Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

**Module**

**8**

**Segmentation**

* [Module Introduction](javascript://)
* **8-1**[Network Segmentation](javascript://)
* **8-2**[Subnet Masks](javascript://)
  + **8-2a**[How Subnet Masks Work](javascript://)
* **8-3**[Calculating Subnets](javascript://)
  + **8-3a**[IPv4 Subnet Calculation in Binary](javascript://)
  + **8-3b**[IPv4 Subnet Calculations Using Formulas](javascript://)
  + **8-3c**[Subnet Mask Tables](javascript://)
  + **8-3d**[Subnetting Questions on Exams](javascript://)
  + **8-3e**[Implementing Subnets on a Network](javascript://)
  + **8-3f**[Variable Length Subnet Mask (VLSM)](javascript://)
  + **8-3g**[Subnets in IPv6](javascript://)
* **8-4**[Virtual LANs (VLANs)](javascript://)
  + **8-4a**[Switch Port Configuration](javascript://)
  + **8-4b**[VLAN Trunks](javascript://)
  + **8-4c**[VLANs and Subnets](javascript://)
  + **8-4d**[Types of VLANs](javascript://)
  + **8-4e**[View Configured VLANs](javascript://)
  + **8-4f**[Dynamic VLAN Assignment](javascript://)
  + **8-4g**[Troubleshoot and Secure VLANs](javascript://)
* **8-5**[Module Review](javascript://)
  + **8-5a**[Module Summary](javascript://)
  + **8-5b**[Key Terms](javascript://)
  + **8-5c**[Review Questions](javascript://)
  + **8-5d**[Hands-On Projects](javascript://)
  + **8-5e**[Capstone Projects](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Introduction

### Objectives

After reading this module and completing the exercises, you should be able to:

* **1**Explain the purposes of network segmentation
* **2**Describe how subnetting works
* **3**Calculate subnets
* **4**Configure VLANs

**On the Job**

I recently provided the technical expertise to build a new FM radio station in rural Wisconsin. In addition to specifying and installing microphones, speakers, and sound boards, I also designed and created the station’s network. Within the station’s building, the network connects studios, office computers, and a VoIP (Voice over IP) telephone system. Beyond the building, the network sends the station’s broadcast signal to its antenna.

When I set up the radio station network, I decided to separate different kinds of network traffic. To do this, I chose to create VLANs, rather than creating multiple physical networks, for several reasons, not the least of which is the cost of acquiring and maintaining multiple network switches. Managing multiple subnets on a single device has simplified deployment and long-term maintenance.

The VLANs are set up as follows:

* VLAN 101 (IP address subnet 10.10.1.0/24) is the transmitter network.
* VLAN 201 (IP address subnet 10.20.1.0/24) is the studio network.
* VLAN 301 (IP address subnet 10.30.1.0/24) is the office network.
* VLAN 401 (IP address subnet 10.40.1.0/24) is the telephone network.

Using VLANs allows the station to keep general Internet traffic off the latency-sensitive studio subnet. The systems on the studio subnet include the audio automation players and the analog-to-digital audio encoders. These computers receive and send digital audio over the network and demand timely delivery of packets. Further, these computers do not need to access Internet resources. We chose to isolate these systems from the others using VLANs (and access lists) to help guarantee the timely delivery of audio data.

Meanwhile, placing our VoIP telephones on a separate VLAN prevents studio audio traffic, as well as the general office and Internet traffic, from interfering with the telephone system traffic.

**David Klann**

**WDRT 91.9FM**

Network segmentation takes the divide-and-conquer approach to network management. When done well, it increases both performance and security on a network. A network can be segmented physically by creating multiple LANs or logically using VLANs (virtual LANs). Either way, the larger broadcast domain is divided into smaller segments, and the IP address space is subdivided as well.

In this module, you’ll learn about two important concepts that enable and support network segmentation: subnets and VLANs. Fundamentally, a subnet is a group of IP addresses, and a VLAN is a group of ports on one or more switches. Subnets and VLANs usually work together, but you’ll learn about each of them separately first.

Before you dig into how subnetting works, you’ll take a brief look at why you might want to segment a network using either multiple LANs or multiple VLANs. Then you’ll explore the important role subnetting plays in network segmentation. And finally, you’ll see how VLANs work and the unique flexibility they offer.

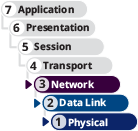
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 8-1Network Segmentation



### Certification

* 4.1

Explain common security concepts.

Average reading time: 8 minutes

When a network is segmented into multiple smaller networks, traffic on one network is separated from another network’s traffic and each network is its own broadcast domain. A network administrator might separate a network’s traffic into smaller portions to accomplish the following goals:

* **Enhance security**—Transmissions in broadcast domains are limited to each network so there’s less possibility of hackers or malware reaching remote, protected networks in the enterprise domain. At the same time, other devices, such as a web server, can be made more accessible from the open Internet than the rest of the network is. For example, a [**screened subnet**](javascript://) (formerly called a DMZ, or demilitarized zone) can provide an area of the network with less stringent security policies that allow traffic from website visitors, while other portions of the network cannot be accessed from the Internet. Enforcing network segmentation is one layer of security in a solid defense-in-depth strategy because it secures sensitive network traffic separately from other traffic.
* **Improve performance**—Segmenting limits broadcast traffic by decreasing the size of each broadcast domain. The more efficient use of bandwidth results in better overall network performance. The [On the Job](javascript://) story at the beginning of this module gave an excellent example of how this applies in a real-world situation where time-sensitive studio traffic was isolated and prioritized differently than general office and Internet traffic.
* **Simplify troubleshooting**—When troubleshooting, rather than examining the whole network for errors or bottlenecks, the network administrator can narrow down the problem area to a smaller network segment. For example, suppose a network is subdivided with separate smaller networks for Accounting, Human Resources, and IT. One day there’s trouble transmitting data only to a certain group of users—those on the Accounting network. This fact gives the network administrator some significant insight into the nature of the problem.

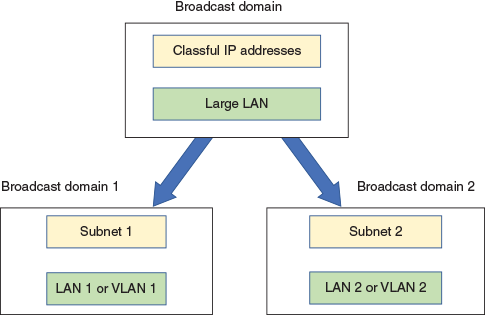
Networks are commonly segmented according to one of the following groupings:

* **Geographic locations**—For example, the floors of a building connected by a LAN, or the buildings connected by a WAN
* **Departmental boundaries**—For example, the Accounting, Human Resources, and Sales departments
* **Device types**—For example, printers, desktops, and IP phones

As you explore options for network segmentation throughout this module, keep in mind there are a variety of ways to go about separating broadcast domains on a network. Each segmentation method addresses different needs, offering varying capabilities and limitations. The OSI model also plays a part in network segmentation. You can use physical devices such as routers to create separate LANs. At layer 2, you can create virtual LANs, which you’ll learn more about later in this module. At layer 3, you can use subnetting to organize devices within the available IP address space, whether the LANs are defined physically or virtually. [Figure 8-1](javascript://) can help you visualize the relationship between these various concepts.

**Figure 8-1**

Network segmentation divides a large broadcast domain into smaller broadcast domains



**Applying Concepts 8-1**

### Binary Calculations

For calculations used with this module’s learning objectives, you’ll need to become comfortable with converting decimal numbers to binary and back, especially for a few, commonly used values. You won’t be allowed to use a calculator during the CompTIA Network+ exam, but when calculating conversions on the job, using a calculator can make the task much simpler. Take, for example, the decimal number 131. Complete the following steps to convert it to a binary number using the Windows 10 Calculator:

1. 1

Open the Calculator app. Click the menu icon, and then click **Programmer**. Verify that the **DEC** option is selected (it should show a blue bar to the left of the option).

1. 2

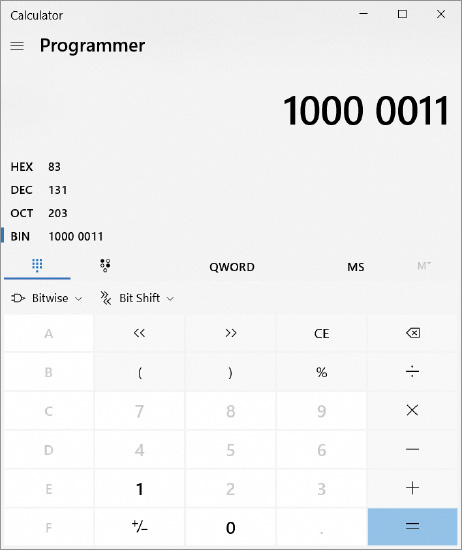
Type **131**. Other formats of this number are listed automatically. The binary equivalent of the decimal number 131, which is 1000 0011, appears next to the BIN option.

1. 3

Select the **BIN** option, as shown in [Figure 8-2](javascript://). Type any 8-digit binary number to convert it to a decimal number. What binary number did you enter? What decimal number does it convert to?

**Figure 8-2**

Use the Windows Calculator app to convert between decimal and binary



Enlarge Image

If you’re connected to the Internet and using a web browser, you can quickly convert binary and decimal numbers using Google calculator:

1. 4

Go to [google.com](http://google.com/" \t "_blank), and then type the number you want to convert along with the desired format in the search text box. For example, to convert the decimal number 131 into binary form, enter **131 in binary**. You see the following result: . The prefix “0b” (that’s a zero, not the letter O) indicates that the following number is in binary format. Notice that Google assumes a number is in decimal form unless stated otherwise.

1. 5

To convert a binary number into decimal form, type 0b (again, that’s the number zero, not the letter O) before the binary number. For example, entering **0b10000011 in decimal** returns the decimal number 131.

To best prepare yourself for the CompTIA Network+ exam, consider manually performing the calculations in this module and use the calculator only to check your results. If your manual calculations don’t match the calculator’s computations, make sure you investigate carefully to see where you made a mistake. Repeat until your calculations are consistently correct.

Regardless of how you go about segmenting a network, you’ll need to find the right balance between separating and connecting devices within each network portion. Let’s begin with a discussion of how subnetting complements physical or virtual segmentation.

**Remember This…**

* List advantages of network segmentation.
* Explain how segmentation increases network security.

**Self-Check**

1. Which of the following techniques does not break up a large broadcast domain into smaller broadcast domains?

Answer

* 1. Adding more routers to a network
  2. Adding more layer 2 switches to a network
  3. Adding more VLANs to a network

1. What is the binary number 1111 1111 in decimal?

Answer

* 1. 255
  2. 100
  3. 8
  4. 192

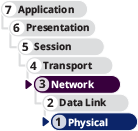
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 8-2Subnet Masks



### Certification

* 1.4

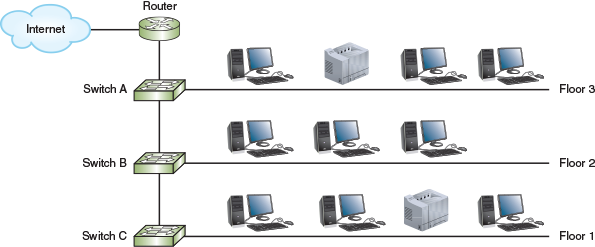
Given a scenario, configure a subnet and use appropriate IP addressing schemes.

Average reading time: 15 minutes

Suppose your company’s network is expanding from 20 or 30 computers and other devices to more than a hundred devices. The network began as a single LAN with workstations and printers connected by a few layer 2 switches, one switch connected to a router, and then on to the ISP. See [Figure 8-3](javascript://). Because there is only a single LAN (broadcast domain), any node on the network can communicate directly with any other node, and the one router serves as the default gateway for the whole network. The entire LAN has one pool of IP addresses, for example, 192.168.89.0/24, with a subnet mask of 255.255.255.0.

**Figure 8-3**

A single LAN with some switches and a router

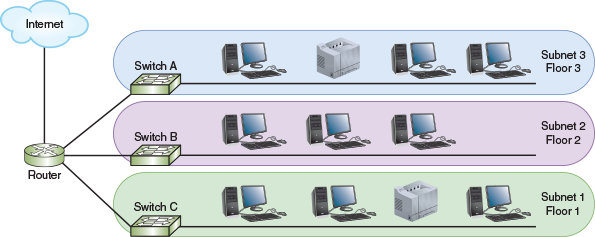


Enlarge Image

As the network grows, you’ll need to better manage network traffic by segmenting the network so that each floor contains a separate LAN, or broadcast domain. One way to accomplish this is to connect each switch separately to the router so each switch is using its own router interface, as illustrated in [Figure 8-4](javascript://).

