, or convert the IP address to and from



binary. Instead, for each of the four octets, choose an action based on the value in the mask. Figure 14-9 shows the results; the circled numbers in the figure refer to the step numbers in the written process to find the subnet ID, as listed in the previous few pages.



**Figure 14-9** Find the Subnet ID: 130.4.102.1, 255.255.240.0

First, examine the three uninteresting octets (1, 2, and 4, in this example). The process keys on the mask, and the first two octets have a mask value of 255, so simply copy the IP address to the place where you intend to write down the subnet ID. The fourth octet has a mask value of 0, so write down a 0 for the fourth octet of the subnet ID.

The most challenging logic occurs in the interesting octet, which is the third octet in this example, because of the mask value 240 in that octet. For this octet, Step 3A asks you to calculate the magic number as  $256 - \text{mask}$ . That means you take the mask's value in the interesting octet (240, in this case) and subtract it from 256:  $256 - 240 = 16$ . The subnet ID's value in this octet must be a multiple of decimal 16, in this case.

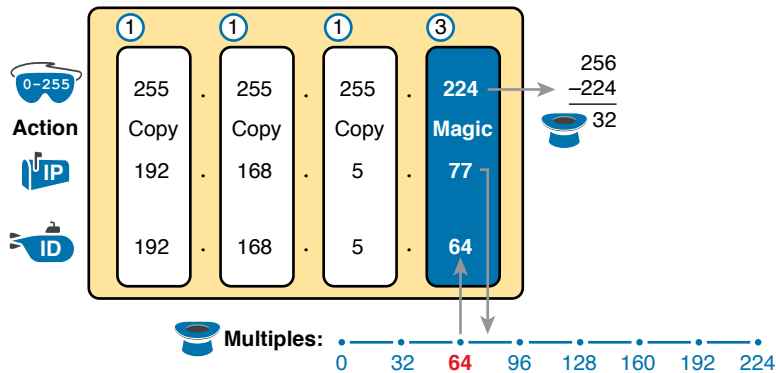
Step 3B then asks you to find the multiples of the magic number (16, in this case) and choose the one closest to the IP address without going over. Specifically, that means that you should mentally calculate the multiples of the magic number, starting at 0. (Do not forget to start at 0!) Count, starting at 0: 0, 16, 32, 48, 64, 80, 96, 112, and so on. Then, find the multiple closest to the IP address value in this octet (102, in this case), without going over 102. So, as shown in Figure 14-9, you make the third octet's value 96 to complete the subnet ID of 130.4.96.0.

## Resident Subnet Example 2

Consider another example: 192.168.5.77, mask 255.255.255.224. Figure 14-10 shows the results.

The three uninteresting octets (1, 2, and 3, in this case) require only a little thought. For each octet, each with a mask value of 255, just copy the IP address.

For the interesting octet, at Step 3A, the magic number is  $256 - 224 = 32$ . The multiples of the magic number are 0, 32, 64, 96, and so on. Because the IP address value in the fourth octet is 77, in this case, the multiple must be the number closest to 77 without going over; therefore, the subnet ID ends with 64, for a value of 192.168.5.64.



**Figure 14-10** Resident Subnet for 192.168.5.77, 255.255.255.224

**Resident Subnet Practice Problems**

Before moving to the next section, take some time to fill in the blanks in Table 14-10. Check your answers against Table 14-16 in the section “Answers to Earlier Practice Problems,” later in this chapter. Complete the table by listing the subnet ID in each case. The text following Table 14-16 also lists explanations for each problem.

**Table 14-10** Practice Problems: Find Subnet ID, Difficult Masks

Problem	IP Address	Mask	Subnet ID
1	10.77.55.3	255.248.0.0	
2	172.30.99.4	255.255.192.0	
3	192.168.6.54	255.255.255.252	
4	10.77.3.14	255.255.128.0	
5	172.22.55.77	255.255.254.0	
6	1.99.53.76	255.255.255.248	

**Finding the Subnet Broadcast Address: Difficult Masks**

To find a subnet’s broadcast address, a similar process can be used. For simplicity, this process begins with the subnet ID, rather than the IP address. If you happen to start with an IP address instead, use the processes in this chapter to first find the subnet ID, and then use the following process to find the subnet broadcast address for that same subnet. For each octet:

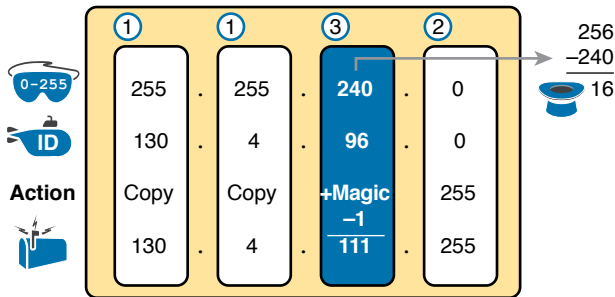


- Step 1.** If the mask octet = 255, copy the subnet ID.
- Step 2.** If the mask octet = 0, write 255.
- Step 3.** If the mask is neither, identify this octet as the *interesting octet*:
  - A.** Calculate the *magic number* as 256 – mask.
  - B.** Take the subnet ID’s value, add the magic number, and subtract 1 (ID + magic – 1).

As with the similar process used to find the subnet ID, you have several options for how to best learn and internalize the process. If you can, stop reading now, use the companion web-site for this book, and watch the videos listed for this chapter. Also, look at the examples in this section, which show the process being used on paper. Then, follow the practice opportunities outlined in the section “Additional Practice for This Chapter’s Processes.”

### Subnet Broadcast Example 1

The first example continues the first example from the section “Finding the Subnet ID: Difficult Masks,” earlier in this chapter, as demonstrated in Figure 14-9. That example started with the IP address/mask of 130.4.102.1, 255.255.240.0, and showed how to find subnet ID 130.4.96.0. Figure 14-11 now begins with that subnet ID and the same mask.



**Figure 14-11** Find the Subnet Broadcast: 130.4.96.0, 255.255.240.0

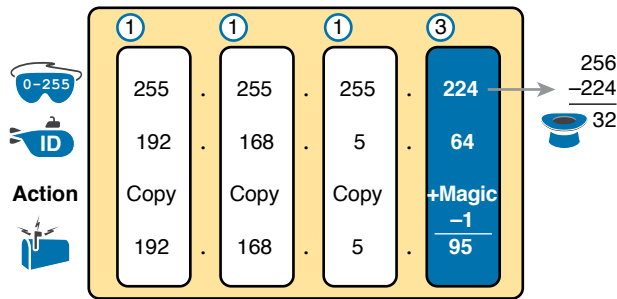
First, examine the three uninteresting octets (1, 2, and 4). The process keys on the mask, and the first two octets have a mask value of 255, so simply copy the subnet ID to the place where you intend to write down the subnet broadcast address. The fourth octet has a mask value of 0, so write down a 255 for the fourth octet.

The logic related to the interesting octet occurs in the third octet in this example because of the mask value 240. First, Step 3A asks you to calculate the magic number, as  $256 - \text{mask}$ . (If you had already calculated the subnet ID using the decimal process in this book, you should already know the magic number.) At Step 3B, you take the subnet ID’s value (96), add the magic number (16), and subtract 1, for a total of 111. That makes the subnet broadcast address 130.4.111.255.

### Subnet Broadcast Example 2

Again, this example continues an earlier example, from the section “Resident Subnet Example 2,” as demonstrated in Figure 14-10. That example started with the IP address/mask of 192.168.5.77, mask 255.255.255.224 and showed how to find subnet ID 192.168.5.64. Figure 14-12 now begins with that subnet ID and the same mask.

First, examine the three uninteresting octets (1, 2, and 3). The process keys on the mask, and the first three octets have a mask value of 255, so simply copy the subnet ID to the place where you intend to write down the subnet broadcast address.



**Figure 14-12** Find the Subnet Broadcast: 192.168.5.64, 255.255.255.224

The interesting logic occurs in the interesting octet, the fourth octet in this example, because of the mask value 224. First, Step 3A asks you to calculate the magic number, as 256 – mask. (If you had already calculated the subnet ID, it is the same magic number because the same mask is used.) At Step 3B, you take the subnet ID’s value (64), add magic (32), and subtract 1, for a total of 95. That makes the subnet broadcast address 192.168.5.95.