**Figure 8-4**

A separate subnet for each floor

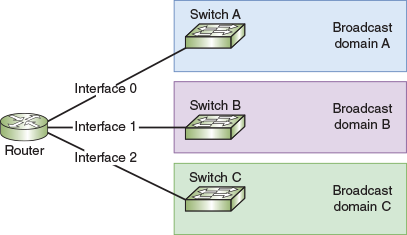


Enlarge Image

A router doesn’t forward broadcast traffic between its interfaces (see [Figure 8-5](javascript://)), and so this configuration will break up the one, large broadcast domain into three smaller broadcast domains. You can think of a router as a broadcast boundary, and fundamentally, routers are tools you can use to divide and conquer network traffic. However, you also need to manage the IP address space at the logical layer. To do this, you need to configure (either manually or through the DHCP server) the clients on each subnet so they know which devices are on their own subnet and which devices are not. And you need to configure the router so it can forward traffic between the LANs as necessary.

**Figure 8-5**

A router divides broadcast domains between its interfaces



At this point, you have three separate and smaller LANs, or subnets, within the larger network. However, a device on Subnet 2, for example, doesn’t yet know that devices on Subnet 3 aren’t still sharing the same LAN. How do you divide the pool of IP addresses so that a computer on Subnet 2 knows to send transmissions for devices on other subnets to the default gateway instead of trying to communicate with them directly? The solution is to divide your pool of IP addresses into three groups, or subnets, one for each LAN or floor of the building. This technique is called subnetting.

Subnetting helps solve the fundamental problem with classful addressing: too many node addresses assigned to each classful network. For example, a single class B network has 65,536 IP addresses all on the one LAN. Imagine the challenges involved in managing such a highly populated LAN, not to mention the poor performance that would result. Subnetting helps manage IP address space more efficiently. Also, though it might not be obvious at this point, using well-chosen subnets provides the following benefits:

* Network documentation is easier to manage.
* Problems are easier to locate and resolve.
* Routers can more easily manage IP address spaces that don’t overlap.
* Routing is more efficient on larger networks when IP address spaces are mathematically related at a binary level.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-2aHow Subnet Masks Work

A device uses a subnet mask to determine which subnet or network it belongs to. To review a little of what you’ve already learned regarding IP addresses and subnet masks, recall that an IPv4 address has 32 bits and is divided into two parts: the network portion, which identifies the network and is called the network ID, and the node portion, which identifies the node and is called the node ID or host ID. You can know where the dividing line is between the portions of bits by looking at the subnet mask, or by checking the CIDR block (such as /24).

**Note 8-1**

You might sometimes find the term network ID used interchangeably with the terms network number or network prefix.

Also, when calculating subnets, nodes are often referred to generically as hosts, even if that node is a networking device such as a firewall or router. The reason for this is that subnetting formulas must distinguish between network ID bits (referred to as “n”) and node or host ID bits (referred to as “h”). This course uses a similar approach and refers to all subnet nodes as hosts to make this distinction easier to track during your calculations. As you work, keep in mind that all hosts are nodes but not all nodes are hosts, even though all nodes are referred to as “hosts” when calculating subnets. Networking devices (such as routers and switches) are nodes that are not hosts. They serve a fundamentally different purpose on a network than do networked devices, or hosts, such as servers, workstations, and printers. However, each node (including non-hosts such as routers and firewalls, and hosts such as servers and printers) receives one of the available host IDs within the subnet.

When a computer is ready to send a transmission to another device, it first compares the bits in its own network ID to the bits in the network ID of the destination device. If the bits match, the other device is on the sending computer’s own network, and it sends the transmission directly to that device. If the bits don’t match, the destination is on another network, and the computer sends the transmission to the default gateway on its network. The gateway is responsible for sending the transmission toward the correct network.

How does a computer use a subnet mask to determine how many bits of its IP address is the network ID? Recall that an IPv4 subnet mask is 32 bits long. The number of 1s in the subnet mask’s bits determines the number of bits in the IP address that belong to the network ID. For example, suppose a computer has an IP address of 192.168.123.132 and its subnet mask in decimal is 255.255.255.0. To identify the bits that make up the network ID, first convert these numbers to binary, as follows:

* IP address 192.168.123.132 in binary:
  + **11000000.10101000.01111011**.10000100
* Subnet mask 255.255.255.0 in binary:
  + 11111111.11111111.11111111.00000000

**Note 8-2**

In this example and in many others in this module, a **bold red font** is used for the network ID portion of an IP address.

A subnet mask is always a series of 1s followed by a series of 0s. The 1s mark the network portion of an IP address and the 0s mark the host portion. Therefore, the network ID portion of the IP address in the example is 24 bits, or the first three octets: **192.168.123**. The host portion is the last octet: 132. Putting these two pieces together and using bold red for the network ID, this IP address is written as **192.168.123**.132.

By convention, you’ll see 0s used to complete the four octets when referring to the network ID and the host portion of an IP address separately, like this:

* Network ID: **192.168.123**.0
* Host portion: 0.0.0.132

Now suppose this computer needs to communicate with a host at **192.168.30**.140. Because the network IDs don’t match (that is, **192.168.123** does not match **192.168.30**), the computer knows the remote host is not on its own network and sends the transmission directly to its default gateway.

**Applying Concepts 8-2**

### Use the Logical ANDing Function to Calculate a Network ID

To calculate a host’s network ID given its IPv4 address and subnet mask, computers follow a logical process of combining bits known as [**ANDing**](javascript://). In ANDing, a bit with a value of 1 combined, or ANDed, with another bit with a value of 1 results in a 1. A bit with a value of 0 ANDed with any other bit results in a 0. If you think of 1 as “true” and 0 as “false,” the logic of ANDing makes sense: ANDing a true statement to a true statement still results in a true statement. But ANDing a true statement to a false statement results in a false statement.

ANDing logic is demonstrated in [Table 8-1](javascript://), which provides every possible combination of having a 1 or 0 bit in an IPv4 address or subnet mask.

**Table 8-1**

### ANDing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IP address bit | 1 | 1 | 0 | 0 |
| Subnet mask bit | 1 | 0 | 1 | 0 |
| **Resulting bit** | **1** | **0** | **0** | **0** |

A sample IPv4 host address, its default subnet mask, and its network ID are shown in [Figure 8-6](javascript://) in both binary and dotted decimal notation. Notice that the address’s fourth octet could have been composed of any combination of 1s and 0s, and the network ID’s fourth octet would still result in all 0s.

**Figure 8-6**

Example of ANDing a host’s network ID

An example in which the I P address and a subnet mask are ANDed to figure out the host's network I D. The I P Address has the bits 1 1 0 0 0 0 0 0 dot 0 0 1 0 0 0 1 0 dot 0 1 0 1 1 0 0 1 dot 0 1 1 1 1 1 1 1. The I P address in decimal notation is 192 dot 34 dot 89 dot 127. The subnet mask has the bits 1 1 1 1 1 1 1 1 dot 1 1 1 1 1 1 1 1 dot 1 1 1 1 1 1 1 1 dot 0 0 0 0 0 0 0 0. The subnet mask in decimal notation is 255 dot 255 dot 255 dot 1. The I P address and subnet mask ANDed gives the bits 1 1 0 0 0 0 0 0 dot 0 0 1 0 0 0 1 0 dot 0 1 0 1 1 0 0 1 dot 0 0 0 0 0 0 0 0 which in decimal notation is 192 dot 34 dot 89 dot 0. This is the host's network I D.

Enlarge Image

[Figure 8-6](javascript://) shows how ANDing logic is applied to an IPv4 address plus a default subnet mask. It works the same way for networks that are subnetted with classless subnet masks, discussed later in this module.

### Legacy Networking: Classful Addressing in IPv4

Recall that every IPv4 address can be associated with a network class—A, B, C, D, or E (though class D and E addresses are reserved for special purposes). Classful addressing is the simplest type of subnetting and uses only whole octets for the network ID and host portions. In the earlier example of **192.168.123**.132, the network ID consists of three octets, and is, therefore, an example of classful addressing. [Table 8-2](javascript://) lists how the 32 bits are allocated with classful addressing for classes A, B, and C.

**Table 8-2**

### Classful Addressing Uses Whole Octets for the Network ID

| **Class** | **Network portion in bold red:** | **Bits in network ID** | **Bits in host portion** |
| --- | --- | --- | --- |
| A | **nnnnnnnn**.hhhhhhhh.hhhhhhhh.hhhhhhhh | 8 | 24 |
| B | **nnnnnnnn.nnnnnnnn**.hhhhhhhh.hhhhhhhh | 16 | 16 |
| C | **nnnnnnnn.nnnnnnnn.nnnnnnnn**.hhhhhhhh | 24 | 8 |

Enlarge Table

There are a couple of interesting mathematical patterns to notice about classful IPv4 addressing:

* The last octet of a classful network ID is always equal to 0 (and may have preceding octets equal to 0). For example, the network ID for a class A network might be **92**.0.0.0, and the network ID for a class B network might be **147.12**.0.0.
* A host cannot be assigned the same address as the network ID, which explains why the last octet of a host’s IP address is almost never 0.
* Each octet can range in value from 0 to 255—you’ll never use a number greater than 255 in any octet for any reason.

Although classful addressing rules no longer restrict addressing options on modern networks, you can still use the classes as a starting point for IPv4 subnet calculations.

Each network class is associated with a default subnet mask, as shown in [Table 8-3](javascript://). For example, by default, a class A address’s first octet (or 8 bits) represents network information. This means that if you work on a network whose hosts are configured with a subnet mask of **11111111** 00000000 00000000 00000000, or 255.0.0.0, you know that the network is likely using class A addresses (though not necessarily). What if you were to “borrow” some host bits to make more subnets on your network? This is where classless addressing—and the need for calculating subnets—comes in. The next section describes how to calculate IPv4 subnets and how to determine the range of usable host addresses on a subnet (that is, addresses used for networked and networking devices), as well as the subnet masks the host addresses use. Later in the section, you will learn how subnetting differs in IPv6.

**Table 8-3**

### Default IPv4 Subnet Masks

| **Network class** | **Default subnet mask (binary)** | **Number of bits used for network information** | **Default subnet mask (dotted decimal notation)** |
| --- | --- | --- | --- |
| A | **11111111** 00000000 00000000 00000000 | 8 | **255**.0.0.0 |
| B | **11111111 11111111** 00000000 00000000 | 16 | **255.255**.0.0 |
| C | **11111111 11111111 11111111** 00000000 | 24 | **255.255.255**.0 |

Enlarge Table

**Remember This…**

* List reasons for creating subnets on a network.
* Explain how subnet masks work.
* Memorize the default subnet masks for class A, B, and C networks.

**Self-Check**

1. Which of the following is not a benefit of subnetting?

Answer

* 1. Problems are easier to locate and resolve.
  2. Available IP address space is managed more efficiently.
  3. Network documentation is easier to manage.
  4. Routers more easily manage IP address spaces that overlap.

1. What is the network ID of the IP address 192.168.72.149/16?

Answer

* 1. 0.0.0.149
  2. 192.168.0.0
  3. 0.0.72.149
  4. 192.168.72.0

**You’re Ready**

You’re now ready to complete [Project 8-1: Test Subnet Boundaries in Packet Tracer](javascript://), or you can wait until you’ve finished reading this module.

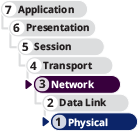
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 8-3Calculating Subnets



### Certification

* 1.4

Given a scenario, configure a subnet and use appropriate IP addressing schemes.

* 1.6

Explain the use and purpose of network services.

Average reading time: 49 minutes

Subnetting, which alters the rules of classful IPv4 addressing, is called classless addressing. To subnet a network, you borrow bits that would represent host information in classful addressing and use those bits instead to represent network information. By doing so, you increase the number of bits available for the network ID, and you also reduce the number of bits available for identifying hosts. Consequently, you increase the number of networks and reduce the number of usable host addresses in each network or subnet. The more bits you borrow for network information, the more subnets you can have, but the fewer hosts each subnet will contain. In many cases, this is actually the goal of subnetting.

Consider the following example of a network that initially has only a single class C subnet and all its hosts (up to 254) are on the same subnet. By borrowing only one bit from the host ID, you can have two subnets with up to 126 hosts (a more manageable number) per subnet. Let’s see how this works.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3aIPv4 Subnet Calculation in Binary

Suppose you have a network with one router and a switch connected to one interface on the router with your devices connected to that one switch. You then add a second switch to a different interface on the router to divide your local network into two LANs—you can then redistribute your network devices so some of them stay connected to the LAN on the first switch, and others are moved to the new LAN on the second switch.

**Note 8-3**

It’s not the addition of the second switch that creates the second LAN—it’s the use of the second interface on the router. If you instead daisy-chained the second switch off the first switch so that both switches were using the same router interface, you would still have only one LAN. In a project at the end of this module, you experiment with the boundaries of subnets and broadcast domains to see how this works.