Subnet Broadcast Address Practice Problems

Before moving to the next section, take some time to do several practice problems on a scratch piece of paper. Go back to Table 14-10, which lists IP addresses and masks, and practice by finding the subnet broadcast address for all the problems in that table. Then check your answers against Table 14-17 in the section “Answers to Earlier Practice Problems,” later in this chapter.

Practice Analyzing Existing Subnets

As with the other subnetting math in this book, using a two-phase approach may help. Take time now to practice until you feel like you understand the process. Then, before the exam, make sure you master the math. Table 14-11 summarizes the key concepts and suggestions for this two-phase approach.

**Table 14-11** Keep-Reading and Take-Exam Goals for This Chapter’s Topics

	Before Moving to the Next Chapter	Before Taking the Exam
Focus On...	Learning how	Being correct and fast
Tools Allowed	All	Your brain and a notepad
Goal: Accuracy	90% correct	100% correct
Goal: Speed	Any speed	20–30 seconds

A Choice: Memorize or Calculate

As described in this chapter, the decimal processes to find the subnet ID and subnet broadcast address do require some calculation, including the calculation of the magic number (256 – mask). The processes also use a DDN mask, so if an exam question gives you a prefix-style mask, you need to convert to DDN format before using the process in this book.

Over the years, some people have told me they prefer to memorize a table to find the magic number. These tables could list the magic number for different DDN masks and prefix masks, so you avoid converting from the prefix mask to DDN. Table 14-12 shows an example of such a table. Feel free to ignore this table, use it, or make your own.

**Table 14-12** Reference Table: DDN Mask Values, Binary Equivalent, Magic Numbers, and Prefixes

Prefix, interesting octet 2	/9	/10	/11	/12	/13	/14	/15	/16
Prefix, interesting octet 3	/17	/18	/19	/20	/21	/22	/23	/24
Prefix, interesting octet 4	/25	/26	/27	/28	/29	/30		
Magic number	128	64	32	16	8	4	2	1
DDN mask in the interesting octet	128	192	224	240	248	252	254	255

## Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 14-13 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

**Table 14-13** Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review memory tables		Website
Practice mask analysis		Website, Appendix F
Practice analyzing existing subnets		Website, Appendix F

## Review All the Key Topics



**Table 14-14** Key Topics for Chapter 14

Key Topic Element	Description	Page Number
List	Definition of a subnet’s key numbers	322
Table 14-2	Key facts about the subnet ID	324
Table 14-3	Key facts about the subnet broadcast address	325
List	Steps to use binary math to find the subnet ID	328
List	General steps to use binary and decimal math to find the subnet ID	330
List	Steps to use decimal and binary math to find the subnet broadcast address	330
List	Steps to use only decimal math to find the subnet ID	334
List	Steps to use only decimal math to find the subnet broadcast address	336

## Key Terms You Should Know

resident subnet, subnet ID, subnet number, subnet address, subnet broadcast address

## Additional Practice for This Chapter’s Processes

You can do more practice with the processes in this chapter with a pair of practice sets. Both give you practice at analyzing existing subnets. You may do each practice set using the following tools:

**Application:** Use the “Analyzing Existing Subnets” exercises 1 and 2 on the companion website, listed under the Chapter Review for this chapter.

**PDF:** Alternatively, practice the same problems found in these apps using companion website Appendix F, “Practice for Chapter 14: Analyzing Existing Subnets.”

## Answers to Earlier Practice Problems

This chapter includes practice problems spread around different locations in the chapter. The answers are located in Tables 14-15, 14-16, and 14-17.