The network ID of the original network is 192.168.89.0, and its subnet mask is 255.255.255.0. Let’s create two subnets of IP addresses, one for each LAN. The results of each of the following steps are shown later in [Table 8-4](javascript://):

1. **Borrow from host bits**—Currently, the network ID is 24 bits. First convert the network ID to binary:
   * Network ID 192.168.89 in binary:
     + **11000000.10101000.01011001**

Borrow one bit from the host portion to give to the network ID, which will then have 25 bits (notice one additional red bit in the fourth octet). Here, the borrowed bit (underlined) is an “x” to show it can have a value of 0 or 1:

* + **11000000.10101000.01011001.x**

How many subnets do you have now? The underlined red bit can be either a 0 or a 1, which gives you the possibility of two subnets:

* + **11000000.10101000.01011001.0**
  + **11000000.10101000.01011001.1**

1. **Determine the subnet mask**—Recall that the subnet mask marks the bits in an IP address that belong to the network ID. Therefore, the subnet mask for both subnets is as follows:
   * 11111111.11111111.11111111.10000000 or decimal 255.255.255.128

To calculate that last octet, you convert binary 10000000 to decimal, which is 128. You can use a calculator to do the conversion, manually calculate it, or memorize this and a handful of other common binary-to-decimal conversions.

1. **Determine the network IDs**—Recall that in the network ID, the underlined red bit can be a 0 or 1. Therefore, the network ID for each subnet is as follows:
   * Subnet 1: **11000000.10101000.01011001.0**0000000 or decimal 192.168.89.0
   * Subnet 2: **11000000.10101000.01011001.1**0000000 or decimal 192.168.89.128

In CIDR notation, the network ID for each subnet is the following:

* + Subnet 1: 192.168.89.0/25
  + Subnet 2: 192.168.89.128/25

1. **Determine the ranges of IP addresses for hosts in the subnet**—Start with the range of available IP addresses for subnet A. For host addresses, use the last seven bits in the last octet. (The first bit for this octet is always 0 and belongs to the network ID.) Start counting in binary and convert to decimal:
   * **0**0000000 is not used because it’s the network ID for this subnet
   * **0**0000001 or decimal 1
   * **0**0000010 or decimal 2
   * **0**0000011 or decimal 3
   * …
   * **0**1111110 or decimal 126
   * **0**1111111 or decimal 127, which is reserved for broadcasting within the subnet and cannot be used as a host address

The range of possible IP addresses is 192.168.89.0 through 192.168.89.127 (which is 128 possibilities). However, the first and last addresses cannot be used. Therefore, the range of host IP addresses for subnet A is 192.168.89.1 through 192.168.89.126 (that’s 128-2, yielding 126 possibilities).

For subnet B, the first bit of the last octet is 1 and the range of host addresses is as follows:

* + **1**0000000 is not used because it’s the network ID for this subnet
  + **1**0000001 in decimal: 129
  + **1**0000010 in decimal: 130
  + **1**0000011 in decimal: 131
  + …
  + **1**1111110 in decimal: 254
  + **1**1111111 in decimal: 255 is not used because it’s reserved for broadcasting

Therefore, the range of host IP addresses for subnet B is 192.168.89.129 through 192.168.89.254 (another 126 possibilities).

**Table 8-4**

### Steps to Divide IP Addresses for Network ID 192.168.89.0 into Two Subnets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step 1: Borrow from host bits.** |  |  |  |  |
| **Network ID** | **192** | **168** | **89** | 0 |
| In binary | **11000000** | **10101000** | **01011001** | 00000000 |
| Borrow 1 bit | **11000000** | **10101000** | **01011001** | **x**0000000 |
| **Step 2: Determine the subnet mask.** | | | | |
| In binary | **11111111** | **11111111** | **11111111** | **1**0000000 |
| In decimal | **255** | **255** | **255** | **128** |
| **Step 3: Determine the network IDs.** | | | | |
| **Network ID 1** | **11000000** | **10101000** | **01011001** | **0**0000000 |
| In decimal | **192** | **168** | **89** | **0** |
| In CIDR notation | 192.168.89.0/25 | | | |
| **Network ID 2** | **11000000** | **10101000** | **01011001** | **1**0000000 |
| In decimal | **192** | **168** | **89** | **128** |
| In CIDR notation | 192.168.89.128/25 |  |  |  |
| **Step 4: Determine range of host IP addresses.** | | | | |
| **Subnet 1:** | | | | |
| First host, binary | **11000000** | **10101000** | **01011001** | **0**00000001 |
| First host, decimal | 192 | 168 | 89 | 1 |
| Last host, binary | **11000000** | **10101000** | **01011001** | **0**1111110 |
| Last host, decimal | 192 | 168 | 89 | 126 |
| **Subnet 2:** | | | | |
| First host, binary | **11000000** | **10101000** | **01011001** | **1**0000001 |
| First host, decimal | 192 | 168 | 89 | 129 |
| Last host, binary | **11000000** | **10101000** | **01011001** | **1**1111110 |
| Last host, decimal | 192 | 168 | 89 | 254 |

Enlarge Table

Notice that you gained a network (the larger network became two smaller subnets), but you lost some host addresses (each subnet offers a possible 126 host addresses for a total of 252 hosts, which is two fewer than the original network’s possible 254 hosts). The difference is because each subnet needs its own network ID and broadcast address. Instead of the original network’s one network ID and one broadcast address, you now have two network IDs and two broadcast addresses, which reduces the total number of available host addresses. Subnetting offers many advantages, but one disadvantage is that you lose possible host addresses each time you divide a network into more subnets.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3bIPv4 Subnet Calculations Using Formulas

Now you’re ready to move on to a more complicated example, performing calculations using formulas and without so much binary involved. Suppose you want to divide your local network, which has a network ID of 192.168.89.0, into six subnets to correspond to your building’s six floors. The following steps walk you through the process:

1. 1

**Decide how many bits to borrow**—How many bits must you borrow from the host portion of the IP addresses to get six subnets? Use this formula to determine the number of bits:

* + n equals the number of bits that must be switched from the host address to the network ID.
  + Y equals the number of subnets that result.

You want six separate subnets (meaning that Y, in this case, should be equal to or greater than 6). Experiment with different values for n until you find a value large enough to give you at least the number of subnets you need. For example, you know that ; however, 4 is not high enough. Instead consider that ; this will give you enough subnets to meet your current needs and allow room for future growth. Now that n equals 3, you know that three bits in the host addresses of your class C network must become network ID bits. You also know that three bits in your subnet mask must change from 0 to 1.

1. 2

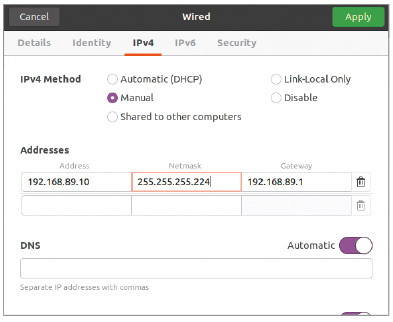
**Determine the subnet mask**—As you know, the default subnet mask for a class C network is **255.255.255**.0, or **11111111 11111111 11111111** 00000000. In this default subnet mask, the first 24 bits indicate the position of network information.

Changing three of the default subnet mask’s bits from host to network information gives you the subnet mask **11111111 11111111 11111111 111**00000. In this modified subnet mask, the first 27 bits indicate the bits for the network ID. For the original class C network whose network ID is 192.168.89.0, the slash notation for its first subnet would now be 192.168.89.0/27 because 27 bits of the subnet’s address are used to provide network information.

Converting from binary to the more familiar dotted decimal notation, this subnet mask becomes 255.255.255.224 because 11100000 in binary is 224 in decimal for the final octet. When you configure the TCP/IP properties of clients on your network, as shown in [Figure 8-7](javascript://), you’ll specify this new subnet mask.

**Figure 8-7**

IPv4 client configuration for a subnet in Ubuntu Desktop



Source: Canonical Group Limited

**Note 8-4**

When examining the subnet mask for a network, if any octet is not 255 or 0, you know that this network is a subnet and classful addressing is not used. The unusual number (224 in the example) is often called the interesting octet. Subtract the interesting octet value from 256 and you get what is called the [**magic number**](javascript://). In this example, the magic number is . This magic number can be used to calculate the network IDs in all the subnets of the larger network, which you’ll see next.

1. 3

**Calculate the network ID for each subnet**—The first three octets of the network ID for the original class C network (192.168.89) are the same for all eight possible subnets. The network IDs differ in the last octet. Use the magic number to calculate each subnet’s network ID as follows:

* + **Subnet 1**: **192.168.89.0**
  + **Subnet 2**: 192.168.89.0 + 32 yields **192.168.89.32**
  + **Subnet 3**: 192.168.89.32 + 32 yields **192.168.89.64**
  + **Subnet 4**: 192.168.89.64 + 32 yields **192.168.89.96**
  + **Subnet 5**: 192.168.89.96 + 32 yields **192.168.89.128**
  + **Subnet 6**: 192.168.89.128 + 32 yields **192.168.89.160**
  + **Subnet 7**: 192.168.89.160 + 32 yields **192.168.89.192**
  + **Subnet 8**: 192.168.89.192 + 32 yields **192.168.89.224**

This method of adding on the same number over and over is called skip-counting. You probably learned this technique in elementary school. For example, skip-counting by twos gives you 0, 2, 4, 6, 8, 10, etc. Skip-counting by threes gives 0, 3, 6, 9, 12, etc. In the last octet, you can skip-count by 32 to get all eight subnets’ final octets: 0, 32, 64, 96, 128, 160, 192, and 224.

1. 4

**Determine the IP address range for hosts in each subnet**—Recall that you have borrowed three bits from what used to be host information in the IP address. That leaves five bits instead of eight available in the last octet of your class C addresses to identify hosts. To calculate the number of possible addresses, keep in mind that each of the five bits has two possible values, a 0 or a 1. Therefore, the number of possible addresses is . But you can’t use two of these addresses for hosts because one is used for the network ID (the one where all five bits are 0 in binary) and one for the broadcast address (the one where all five bits are 1 in binary). That leaves you 30 host addresses in each subnet. As a shortcut to calculating the number of hosts, you can use the following formula:

* + h equals the number of bits remaining in the host portion.
  + Z equals the number of hosts available in each subnet.

In summary,  yields 30 possible hosts per subnet.

In this example, you can have a maximum of 8 (number of subnets) × 30 (number of hosts per subnet), or 240, unique host addresses on the entire, larger network. Recall that each time you subnet a network, you lose two possible host addresses with each subnet. This overhead is the price you pay for subnetting a network, in exchange for the advantages you gain.

When calculating subnets, you’ll work with the following information, some of which is initially known and some of which must be calculated:

* Number of subnets
* Number of host addresses per subnet
* Network ID for each subnet
* Broadcast address for each subnet
* Range of possible host addresses within each subnet

In this example, you’ve already calculated the number of subnets, number of host addresses, and each subnet’s network ID. Once you know the network ID of the subnets, calculating the address range of hosts and each subnet’s broadcast address is relatively simple. For example, take subnet 5. The network ID is 192.168.89.128. The broadcast address is the last address before the next subnet’s network ID. In this case, that’s 192.168.89.160 – 1, which is 192.168.89.159.

For the host addresses in this subnet, start back at the network ID. You won’t use the network ID for a host address, so you start with the next value and keep going until you reach the broadcast address for the subnet. Therefore, the available host address range for subnet 5 includes 192.168.89.129 through 192.168.89.158, yielding for this particular subnet a total of 30 available host addresses.

[Table 8-5](javascript://) lists the network ID, broadcast address, and the range of usable host addresses for each of the eight subnets in this subnetted class C network. Together, the existing network ID plus the additional bits used for subnet information are sometimes called the extended network prefix.

**Table 8-5**

### Subnet Information for Eight Subnets in a Sample IPv4 Class C Network

| **Subnet number** | **Network ID (extended network prefix)** | **Range of host addresses** | **Broadcast address** |
| --- | --- | --- | --- |
| 1 | 192.168.89.0 or**11000000 10101000 01011001 000**00000 | 192.168.89.1-30 | 192.168.89.31 or **11000000 10101000 01011001 000**11111 |
| 2 | 192.168.89.32 or **11000000 10101000 01011001 001**00000 | 192.168.89.33-62 | 192.168.89.63 or **11000000 10101000 01011001 001**11111 |
| 3 | 192.168.89.64 or **11000000 10101000 01011001 010**00000 | 192.168.89.65-94 | 192.168.89.95 or **11000000 10101000 01011001 010**11111 |
| 4 | 192.168.89.96 or **11000000 10101000 01011001 011**00000 | 192.168.89.97-126 | 192.168.89.127 or **11000000 10101000 01011001 011**11111 |
| 5 | 192.168.89.128 or **11000000 10101000 01011001 100**00000 | 192.168.89.129-158 | 192.168.89.159 or **11000000 10101000 01011001 100**11111 |
| 6 | 192.168.89.160 or **11000000 10101000 01011001 101**00000 | 192.168.89.161-190 | 192.168.89.191 or **11000000 10101000 01011001 101**11111 |
| 7 | 192.168.89.192 or **11000000 10101000 01011001 110**00000 | 192.168.89.193-222 | 192.168.89.223 or **11000000 10101000 01011001 110**11111 |
| 8 | 192.168.89.224 or **11000000 10101000 01011001 111**00000 | 192.168.89.225-254 | 192.168.89.255 or **11000000 10101000 01011001 111**11111 |

Enlarge Table

**Note 8-5**

You can also calculate the magic number by raising 2 to the power of the number of bits in the host portion of the subnet mask’s interesting octet. Use this formula:

In this example, the host portion has five bits, meaning **h** equals 5. Therefore, the magic number is . You can then use this number to determine the subnets’ network IDs.