**Table 14-15** Answers to Problems in Table 14-9

	IP Address	Mask	Subnet ID	Broadcast Address
1	10.77.55.3	255.255.255.0	10.77.55.0	10.77.55.255
2	172.30.99.4	255.255.255.0	172.30.99.0	172.30.99.255
3	192.168.6.54	255.255.255.0	192.168.6.0	192.168.6.255
4	10.77.3.14	255.255.0.0	10.77.0.0	10.77.255.255
5	172.22.55.77	255.255.0.0	172.22.0.0	172.22.255.255
6	1.99.53.76	255.0.0.0	1.0.0.0	1.255.255.255

**Table 14-16** Answers to Problems in Table 14-10

	IP Address	Mask	Subnet ID
1	10.77.55.3	255.248.0.0	10.72.0.0
2	172.30.99.4	255.255.192.0	172.30.64.0
3	192.168.6.54	255.255.255.252	192.168.6.52
4	10.77.3.14	255.255.128.0	10.77.0.0
5	172.22.55.77	255.255.254.0	172.22.54.0
6	199.53.76	255.255.255.248	199.53.72

The following list explains the answers for Table 14-16:

1. The second octet is the interesting octet, with magic number  $256 - 248 = 8$ . The multiples of 8 include 0, 8, 16, 24, ..., 64, 72, and 80. 72 is closest to the IP address value in that same octet (77) without going over, making the subnet ID 10.72.0.0.
2. The third octet is the interesting octet, with magic number  $256 - 192 = 64$ . The multiples of 64 include 0, 64, 128, and 192. 64 is closest to the IP address value in that same octet (99) without going over, making the subnet ID 172.30.64.0.
3. The fourth octet is the interesting octet, with magic number  $256 - 252 = 4$ . The multiples of 4 include 0, 4, 8, 12, 16, ..., 48, 52, and 56. 52 is the closest to the IP address value in that same octet (54) without going over, making the subnet ID 192.168.6.52.
4. The third octet is the interesting octet, with magic number  $256 - 128 = 128$ . Only two multiples exist that matter: 0 and 128. 0 is the closest to the IP address value in that same octet (3) without going over, making the subnet ID 10.77.0.0.
5. The third octet is the interesting octet, with magic number  $256 - 254 = 2$ . The multiples of 2 include 0, 2, 4, 6, 8, and so on—essentially all even numbers. 54 is closest to the IP address value in that same octet (55) without going over, making the subnet ID 172.22.54.0.
6. The fourth octet is the interesting octet, with magic number  $256 - 248 = 8$ . The multiples of 8 include 0, 8, 16, 24, ..., 64, 72, and 80. 72 is closest to the IP address value in that same octet (76) without going over, making the subnet ID 199.53.72.

**Table 14-17** Answers to Problems in the Section “Subnet Broadcast Address Practice Problems”

	Subnet ID	Mask	Broadcast Address
1	10.72.0.0	255.248.0.0	10.79.255.255
2	172.30.64.0	255.255.192.0	172.30.127.255
3	192.168.6.52	255.255.255.252	192.168.6.55
4	10.77.0.0	255.255.128.0	10.77.127.255
5	172.22.54.0	255.255.254.0	172.22.55.255
6	199.53.72	255.255.255.248	199.53.79

The following list explains the answers for Table 14-17:

1. The second octet is the interesting octet. Completing the three easy octets means that the broadcast address in the interesting octet will be 10.\_\_\_\_.255.255. With magic number  $256 - 248 = 8$ , the second octet will be 72 (from the subnet ID), plus 8, minus 1, or 79.
2. The third octet is the interesting octet. Completing the three easy octets means that the broadcast address in the interesting octet will be 172.30.\_\_\_\_.255. With magic number  $256 - 192 = 64$ , the interesting octet will be 64 (from the subnet ID), plus 64 (the magic number), minus 1, for 127.
3. The fourth octet is the interesting octet. Completing the three easy octets means that the broadcast address in the interesting octet will be 192.168.6.\_\_\_\_. With magic number  $256 - 252 = 4$ , the interesting octet will be 52 (the subnet ID value), plus 4 (the magic number), minus 1, or 55.
4. The third octet is the interesting octet. Completing the three easy octets means that the broadcast address will be 10.77.\_\_\_\_.255. With magic number  $256 - 128 = 128$ , the interesting octet will be 0 (the subnet ID value), plus 128 (the magic number), minus 1, or 127.
5. The third octet is the interesting octet. Completing the three easy octets means that the broadcast address will be 172.22.\_\_\_\_.255. With magic number  $256 - 254 = 2$ , the broadcast address in the interesting octet will be 54 (the subnet ID value), plus 2 (the magic number), minus 1, or 55.
6. The fourth octet is the interesting octet. Completing the three easy octets means that the broadcast address will be 1.99.53.\_\_\_\_. With magic number  $256 - 248 = 8$ , the broadcast address in the interesting octet will be 72 (the subnet ID value), plus 8 (the magic number), minus 1, or 79.