Several websites provide excellent tools that can help you calculate subnet information or to check your calculations after performing them manually. Two such sites are [subnetmask.info](http://subnetmask.info/" \t "_blank) and cidr.xyz. Other websites and apps can give you practice calculating subnets in preparation for your certification exams. Check out the website [subnettingquestions.com](http://subnettingquestions.com/" \t "_blank) in your browser or the Subnetting Practice app with the “/24” icon on both Android and iPhone.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3cSubnet Mask Tables

Class A, class B, and class C networks can all be subnetted. But because each class reserves a different number of bits for network information, each class has a different number of host information bits that can be used for subnet information. The number of hosts and subnets on your network will vary depending on your network class and the way you use subnetting.

[Table 8-6](javascript://) lists the numbers of subnets and hosts that can be created by subnetting a class B network. Notice the available subnet masks that can be used instead of the default class B subnet mask of 255.255.0.0. Also compare the listed numbers of hosts per subnet to the 65,534 hosts available on a class B network that is not subnetted.

**Table 8-6**

### IPv4 Class B Subnet Masks

| **Subnet mask** | **CIDR block** | **Number of subnets on network** | **Number of hosts per subnet** |
| --- | --- | --- | --- |
| 255.255.128.0 or 11111111 11111111 10000000 00000000 | /17 | 2 | 32,766 |
| 255.255.192.0 or 11111111 11111111 11000000 00000000 | /18 | 4 | 16,382 |
| 255.255.224.0 or 11111111 11111111 11100000 00000000 | /19 | 8 | 8190 |
| 255.255.240.0 or 11111111 11111111 11110000 00000000 | /20 | 16 | 4094 |
| 255.255.248.0 or 11111111 11111111 11111000 00000000 | /21 | 32 | 2046 |
| 255.255.252.0 or 11111111 11111111 11111100 00000000 | /22 | 64 | 1022 |
| 255.255.254.0 or 11111111 11111111 11111110 00000000 | /23 | 128 | 510 |
| 255.255.255.0 or 11111111 11111111 11111111 00000000 | /24 | 256 | 254 |
| 255.255.255.128 or 11111111 11111111 11111111 10000000 | /25 | 512 | 126 |
| 255.255.255.192 or 11111111 11111111 11111111 11000000 | /26 | 1024 | 62 |
| 255.255.255.224 or 11111111 11111111 11111111 11100000 | /27 | 2048 | 30 |
| 255.255.255.240 or 11111111 11111111 11111111 11110000 | /28 | 4096 | 14 |
| 255.255.255.248 or 11111111 11111111 11111111 11111000 | /29 | 8192 | 6 |
| 255.255.255.252 or 11111111 11111111 11111111 11111100 | /30 | 16,384 | 2 |

Enlarge Table

[Table 8-7](javascript://) lists the numbers of subnets and hosts that can be created by subnetting a class C network. Notice that a class C network allows for fewer subnets than a class B network. This is because class C addresses have fewer host information bits that can be borrowed for network information. In addition, fewer bits are left over for host information, which leads to a lower number of hosts per subnet than the number available to class B subnets.

**Table 8-7**

### IPv4 Class C Subnet Masks

| **Subnet mask** | **CIDR block** | **Number of subnets on network** | **Number of hosts per subnet** |
| --- | --- | --- | --- |
| 255.255.255.128 or 11111111 11111111 11111111 10000000 | /25 | 2 | 126 |
| 255.255.255.192 or 11111111 11111111 11111111 11000000 | /26 | 4 | 62 |
| 255.255.255.224 or 11111111 11111111 11111111 11100000 | /27 | 8 | 30 |
| 255.255.255.240 or 11111111 11111111 11111111 11110000 | /28 | 16 | 14 |
| 255.255.255.248 or 11111111 11111111 11111111 11111000 | /29 | 32 | 6 |
| 255.255.255.252 or 11111111 11111111 11111111 11111100 | /30 | 64 | 2 |

Enlarge Table

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3dSubnetting Questions on Exams

Although it’s impossible to know for sure, you’re likely to encounter two types of subnet calculation problems on the CompTIA Network+ exam:

* Given certain network requirements (such as required number of hosts or required number of subnets), calculate possible subnets and host IP address ranges.
* Given an IP address, determine its subnet’s network ID, broadcast address, and first/last host addresses.

You’ve already seen steps for solving the first type of problem using a class C network ID. Now you’re ready to work through another example of the same type of problem, but this time you’ll begin with a class B network ID. Then you’ll work through an example of the second type of subnetting problem.

**Applying Concepts 8-3**

### Calculate IPv4 Subnets and Host IP Address Ranges

Suppose your organization uses the class B network ID of 172.20.0.0 for its entire network and wants to create 15 subnets. Complete the following steps, answering the questions as you go:

1. 1

You first need to decide how many bits to borrow from the host address bits. Recall that you can use the formula . To get at least 15 new subnets (without creating more subnets than necessary), how many bits must be borrowed from the host address portion? How many bits total will be used for identifying a host’s subnet?

1. 2

You can now calculate the subnet mask. The default subnet mask for a class B network is 255.255.0.0, and so the third octet is the one that will change. What is the subnet mask for these subnets, written in dotted decimal notation?

1. 3

The magic number will tell you by what amount to skip-count when you’re listing the subnets’ network IDs. There are two ways to calculate the magic number: Subtract the interesting octet’s value from 256, or use the formula  where h equals the number of host bits in the interesting octet. What is the magic number you can use to calculate the network IDs?

1. 4

Now you can calculate the network IDs for each subnet. Begin with the original network ID. Then in the third octet, count up by the magic number with each iteration. The last subnet’s network ID will be equal to 256 minus the magic number, because you can’t use 256 itself in any IP address. What is the CIDR notation for the first subnet’s network ID? For the second subnet’s network ID? For the last subnet’s network ID?

1. 5

If 20 bits are used to identify the network and subnet, that leaves 12 bits to identify each host. Using the formula  where h equals the total number of host bits, how many host addresses are possible in each subnet? (You might need a calculator for this step, such as the Windows 10 Calculator app in Scientific mode. Note that it’s unusual to see an exam question where you’re working with bits in the third octet because you can’t use a calculator during the exam.)

1. 6

The range of available host addresses consists of all the possible IP addresses between the network ID and the broadcast address (which is one below the network ID for the next subnet). What is the range of host addresses for the first subnet? For the second subnet? For the last subnet?

Now you’re ready to work through the other type of subnetting problem that you’ll likely see on your CompTIA Network+ exam: calculating a host’s network information.

**Applying Concepts 8-4**

### Calculate an IPv4 Host’s Network Information

This time, you’ll work backwards in your calculations by starting with one host’s IP address information. Suppose a server on your network displays the following IPv4 network configuration:

Your task is to determine the network ID of the subnet this server is located on, the broadcast address, and the range of available host addresses on this subnet. Complete the following steps, answering the questions as you go:

1. 1

You don’t necessarily need to use binary for these calculations. Find the magic number and go from there. As with your earlier calculations, you need to subtract the interesting octet’s value from 256 to get the magic number. What is the magic number?

1. 2

If the interesting octet is located at the end of the subnet mask, you can assume the first three octets of the IP address identify the classful network ID before this network was subnetted. This network ID also serves as the network ID for the first subnet. What is the network ID of the first subnet?

1. 3

You can now use the magic number to calculate the remaining subnets’ network IDs. What is the second subnet’s network ID? What is the final subnet’s network ID?

1. 4

To narrow this down to your server’s subnet, either skip-count up from a lower numbered subnet or skip-count down from a higher numbered subnet. Either way will work. You’re looking for a network ID that is as close to the server’s IP address as possible without going over. What is the network ID of the server’s subnet?

1. 5

You can look at the next higher subnet’s network ID and subtract 1 to determine the broadcast address of the server’s subnet. What is the broadcast address?

1. 6

Finally, any IP address between the subnet’s network ID and its broadcast address is the range of available host IP addresses. What is this range?

### Exam Tip

If these calculation processes seem overwhelming, you’re not alone. Many people have developed a variety of handy shortcuts for calculating subnets. In a Hands-On Project at the end of this module, you’ll see how to use a shortcut to answer each of the two primary types of subnetting exam questions. If these shortcuts don’t resonate with you, you can search online to find a shortcut that does click. There are many options, and sometimes it’s just a matter of finding the approach that is easiest for you to remember and work with.

Go to pg.

[**help**](javascript://)

Application Opened

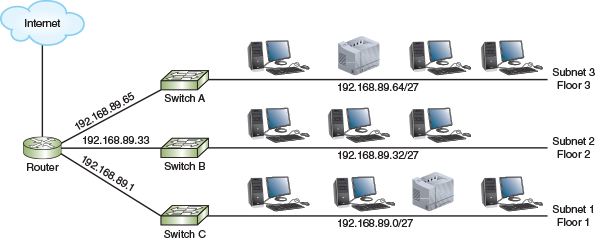
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3eImplementing Subnets on a Network

Now that you’ve calculated the subnets for the scenarios presented earlier in the module, how do you implement them? [Figure 8-8](javascript://) shows the subnets assigned to the three LANs you saw earlier in [Figure 8-4](javascript://). Also in [Figure 8-8](javascript://), you can see the IP address of the default gateway for each LAN, which is the IP address assigned to the router’s interface on that LAN. Note that only three of the eight possible subnets listed earlier in [Table 8-5](javascript://) are used.

**Figure 8-8**

Subnets 1, 2, and 3 and their respective default gateways

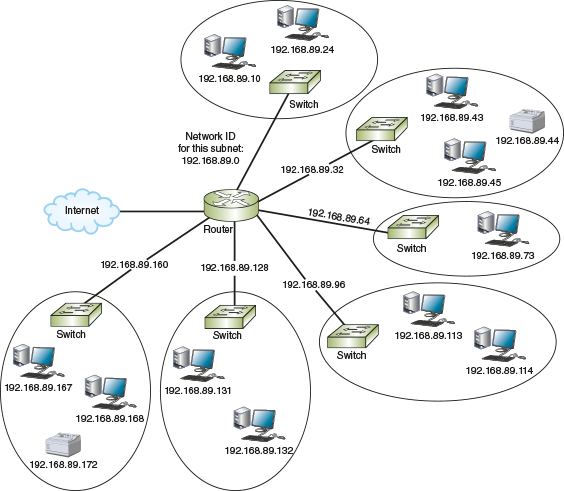


Enlarge Image

[Figure 8-9](javascript://) illustrates another scenario in which an enterprise network uses the same class C range of private addresses that begin with 192.168.89. The network administrator has subnetted this class C network into six (of eight possible) smaller networks.

**Figure 8-9**

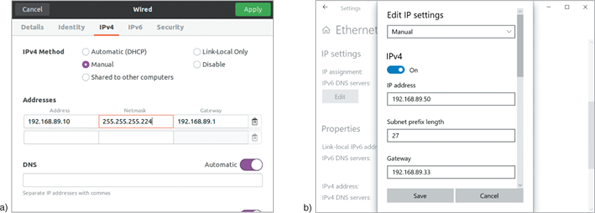
One router connecting several LANs, each assigned a subnet



The administrator must program each interface on the router with its IP address and subnet mask for its subnet. Though tedious on larger networks, static IP addressing can also be used on network hosts. [Figure 8-10](javascript://)a shows the TCP/IPv4 properties dialog box of an Ubuntu workstation on the first subnet. [Figure 8-10](javascript://)b shows the static configuration for a Windows workstation on the second subnet. As shown in the figure, the first IP address in the range of host addresses for the subnet is assigned to the router’s interface on the subnet, which serves the subnet as its default gateway. This convention varies between organizations, though. Some network admins prefer to use the last available host address in a range for the default gateway.

**Figure 8-10**

Static IP configurations for workstations on two subnets



Enlarge Image

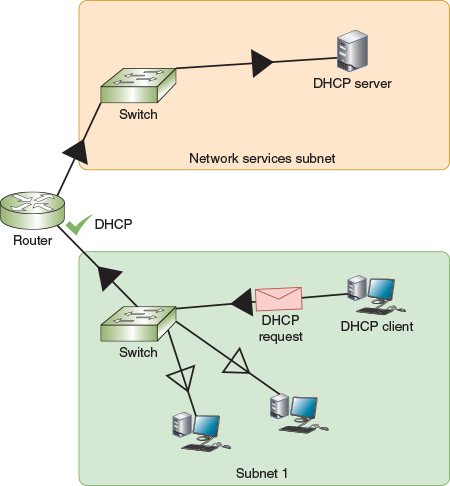
Source: Canonical Group Limited

For dynamic IP addressing, the administrator programs each subnet’s DHCP server with the network ID, subnet mask, range of IP addresses, and default gateway for the subnet. In many cases, however, it’s cost prohibitive to create a separate DHCP server for each subnet. Recall that DHCP relies on broadcast transmissions to function properly, and broadcast messages are bound by routers. However, there are times you want some types of broadcast traffic, such as DHCP messages, to travel beyond each broadcast domain. This allows hosts in various subnets to access centralized network services. Recall that DHCP relies on UDP at the transport layer. Other centralized network services also use UDP traffic, including TFTP, NTP, and DNS. Configuring [**UDP forwarding**](javascript://) on your network allows routers, firewalls, or layer 3 switches to forward this UDP traffic across broadcast domains, which enables centralization of key network services. How does this work?

As illustrated in [Figure 8-11](javascript://), a LAN device sends a broadcast message on the local subnet intended for a local network server (such as DHCP). When the switch broadcasts this message to all its interfaces, a [**relay agent**](javascript://)—a router, firewall, or layer 3 switch configured to support UDP forwarding—detects the message according to a list of services that are enabled for forwarding (which might include DHCP, DNS, and NTP, for example). Forwarded network services are identified by port, such as port 67 for DHCP or port 123 for NTP. When one of these broadcast messages is detected on the local subnet, the relay agent forwards those messages to a designated [**IP helper address**](javascript://) on another network. This IP helper address might point specifically to a single server, such as when only DHCP traffic is forwarded. Or it might be the broadcast address for the other subnet so several network servers can monitor traffic for messages, such as when you have DHCP, NTP, and DNS all forwarded to a centralized network services subnet. The intended network server detects the message, collects the needed information from the request, and responds to the request as configured. The relay agent then routes the response back to the client in the original subnet.

**Figure 8-11**

The router detects the DHCP broadcast message and forwards the message to the DHCP server on a different subnet



For example, consider the scenario where a centrally managed DHCP server provides DHCP assignments to multiple subnets (and VLANs). The following steps describe this process:

1. Step 1

A DHCP relay agent programmed to support UDP forwarding on port 67 receives the DHCP request from a client in one of its local broadcast domains.

1. Step 2

The relay agent repackages the message with the IP helper information and routes this transmission to its new destination in a different broadcast domain.

1. Step 3

The DHCP server notes the relay agent’s source interface IP address and assigns the DHCP client an IP address on the same subnet.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

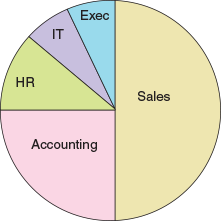
## 8-3fVariable Length Subnet Mask (VLSM)

Hosting your centralized network services, such as DHCP, within a single subnet that is accessed by all other subnets is one way to simplify network management and reduce management overhead. As you can imagine, however, this subnet of centralized network servers will have a lot fewer hosts than a subnet of a few dozen (or a few hundred) employees. Traditional subnetting results in multiple subnets that are all the same size, and this uniformity in subnet size can be inefficient in complex networks. [**VLSM (Variable Length Subnet Mask)**](javascript://) allows subnets to be further subdivided into smaller and smaller groupings until each subnet is about the same size as the necessary IP address space. This is often referred to as “subnetting a subnet.”

To understand how this works, consider a pizza being shared by members of a young family. Dad might need a very large slice of the pizza, while Mom prefers a medium slice, and the children each need smaller slices. Similarly, with VLSM, some subnets can have larger “slices” of the network, while other subnets (such as a two-point connection between two routers) can be limited to only a few host addresses. See [Figure 8-12](javascript://).

**Figure 8-12**

VLSM creates subnets of various sizes



To create VLSM subnets, you create the largest subnet first. Then you create the next largest subnet, and then the next one, and so on, until you have divided up all the remaining space. In this way, you ensure that the largest subnets get the space they need, and the smallest subnets are also sized appropriately. Let’s work through an example.

Suppose you need to configure the subnets shown in [Table 8-8](javascript://) using the 192.168.10.0/24 IP address space. The Sales department needs the greatest number of hosts. At the other end of the spectrum, your WAN links only need two hosts each. The other subnets fall somewhere in the middle.

**Table 8-8**

### Subnets of Various Sizes Needed on the Network

| **Subnet** | **Included hosts** | **Number of hosts** | **CIDR notation (as calculated next)** |
| --- | --- | --- | --- |
| 1 | Sales | 120 | 192.168.10.0 /25 |
| 2 | Accounting | 58 | 192.168.10.128 /26 |
| 3 | HR | 25 | 192.168.10.192 /27 |
| 4 | IT | 6 | 192.168.10.224 /29 |
| 5 | Executives | 5 | 192.168.10.232 /29 |
| 6 | WAN link | 2 | 192.168.10.240 /30 |
| 7 | WAN link | 2 | 192.168.10.244 /30 |

Enlarge Table

1. Step 1

Determine the appropriate subnet mask and other network information for the largest subnet. By borrowing one bit from the host bits, you get the following available subnets:

* + Subnet 1: 192.168.10.0 /25 assigned to Sales
  + Subnet 2: 192.168.10.128 /25 for smaller subnets

You assign the first of these subnets to the Sales department. Now you can use the second subnet for further calculations.

1. Step 2

Determine the appropriate subnet mask and other network information for the next largest subnet. By borrowing one more bit from the host bits, you get the following available subnets:

* + Subnet 2: 192.168.10.128 /26 assigned to Accounting
  + Subnet 3: 192.168.10.192 /26 for smaller subnets

You assign the first of these subnets to the Accounting department. Now you can use the remaining subnet for further calculations.

1. Step 3

Determine the appropriate subnet mask and other network information for the next largest subnet. By borrowing one more bit from the host bits, you get the following available subnets:

* + Subnet 3: 192.168.10.192 /27 assigned to HR
  + Subnet 4: 192.168.10.224 /27 for smaller subnets

You assign the first of these subnets to the Human Resources department. Now you can use the other subnet for further calculations.

1. Step 4

The next two departments are about the same size, and they’ll each fit within a /29 subnet. By borrowing two more bits from the host bits this time, you get the following available subnets:

* + Subnet 4: 192.168.10.224 /29 assigned to IT
  + Subnet 5: 192.168.10.232 /29 assigned to Executives
  + Subnet 6: 192.168.10.240 /29 for smaller subnets
  + Subnet 7: 192.168.10.248 /29 for future use

You assign the first two of these subnets to the IT department and the Executive suite. Now you can use one of the other subnets for further calculations.

1. Step 5

The last two required subnets only need two host addresses each, and they’ll each fit within a /30 subnet. By borrowing one more bit from the host bits to further subdivide Subnet 6, and renumbering the remaining space to be Subnet 8 (which will be reserved for future use on your network), you get the following available subnets:

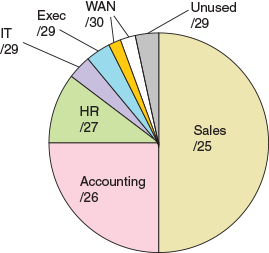
* + Subnet 6: 192.168.10.240 /30 assigned to WAN link
  + Subnet 7: 192.168.10.244 /30 assigned to WAN link
  + Subnet 8: 192.168.10.248 /29 for future use

You assign each of these subnets to a WAN link, with the final subnet left over for future use.

[Figure 8-13](javascript://) shows the mathematically determined distribution, with each department allocated the IP address space it needs.

**Figure 8-13**

Actual subnet allocations



This is an efficient way to define IP address spaces on a network. However, in reality, it’s not a good idea to configure subnets so tightly. In this case, for example, there’s very little room for future growth. Most companies should allow for significant growth, especially as technology continues to expand the need for IP addresses on a network.

One way to prepare for this growth is to begin with a larger IP address space. For example, you might start with a /23 or even a /22 network. Then subdivide from there, giving each subnet significantly more host addresses than it currently needs. This works for private IP addresses, but not so much for public IP addresses. Another way to account for future growth is to convert the network to IPv6 addressing instead of IPv4. Let’s look at how IPv6 subnetting works.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-3gSubnets in IPv6

Recall that IPv6 addresses are composed of 128 bits, compared with IPv4’s 32-bit addresses. That means many more addresses are available in IPv6, compared with IPv4’s available addresses. Given so many addresses, an ISP can offer each of its customers an entire IPv6 subnet, or thousands of addresses, rather than a handful of IPv4 addresses that must be shared among all the company’s nodes. In this case, subnetting helps network administrators manage the enormous volume of IPv6 addresses.

Subnetting in IPv6 is simpler than subnetting in IPv4, and it differs from IPv4 in substantial ways:

* IPv6 addressing uses no classes. There are no IPv6 equivalents to IPv4’s class A, class B, or class C networks. Every IPv6 address is classless.
* IPv6 does not use subnet masks.
* A single IPv6 subnet can supply 18,446,744,073,709,551,616 IPv6 addresses.

Let’s see how these numbers pan out. Recall that a unicast address is an address assigned to a single interface on the network. Also recall that every unicast address can be represented in binary form, but it’s more commonly written as eight blocks of four hexadecimal characters separated by colons. For example, 2608:FE10:1:AA:002:50FF:FE2B:E708 is a valid IPv6 address. As shown in [Figure 8-14](javascript://), let’s divide that address into parts:

**Figure 8-14**

Network prefix and interface ID in an IPv6 address

An I P v 6 address is made up of a network prefix and an interface I D each occupying 64 bits. 2608 colon F E 10 colon 1 colon A A is a network prefix. 0 0 2 colon 50 F F colon F E 2 B colon E 708 is an interface I D.

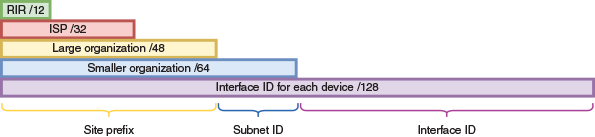
Enlarge Image

* The last four blocks, which equate to the last 64 bits, identify the interface. (On many IPv6 networks, those 64 bits are based on the interface’s EUI-64 version of each device’s MAC address.)
* The first four blocks or 64 bits normally identify the network and serve as the network prefix, also called the [**site prefix**](javascript://) or [**global routing prefix**](javascript://). In the IPv6 address **2608:FE10:1:AA**:002:50FF:FE2B:E708, the site prefix is **2608:FE10:1:AA** and the interface ID is 002:50FF:FE2B:E708. You might see site prefixes represented as, for example, 2608:FE10:1:AA::/64, where the number of bits that identify the network follow a slash.
* The fourth hexadecimal block in the site prefix can be altered to create subnets within a site. Let’s take a closer look at how that block fits into the big picture.

As shown in [Figure 8-15](javascript://), an RIR (regional Internet registry) might assign an ISP a block of addresses that share a 32-bit routing prefix, such as 2608:FE10::/32. That ISP, in turn, might assign a very large organization a block of addresses that share the same 48-bit site prefix, such as 2608:FE10:1::/48, and smaller business customers might receive a 56-bit site prefix, such as 2608:FE10:1:AA::/56, or a 64-bit site prefix, such as 2608:FE10:1:AA::/64.

**Figure 8-15**

Hierarchy of IPv6 routes and subnets



Enlarge Image

The subnet ID is one block long, which is four hexadecimal characters, or 16 bits in binary. An organization with a /48 site prefix can use all 16 bits to create up to 65,536 subnets. A /56 site prefix can create up to 256 subnets, and a /64 site prefix has only the single subnet, which contains over 18 quintillion possible host addresses (this is more than twice the estimated number of grains of sand in all the beaches and deserts of the earth). As you can see, IPv6 allows for a huge number of potential hosts on a single network.

Consider your sample network with a site prefix of 2608:FE10:1/48 and see what happens with the next block of bits at a binary level. In binary, that fourth block, the Subnet ID, could be all zeroes:

* 0000 0000 0000 0000

Or it could be all 1s:

* 1111 1111 1111 1111

And then there’s every possible combination in between:

* 0000 0000 0000 0001
* 0000 0000 0000 0010
* 0000 0000 0000 0011
* 0000 0000 0000 0100
* …
* 1111 1111 1111 1100
* 1111 1111 1111 1101
* 1111 1111 1111 1110

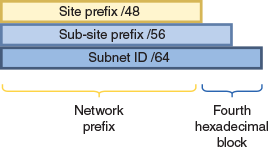
That’s 65,536 possible subnets. A sample network with a site prefix of 2608:FE10:1:AA/56 can work with eight of those bits to create 256 possible subnets:

* 0000 0000
* 0000 0001
* 0000 0010
* …
* 1111 1101
* 1111 1110
* 1111 1111

Sometimes organizations further subdivide this block into site, sub-site, and subnet IDs. For example, consider [Figure 8-16](javascript://), where the Subnet ID block is managed at two levels: the first half for subsites (such as offices in different states or different cities) and the second half for subnets within each site (such as floors in a building or departments located at each site).

**Figure 8-16**

The Subnet ID block can be used to identify subsites within an organization



Calculating subnets can feel overwhelming when you’re first learning how to work with these numbers. The key here is practice. Find a subnet calculation system that works for you, and then practice often. Rest assured that you don’t have to be a “math” person to learn these skills.

Now that you have learned how subnets manage IP address spaces at the network layer, you’re ready to explore network segmentation at the data link layer: VLANs.