*This page intentionally left blank*



# Part IV Review

Keep track of your part review progress with the checklist in Table P4-1. Details on each task follow the table.

**Table P4-1** Part IV Part Review Checklist

Activity	1st Date Completed	2nd Date Completed
Repeat All DIKTA Questions		
Answer Part Review Questions		
Review Key Topics		
Subnetting Exercises in Appendices on Companion Website		
Videos on Companion Website		
Subnetting Exercises on Author's Blog		
Subnetting Exercises in IP Subnetting Practice Question Kit		
Subnetting Labs in Pearson Network Simulator		

## Repeat All DIKTA Questions

For this task, use the PCPT software to answer the “Do I Know This Already?” questions again for the chapters in this part of the book.

## Answer Part Review Questions

For this task, use PCPT to answer the Part Review questions for this part of the book.

## Review Key Topics

Review all key topics in all chapters in this part, either by browsing the chapters or by using the Key Topics application on the companion website.

## Watch Videos

Chapter 14 recommends several videos as listed on this book's companion website. These videos help you understand how to use the process in the book to find facts about subnets, like the range of usable addresses in the subnet.

## Subnetting Exercises

Chapters 12, 13, and 14 list some subnetting exercises, along with time and accuracy goals. Now is a good time to work on those goals. Some options include the following:

**Practice from this book's appendices or web applications:** The Chapter Review sections of Chapters 12, 13, and 14 mention addressing and subnetting exercises included with this book. Find all the related applications in the Part IV Review section of the companion website:

Appendix D, "Practice for Chapter 12: Analyzing Classful IPv4 Networks"

Appendix E, "Practice for Chapter 13: Analyzing Subnet Masks"

Appendix F, "Practice for Chapter 14: Analyzing Existing Subnets"

**Pearson Network Simulator:** The full Pearson ICND1 or CCNA simulator has subnetting math exercises that you can do by using CLI commands. Look for the labs with "IP Address Rejection" and "Subnet ID Calculation" in their names.

**Author's blog:** I've written a few dozen subnetting exercises on the blog over the years. Just look at the Questions menu item at the top of the page, and you will see a variety of IPv4 addressing and subnetting question types. Start at <http://blog.certskills.com>.



Parts V and VI work together to reveal the details of how to implement IPv4 routing in Cisco routers. To that end, Part V focuses on the most common features for Cisco routers, including IP address configuration, connected routes, and static routes. Part VI then goes into some detail about the one IP routing protocol discussed in this book: OSPF Version 2 (OSPFv2).

Part V follows a progression of topics. First, Chapter 15 examines the fundamentals of routers—the physical components, how to access the router command-line interface (CLI), and the configuration process. Chapter 15 makes a close comparison of the switch CLI and its basic administrative commands so that you have to learn only new commands that apply to routers but not to switches.

Chapter 16 then moves on to discuss how to configure routers to route IPv4 packets in the most basic designs. Those designs require a simple IP address/mask configuration on each interface, with the addition of a static route command—a command that directly configures a route into the IP routing table—for each destination subnet.

By the end of Chapter 16, you should have a solid understanding of how to enable IP addressing and routing in a Cisco router, so Chapter 17 continues the progression into more challenging but more realistic configurations related to routing between subnets in a LAN environment. Most LANs use many VLANs, with one subnet per VLAN. Cisco routers and switches can be configured to route packets between those subnets, with more than a few twists in the configuration.

Finally, Part V closes with a chapter about troubleshooting IPv4 routing. The chapter features the **ping** and **tracert** commands, two commands that can help you discover not only whether a routing problem exists but also where the problem exists. Chapters 15, 16, and 17 show how to confirm whether a route has been added to one router's routing table, while the commands discussed in Chapter 18 teach you how to test the end-to-end routes from sending host to receiving host.