**Remember This…**

* Calculate IPv4 subnets.
* Explain how to implement VLSM.
* Use CIDR notation.
* Describe UDP forwarding, IP helper addresses, and DHCP relay.

**Self-Check**

1. If a server has a subnet mask of 255.255.255.224, how many bits in its IP address identify the host?

Answer

* 1. 27 bits
  2. 8 bits
  3. 30 bits
  4. 5 bits

1. What is the minimum number of bits that should be borrowed to create 14 subnets?

Answer

* 1. 4 bits
  2. 2 bits
  3. 16 bits
  4. 8 bits

1. Suppose you’re calculating the range of host IP addresses for a subnet (the targeted subnet). If the next subnet’s network ID is 192.168.42.128, what is the targeted subnet’s broadcast address?

Answer

* 1. 192.168.42.96
  2. 192.168.42.0
  3. 192.168.42.127
  4. 192.168.42.255

**You’re Ready**

You’re now ready to complete [Project 8-2: Calculate Subnets](javascript://), or you can wait until you’ve finished reading this module.

**You’re Ready**

You’re now ready to complete [Project 8-3: Shortcuts to Subnet Calculations](javascript://), or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 8-4Virtual LANs (VLANs)

### Certification

* 1.4

Given a scenario, configure a subnet and use appropriate IP addressing schemes.

* 2.3

Given a scenario, configure and deploy common Ethernet switching features.

* 4.1

Explain common security concepts.

* 4.2

Compare and contrast common types of attacks.

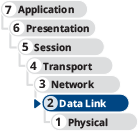
* 4.3

Given a scenario, apply network hardening techniques.

* 5.5

Given a scenario, troubleshoot general networking issues.

Average reading time: 28 minutes



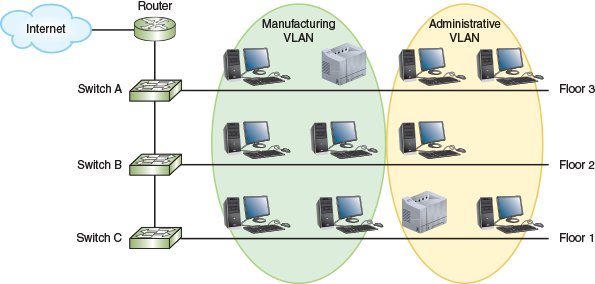
Let’s begin with a discussion about the similarities and differences between subnets and VLANs. As you’ve learned, a subnet groups IP addresses so that clients on a large network can be logically organized into smaller networks. As you’ve also seen, this is often accomplished by using multiple ports on a router, which creates multiple broadcast domains within the larger network with subnets organizing the available IP address space.

By contrast, a [**VLAN (virtual local area network or virtual LAN)**](javascript://) groups ports on one or more switches so that some of the local traffic on each switch is forced to go through a router, thereby limiting the traffic to a smaller broadcast domain. As virtual LANs, VLANs abstract the broadcast domain from the networking hardware. This is similar to how VMs abstract computing functions from a computer’s hardware. When using VLANs, the boundaries of the broadcast domain can be virtually defined anywhere within a single physical LAN.

Recall the example given earlier in this module of a large network on three floors of a building, as shown earlier in [Figure 8-3](javascript://). Rather than running each switch’s connection all the way to the router so you can connect each switch to a different router interface, you could instead use managed switches and VLANs to segment the network. For example, suppose you segment the network by department in the company rather than by floors in the building, as shown in [Figure 8-17](javascript://).

**Figure 8-17**

A simple VLAN design



Enlarge Image

To do this, you might need to install managed switches to replace the original switches; however, most modern switches have built-in VLAN functionality. You’ll then assign each host to a specific VLAN by configuring the switch port each host is connected to. Essentially, you’ll tell the switch to “tag” traffic from that port as belonging to a specific VLAN. Once the tag is added to the host’s transmissions, all other switches in the network treat that tagged traffic as belonging only to the one VLAN.

Although you can add router interfaces to separate a large LAN into manageable smaller LANs, reasons for using VLANs to do the job instead include the following:

* Identify groups of devices whose data should be given priority handling, such as executive client devices or an ICS (industrial control system) that manages a refrigeration system or a gas pipeline.
* Isolate connections with heavy or unpredictable traffic patterns, such as when separating heavy VoIP traffic from other network activities.
* Isolate groups of devices that rely on legacy protocols incompatible with the majority of the network’s traffic, such as a legacy SCADA (supervisory control and data acquisition) system monitoring an oil refinery.
* Separate groups of users who need special or limited security or network functions, such as when setting up a guest network.
* Configure temporary networks, such as when making specific network resources available to a short-term project team.
* Reduce the cost of networking equipment, such as when upgrading a network design to include additional departments or new types of network traffic.

Go to pg.

[**help**](javascript://)

Application Opened

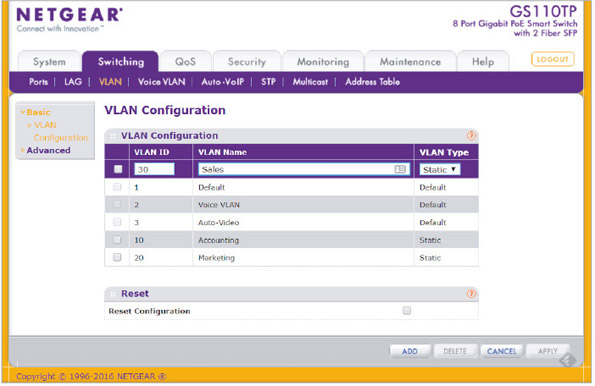
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-4aSwitch Port Configuration

Recall that managed switches can be configured via a CLI or a web-based management GUI (see [Figure 8-18](javascript://)). VLANs can only be implemented through managed switches, whose ports can be partitioned into groups.

**Figure 8-18**

Configure VLANs on a managed switch’s management interface



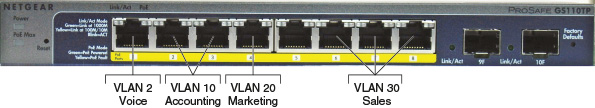
Enlarge Image

Source: Netgear

[Figure 8-19](javascript://) shows how the switch’s physical ports are grouped by VLAN, according to some of the VLANs listed in [Figure 8-18](javascript://). Notice that, for this switch, VLANs 1, 2, and 3 are default VLANs and cannot be deleted.

**Figure 8-19**

Each port on a managed switch might be configured for a different VLAN



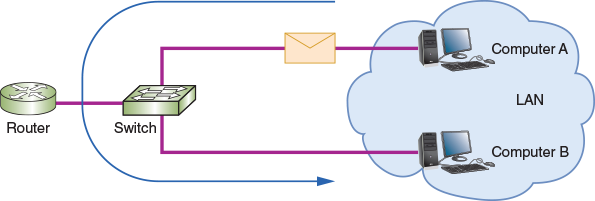
Enlarge Image

Recall that switches are primarily layer 2 devices (although, of course, you’ve learned about switches functioning at other layers to perform functions in addition to switching). By sorting traffic based on layer 2 information, VLANs create two or more broadcast domains from a single broadcast domain, which is also a layer 2 construct. Let’s look at some illustrations to see how this works.

[Figure 8-20](javascript://) shows a basic network with one broadcast domain. The switch manages all network traffic on the LAN unless a host on the network wants to communicate with a host on another network, and then that traffic goes through the router. For example, if Computer A on the LAN sends a message to Computer B on the same LAN, the switch handles the message without involving the router.

**Figure 8-20**

A switch connecting devices within a LAN to each other and to a router

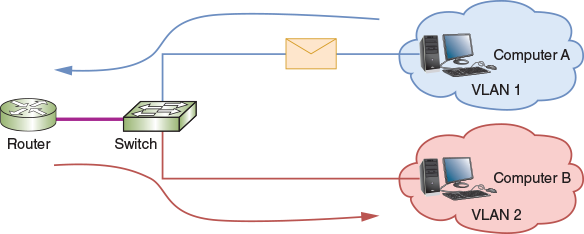


Enlarge Image

[Figure 8-21](javascript://) shows what happens when ports on a managed switch are partitioned into two VLANs. Traffic within each VLAN still goes through the switch as normal to reach other devices on the same VLAN. Traffic to hosts on other networks still goes through the router. However, traffic between hosts on VLAN 1 and VLAN 2 must now also go through the router, which is called inter-VLAN routing. This simple VLAN configuration, where one router connects to a switch that supports multiple VLANs, is sometimes called a “router-on-a-stick.”

**Figure 8-21**

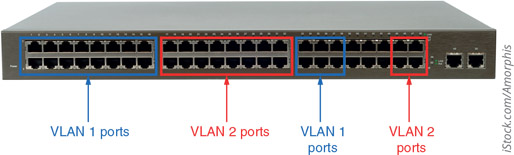
A managed switch with its ports partitioned into two groups, each belonging to a different VLAN



To visualize what happens at the hardware level, look at [Figure 8-22](javascript://). Here you can see that several ports on the switch are assigned to VLAN 1 or VLAN 2. The ports for a VLAN don’t have to be located next to each other—each port is individually configured to belong to a specific VLAN. Any device that is connected to a VLAN-configured port is automatically considered to be part of that VLAN. All transmissions coming from the connected host will be associated with the VLAN on that switch’s port.

**Figure 8-22**

Each port on a switch can be assigned to a different VLAN

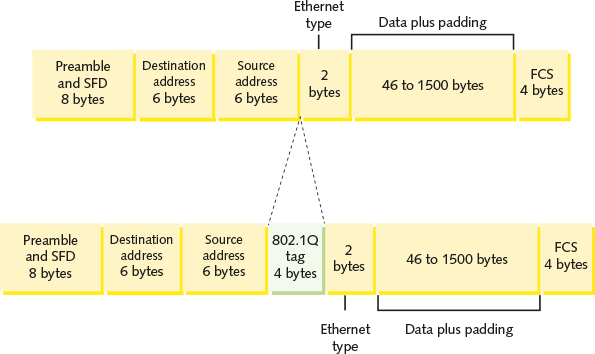


[iStock.com/Amorphis](http://istock.com/Amorphis" \t "_blank)

To identify the transmissions that belong to each VLAN, the switch adds a small field called a [**tag**](javascript://) to the Ethernet header that identifies the port through which messages arrive at the switch (see [Figure 8-23](javascript://)). The tag travels with the transmission until it reaches one of the following:

**Figure 8-23**

The 802.1Q VLAN tag is inserted after the Source address field in an Ethernet frame



Enlarge Image

* The switch port connected to the destination device, if the destination device is connected to the same switch as the sending device
* A router for routing to the correct VLAN, if the destination device is connected to a different switch

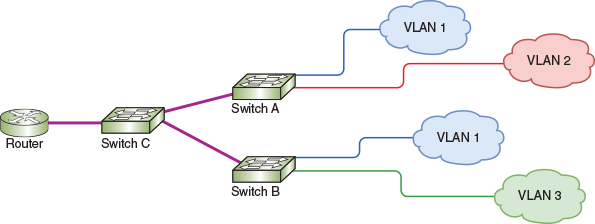
At that point, the tag is stripped from the frame. If the frame is being routed to a new VLAN, the router adds a new tag, which is then removed once the frame reaches its final switch port. In most cases, neither the sending device nor the receiving device is aware of the VLAN infrastructure.

Port tagging is specified in the IEEE [**802.1Q**](javascript://) standard, which defines how VLAN information appears in frames and how switches interpret that information. Note that the 802.1Q standard is sometimes referred to simply as “dot1q,” which will be easy to remember when you start working with VLAN commands in a Packet Tracer project at the end of this module.

You’ve seen that a switch can support more than one VLAN. Similarly, a VLAN can include ports from more than one switch. Suppose you add a couple more switches to the LAN, as in [Figure 8-24](javascript://). Switch B’s ports in this example network can be configured with the same or different VLANs as the ports on Switch A.

**Figure 8-24**

Three switches on a LAN with multiple VLANs



Enlarge Image

Consider the following scenarios:

* Traffic from a device on VLAN 1 connected to Switch A can travel to another device on VLAN 1 connected to Switch B as if it were local traffic (that is, same broadcast domain and won’t cross the router interface) because these two devices are in the same VLAN.
* Devices on separate VLANs—even if they’re connected to the same switch—can’t talk to each other without their traffic going through the router. Therefore, transmissions from a device on VLAN 1 connected to Switch B must go through the router to reach a device on VLAN 3, even though both devices are plugged into the same switch.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

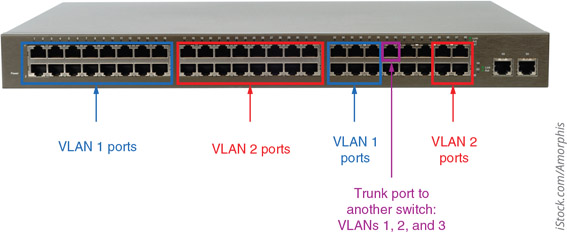
## 8-4bVLAN Trunks

Notice in [Figure 8-24](javascript://) that Switch A is connected to devices on two VLANs, and it’s also connected to Switch C. These are two very different types of connections. Ports connected to endpoint devices (hosts) are usually configured to support traffic for only one VLAN. However, the port that connects to Switch C (which is a node but not a host) must be able to carry traffic for multiple VLANs. Therefore, each port on a switch that supports VLANs is configured as one of two types of VLAN ports:

* [**Access port**](javascript://)—Connects the switch to a host, such as a workstation, server, or printer. The device connected to an access port does not know which VLAN it belongs to, nor can it recognize other VLANs on the same switch. An access port typically carries traffic for only one VLAN.
* [**Trunk port**](javascript://)—Connects the switch to a networking device such as a router or another switch (or possibly a server). This interface manages traffic from multiple VLANs, as shown in [Figure 8-25](javascript://). A trunk line (or just “trunk”) is a link between two trunk ports.

**Figure 8-25**

A trunk port supports traffic from multiple VLANs

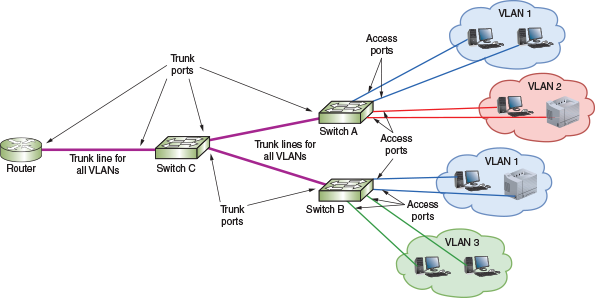


[iStock.com/Amorphis](http://istock.com/Amorphis" \t "_blank)

With **[trunking](javascript://)**, a single switch can support traffic belonging to several VLANs across the network. The term trunk originated in the telephony field, where it refers to an aggregation of logical connections over one physical connection. For example, a trunk carries signals for many residential telephone lines in the same neighborhood over one cable. Similarly, in the context of switching, a trunk is a single physical connection between networking devices through which many logical VLANs can transmit and receive data. [Figure 8-26](javascript://) shows the relative location of access ports, trunk ports, and trunk lines on the sample network.

**Figure 8-26**

Each trunk line carries traffic for multiple VLANs



Enlarge Image

Trunking protocols assign and interpret the VLAN tags in Ethernet frames, thereby managing the distribution of VLAN frames through a trunk. The most popular protocol for exchanging VLAN information over trunks is Cisco’s VTP (VLAN Trunk Protocol). VTP allows changes to a VLAN database on one switch, called the stack master, to be communicated to all other switches in the network. This provides network administrators with the ability to centrally manage all VLANs by making changes to a single switch. Other switches besides the stack master in the same VTP domain can also communicate VLAN updates, such as the addition of a new VLAN.

Go to pg.

[**help**](javascript://)

Application Opened

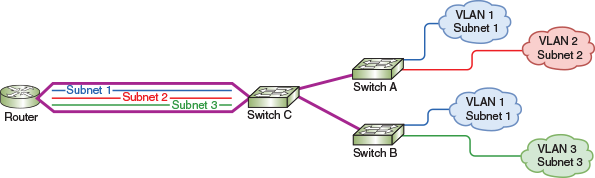
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-4cVLANs and Subnets

In most situations, each VLAN is assigned its own subnet of IP addresses. This means that a particular subnet, working at layer 3, includes the same group of hosts as a specific VLAN, working at layer 2. For example, the sample network (shown earlier in [Figure 8-24](javascript://) and again here in [Figure 8-27](javascript://)) is divided into three subnets where , , and .

**Figure 8-27**

Three subnets are connected to a single router interface

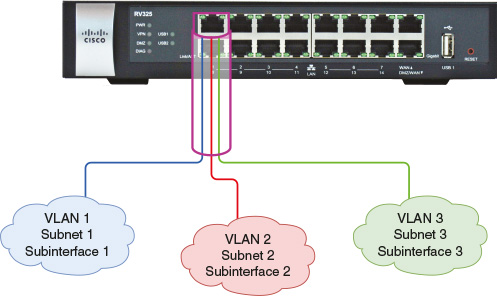


Enlarge Image

As traffic from each VLAN reaches the router, the router sees three logical LANs connected to a single router port, as you can see in [Figure 8-28](javascript://). Each of these logical interfaces on the one physical interface is called a **[subinterface](javascript://)**.

**Figure 8-28**

One router interface is configured to support three different subnets



For example, if these three VLANs are connected to the router’s fastethernet0/0 interface, they might be configured to use the following subinterfaces using decimal numbers at the end of each interface’s name:

* fastethernet0/0.1 for Subnet 1 and VLAN 1
* fastethernet0/0.2 for Subnet 2 and VLAN 2
* fastethernet0/0.3 for Subnet 3 and VLAN 3

Also, each VLAN and subnet combination acts as a single broadcast domain. Keep in mind, though, that VLANs and subnets serve two different segmentation purposes: Subnets organize IP addressing space at layer 3, while VLANs segment network traffic at layer 2. Additionally, you won’t necessarily see a 1:1 relationship between subnets and VLANs on every network—for example, you might see multiple subnets assigned to a single VLAN. Although it is possible to do otherwise, network administrators find life much easier when they adhere to the following rule:

So how do VLAN clients receive the appropriate IP address assignments from the subnet’s range of addresses portioned to each VLAN? One way to do this is to run a DHCP server for the entire network and use a DHCP relay agent to help sort DHCP requests by subnet, as described earlier in this module. Instead, the router can provide DHCP services with each subinterface configured with its own, subnetted range of IP addresses. In a project at the end of this module, you get practice doing exactly this using a router in Packet Tracer.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

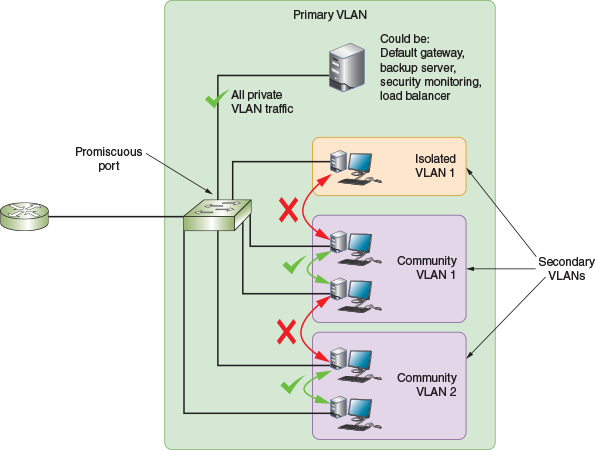
## 8-4dTypes of VLANs

You already know that different types of IP addresses serve different purposes (such as private, public, loopback, and APIPA IP addresses). The same is true of VLANs. Here are common VLAN types you’ll likely encounter when managing a network:

* [**Default VLAN**](javascript://)—Typically preconfigured on a switch and initially includes all the switch’s ports. Other VLANs might be preconfigured as well, depending on the device and manufacturer (such as the voice and video VLANs shown earlier in [Figure 8-18](javascript://)). The default VLAN cannot be renamed or deleted; however, ports in the default VLAN can be reassigned to other VLANs.
* **Native VLAN**—Receives all untagged frames from untagged ports. By default, this is the same as the default VLAN. However, this configuration poses a security risk when untagged traffic can travel in a VLAN-managed network. To protect the network from unauthorized traffic, the native VLAN should be changed to an unused VLAN so that untagged traffic essentially runs into a dead-end. To do this on a Cisco switch, for example, use the command switchport trunk native vlan. On a Juniper switch, the native VLAN is configured with the command set port-mode trunk followed by set native-vlan-id. Each switch port can be configured for a different native VLAN using these commands. However, switch ports on each end of a trunk should agree on the native VLAN assignment. A native VLAN mismatch (or just VLAN mismatch) is a configuration error that occurs when ports don’t agree.
* [**Data VLAN**](javascript://) **(or user VLAN)**—Carries user-generated traffic, such as email, web browsing, or database updates.
* **Management VLAN**—Can be used to provide administrative access to a switch. By default, this might be the same as the default VLAN; however, this poses a security risk and should be changed.
* [**Voice VLAN**](javascript://)—Supports VoIP traffic, which requires high bandwidths, priority over other traffic, flexible routing, and minimized latency.
* [**Private VLAN**](javascript://)—Partitions a VLAN broadcast domain into subdomains. Within a private VLAN is a single primary VLAN that defines the entire private VLAN’s domain, plus one or more secondary VLANs that each occupy a portion of the VLAN’s domain, as shown in [Figure 8-29](javascript://). Hosts assigned to a secondary VLAN cannot communicate outside their own subdomain within the private VLAN, which provides options for isolated VLAN traffic. However, a server or load balancer might instead be connected to a [**promiscuous port**](javascript://) within the primary VLAN so it can communicate with hosts inside all the secondary VLANs. Two types of secondary VLANs are illustrated in [Figure 8-29](javascript://) and are described next:

**Figure 8-29**

A private VLAN restricts communication between members of the VLAN



Enlarge Image

* + **Isolated VLAN**—The host on each switch port is completely isolated from hosts in the same and other secondary VLANs within the primary VLAN. A host in an isolated VLAN might or might not be allowed to communicate on a trunk line with the larger network (such as the Internet). This is commonly used when you want to isolate traffic within the VLAN but still allow each host to connect with the larger network or the Internet. For example, you might host an email server or a database server within an isolated VLAN.
  + **Community VLAN**—Hosts within the same community VLAN can communicate with each other but not with hosts in other secondary VLANs. Hosts in a community VLAN might or might not be allowed to communicate on a trunk line with the larger network and on to the Internet. You might use a community VLAN to group workstations within a department or host devices serving a specific customer (such as a tenant in a small office building).

In addition to defining the types of traffic handled by a VLAN, you can also specify security parameters, filtering instructions (for example, if the switch should not forward any frames from a certain VLAN), performance requirements for certain ports, and network addressing and management options. Options vary according to the switch manufacturer and model. In a Capstone Project at the end of this module, you will have the opportunity to create and configure VLANs on a switch in your Packet Tracer network.

Go to pg.

[**help**](javascript://)

Application Opened

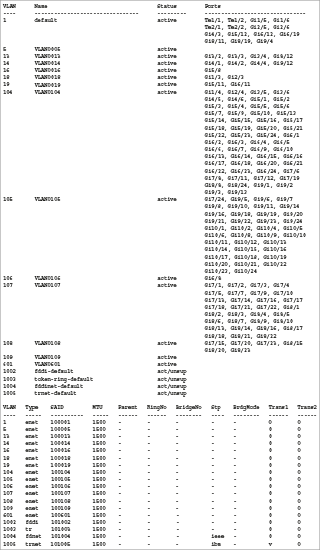
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-4eView Configured VLANs

Once you create a VLAN, you also maintain it via the switch’s software. [Figure 8-30](javascript://) illustrates the output of a show vlan command on a Cisco switch on a large, enterprise network. The show vlan command is used to list the current VLANs recognized by a switch. The OSs on other manufacturers’ switches include similar maintenance commands.

**Figure 8-30**

Output of the show vlan command on a Cisco switch



Enlarge Image

[Figure 8-30](javascript://) lists 18 VLANs configured on the network. The following list analyzes this output:

* The first half of the command output shows each VLAN’s number, name, status, and which ports belong to it. For example, VLAN number 18, which is named “VLAN0018,” is active and contains the ports “Gi1/3” and “Gi2/3.” A port called “Gi1/3,” in this case, refers to the third port on the first Gigabit Ethernet module of this switch.
* VLAN number 1 and VLANs 1002 through 1005 are defaults pre-established on the Cisco switch. Other than VLAN 1, these default VLANs are not currently in use.
* The second half of the command output provides additional information about each VLAN, including the type of network it operates on. In this example, all VLANs that are active and not pre-established defaults use Ethernet, which is indicated by the enet type.
* Each VLAN is assigned a different SAID (security association identifier), which indicates to other connectivity devices which VLAN a transmission belongs to. By default, Cisco switches assign a VLAN the SAID of 100,000 plus the VLAN number.
* In this example, each VLAN is configured to transmit and receive frames with an MTU (maximum transmission unit) of 1500 bytes, which is the default selection. Rarely do network administrators change this variable.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-4fDynamic VLAN Assignment

You’ve read how devices can be assigned to a VLAN based on the switch port the device is connected to. This is called a [**static VLAN assignment**](javascript://). A device might instead receive a [**dynamic VLAN assignment**](javascript://) according to other criteria, such as the following:

* Client device information, such as MAC address or location, can be used to group devices by VLAN when they first join the network.
* Authentication processes in cooperation with a RADIUS server can be used to further distinguish which traffic should be assigned to which VLANs, such as when an IT employee signs in on a workstation compared to when a sales employee signs in on the same device. You’ll learn more about RADIUS when studying authentication and security through network design.
* Devices that have not yet authenticated to the network or whose authentication failed can be placed in a quarantine VLAN for basic Internet access and for communications required to attempt authentication.
* All WLAN traffic might be grouped within the same VLAN. Alternatively, authentication methods might be used to further segment wireless traffic, or VLAN traffic can be tagged according to the SSID the device is connected to.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 8-4gTroubleshoot and Secure VLANs

Configuration errors are a common cause of VLAN problems. The show vlan command discussed earlier yields information that can help you identify misconfigurations. Start by checking the configuration against your documentation and then check physical connections. If that doesn’t work, consider these common configuration errors:

* **Incorrect VLAN assignment**—This can happen due to a variety of situations, including plugging a device into the wrong switch port or misconfigurations of the client authentication process in which a VLAN is assigned to a device before the authentication process is complete. If devices are not able to communicate with network services as expected, checking VLAN assignment and other configuration options should be an early step in the troubleshooting process.
* **Incorrect port mode**—Switch ports connected to endpoints, such as workstations and servers, should nearly always use access mode. Switch ports connected to other network devices should be configured in trunk mode only if that connection must support multiple VLANs.
* **VLAN isolation**—By grouping certain nodes into a VLAN, you are not merely including those nodes—you are also excluding other groups of nodes. This means you can potentially cut off an entire group of nodes from the rest of the network. VLANs must be connected to and configured on a router or layer 3 switch to allow different VLANs to exchange data outside their own broadcast domain.

Hackers sometimes take advantage of the way VLANs are tagged to implement an attack called [**VLAN hopping**](javascript://). The attacker generates transmissions that appear, to the switch, to belong to a protected VLAN, then crosses VLANs to access sensitive data or inject harmful software. There are two approaches to VLAN hopping:

* **Double tagging**—The hacker stacks VLAN tags in Ethernet frames. When the first, legitimate tag is removed by a switch, the second, illegitimate tag is revealed, tricking a switch into forwarding the transmission on to a restricted VLAN.
* **Switch spoofing**—An attacker connects to a switch and then makes the connection look to the switch as if it’s a trunk line. The switch might auto-configure its port into trunk mode when it detects trunk mode on the other end of the connection. A hacker can then feed their own VLAN traffic into that port and access VLANs throughout the network.

The following mitigation efforts will reduce the risk of VLAN hopping and further help enforce network segmentation:

* Don’t use the default VLAN.
* Change the native VLAN to an unused VLAN ID.
* Disable auto-trunking on switches that don’t need to support traffic from multiple VLANs.
* On switches that do carry traffic from multiple VLANs, configure all ports as access ports unless they’re used as trunk ports.
* Specify which VLANs are supported on each trunk instead of accepting a range of all VLANs.
* Use physical security methods such as door locks to restrict access to network equipment.

**Remember This…**

* Explain how port tagging supports VLANs.
* Configure subinterfaces on a router.
* Compare common types of VLANs, including default VLAN, data VLAN, voice VLAN, and private VLAN.
* Describe options for dynamic VLAN assignments.
* List common VLAN problems, including incorrect VLAN, incorrect port mode, and VLAN isolation.
* Explain the threat posed by VLAN hopping.

**Self-Check**

1. At what OSI layer do VLANs function?

Answer

* 1. Network layer
  2. Transport layer
  3. Physical layer
  4. Data link layer

1. Suppose you have a small network with one router, one switch, and a few computers that are grouped into three VLANs. Which of the following statements is false?

Answer

* 1. Traffic between computers on the same VLAN must go through the router.
  2. Traffic from any computer to another network must go through the router.
  3. Traffic between computers on different VLANs must go through the router.

1. Which VLAN on a switch cannot be renamed or deleted?

Answer

* 1. Native VLAN
  2. Management VLAN
  3. Default VLAN
  4. Data VLAN

**You’re Ready**

You’re now ready to complete [Project 8-4: Configure VLANs Using a Switch’s GUI](javascript://), or you can wait until you’ve finished the Review Questions for this module.

**You’re Ready**

After you finish the Hands-On Projects, you’re ready to complete the [Module 8 Capstone Projects](javascript://).

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 8-5a**Module Summary**

### Network Segmentation

* A network administrator might separate a network’s traffic into smaller portions to enhance security, improve network performance, and simplify troubleshooting.
* Networks are commonly segmented according to geographic locations, departmental boundaries, or device types.
* You can use physical devices such as routers to create separate LANs. At layer 2, you can create virtual LANs. At layer 3, you can use subnetting to organize devices within the available IP address space, whether the LANs are defined physically or virtually.

### Subnet Masks

* Connecting each switch to a different router interface will segment network traffic. A router doesn’t forward broadcast traffic between its interfaces, and so this configuration will break up the one, large broadcast domain into three smaller broadcast domains. You can think of a router as a broadcast boundary, and fundamentally, routers are tools you can use to divide and conquer network traffic.
* Managing the IP address space at the logical layer helps clients know which devices are on their own subnet and which devices are on other networks. Subnetting helps solve the fundamental problem with classful addressing: too many node addresses assigned to each classful network. Well-chosen subnets make network documentation easier to manage, helps to locate and resolve network problems, and makes routing more efficient.
* A device uses a subnet mask to determine which subnet or network it belongs to. When a computer is ready to send a transmission to another device, it first compares the bits in its own network ID to the bits in the network ID of the destination device. If the bits match, the other device is on the sending computer’s own network, and it sends the transmission directly to that device. If the bits don’t match, the destination is on another network, and the computer sends the transmission to the default gateway on its network.
* In classful addressing, each network class is associated with a default subnet mask. In contrast, classless addressing allows the network to “borrow” host bits to identify subnets within the network.

### Calculating Subnets

* Subnetting, or classless addressing, alters the rules of classful IPv4 addressing. To subnet a network, you borrow bits that would represent host information in classful addressing and use those bits instead to represent network information. By doing so, you increase the number of bits available for the network ID, and you also reduce the number of bits available for identifying hosts. Consequently, you increase the number of networks and reduce the number of usable host addresses in each network or subnet. The more bits you borrow for network information, the more subnets you can have, but the fewer hosts each subnet will contain.
* To calculate subnets on a network, you first determine how many host bits to borrow, then calculate the subnet mask, determine the network ID for each subnet, and identify the range of IP addresses for hosts on each subnet.
* Common formulas used in calculating subnet information include , 256 – the interesting octet, and . You can also use subnet mask tables to look up common calculations.
* There are two common types of subnetting questions on certification exams:
  + Given certain network requirements (such as required number of hosts or required number of subnets), calculate possible subnets and host IP address ranges.
  + Given an IP address, determine its subnet’s network ID, broadcast address, and first/last host addresses.
* The administrator must program each interface on the router with its IP address and subnet mask for its subnet. Though tedious on larger networks, static IP addressing can also be used on network hosts. For dynamic IP addressing, the administrator programs each subnet’s DHCP server with the network ID, subnet mask, range of IP addresses, and default gateway for the subnet.
* Allowing some types of broadcast traffic, such as DHCP messages, to travel beyond each broadcast domain lets hosts in various subnets access centralized network services. DHCP and other centralized network services use UDP traffic, including TFTP, NTP, and DNS. Configuring UDP forwarding on the network allows routers, firewalls, or layer 3 switches to forward this UDP traffic across broadcast domains, which enables centralization of key network services.
* Traditional subnetting results in multiple subnets that are all the same size, and this uniformity in subnet size can be inefficient in complex networks. VLSM (Variable Length Subnet Mask) allows subnets to be further subdivided into smaller and smaller groupings until each subnet is about the same size as the necessary IP address space.
* IPv6 addressing uses no classes and does not use subnet masks. A single IPv6 subnet can supply 18,446,744,073,709,551,616 IPv6 addresses. The fourth hexadecimal block in the site prefix can be altered to create subnets within a site.

### Virtual LANs (VLANs)

* A VLAN (virtual local area network or virtual LAN) groups ports on one or more switches so that some of the local traffic on the switch is forced to go through a router, thereby limiting the traffic to a smaller broadcast domain. As virtual LANs, VLANs abstract the broadcast domain from the networking hardware. When using VLANs, the boundaries of the broadcast domain can be virtually defined anywhere within a single physical LAN.
* VLANs can only be implemented through managed switches, whose ports can be partitioned into groups. When ports on a managed switch are partitioned into VLANs, traffic within each VLAN still goes through the switch as normal to reach other devices on the same VLAN. Traffic to hosts on other networks still goes through the router. However, traffic between hosts on VLAN 1 and VLAN 2 must now also go through the router, which is called inter-VLAN routing.
* Two types of VLAN ports include access ports, which connect the switch to hosts and typically carries traffic for only one VLAN, and trunk ports, which connect a switch to another networking device such as a router or another switch and manages traffic from multiple VLANs.
* In most situations, each VLAN is assigned its own subnet of IP addresses. This means that a particular subnet, working at layer 3, includes the same group of hosts as a specific VLAN, working at layer 2.
* Common VLAN types include default VLANs, a native VLAN, data VLANs, the management VLAN, a voice VLAN, and a private VLAN, which contains secondary VLANs.
* On Cisco devices, use the show vlan command to view a list of the current VLANs configured on a switch. The OSs on other manufacturers’ switches include similar maintenance commands.
* Devices can be assigned to a VLAN based on the switch port the device is connected to. This is called a static VLAN assignment. A device might instead receive a dynamic VLAN assignment according to other criteria, including client device information, results of authentication processes, the lack of completed authentication, or WLAN association.
* Common VLAN problems include incorrect VLAN assignment, incorrect port mode, or unintended isolation of devices on a VLAN. VLAN hopping is an attack type specific to the VLAN configurations.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 8-5b**Key Terms**

* [**802.1Q**](javascript://)
* [**access port**](javascript://)
* [**ANDing**](javascript://)
* [**data VLAN**](javascript://)
* [**default VLAN**](javascript://)
* [**dynamic VLAN assignment**](javascript://)
* [**global routing prefix**](javascript://)
* [**IP helper address**](javascript://)
* [**magic number**](javascript://)
* [**private VLAN**](javascript://)
* [**promiscuous port**](javascript://)
* [**relay agent**](javascript://)
* [**screened subnet**](javascript://)
* [**site prefix**](javascript://)
* [**static VLAN assignment**](javascript://)
* [**subinterface**](javascript://)
* [**tag**](javascript://)
* [**trunk port**](javascript://)
* [**trunking**](javascript://)
* [**UDP forwarding**](javascript://)
* [**VLAN (virtual local area network or virtual LAN)**](javascript://)
* [**VLAN hopping**](javascript://)
* [**VLSM (Variable Length Subnet Mask)**](javascript://)
* [**voice VLAN**](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 8-5c**Review Questions**

1. How many bits of a class A IP address are used for host information?
   1. 8 bits
   2. 16 bits
   3. 24 bits
   4. 32 bits
2. What is the formula for determining the number of possible hosts on a network?
3. Which of the following is not a good reason to segment a network?
   1. To limit access to broadcast domains
   2. To reduce the demand on bandwidth
   3. To increase the number of networking devices on a network
   4. To narrow down the location of problems on a network
4. What is the greatest number of bits you could borrow from the host portion of a class B subnet mask and still have at least 130 hosts per subnet?
   1. 0 bits
   2. 8 bits
   3. 9 bits
   4. 10 bits
5. What do well-chosen subnets accomplish?
   1. IP address spaces overlap for easier management.
   2. Network documentation is easier to manage.
   3. Routing efficiency is decreased by ensuring IP address spaces are not mathematically related.
   4. Problems affect the entire network, making them more difficult to pin down.
6. Which formulas can be used to calculate the magic number? Choose two.
   1. 256 – the interesting octet
7. Which hexadecimal block in an IPv6 address is used for the Subnet ID?
   1. The first one
   2. The third one
   3. The fourth one
   4. The eighth one
8. Which assignment technique requires a RADIUS server?
   1. Dynamic VLAN assignment
   2. Dynamic IP address assignment
   3. Static IP address assignment
   4. Static VLAN assignment
9. Which port mode on a switch enables that port to manage traffic for multiple VLANs?
   1. Private
   2. Community
   3. Access
   4. Trunk
10. Which IEEE standard determines how VLANs work on a network?
    1. 802.1X
    2. 802.11
    3. 802.3af
    4. 802.1Q
11. What is the network ID with CIDR notation for the IP address 172.16.32.108 whose subnet mask is 255.255.255.0?
12. Suppose your company has leased one class C license, 120.10.10.0, and wants to sublease the first half of these IP addresses to another company. What is the CIDR notation for the subnet to be subleased? What is the subnet mask for this network?
13. Subnetting operates at the  layer while VLANs function at the  layer.
14. Which VLAN on a switch manages untagged frames?
15. An attacker configures a VLAN frame with two tags instead of just one. The first tag directs the frame to the authorized VLAN. After the frame enters the first VLAN, the switch appropriately removes the tag, then discovers the next tag, and sends the frame along to a protected VLAN, which the attacker is not authorized to access. What kind of attack is this?
16. What area of a network can provide less stringent security so a web server is more accessible from the open Internet?
17. On which networking device do you configure VLANs?
18. Which IP addressing technique subnets a subnet to create subnets of various sizes?
19. Which VLAN type would be the best fit for a company’s web servers that need to be accessible from the Internet but should not be able to communicate with each other?
20. Which Cisco command lists configured VLANs on a switch?

Go to pg.

[**help**](javascript://)