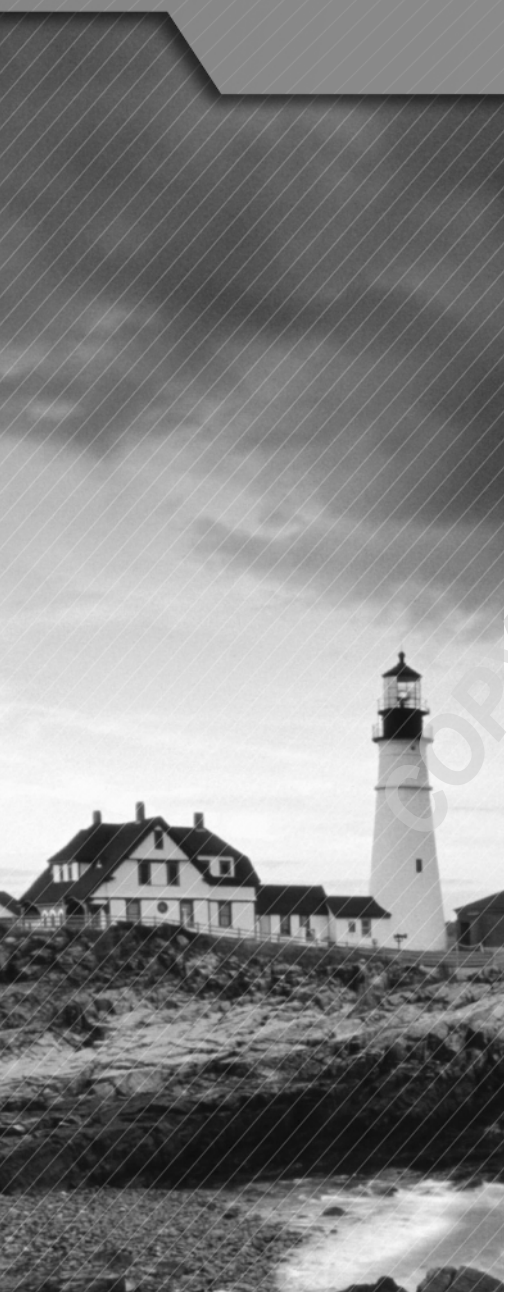
Chapter

1

Network Fundamentals



THE FOLLOWING CCNA ROUTING AND SWITCHING EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

- ✓ Compare and contrast OSI and TCP/IP models.
- ✓ Compare and contrast TCP and UDP protocols.
- ✓ Describe the impact of infrastructure components in an enterprise network.
- ✓ Describe the effects of cloud resources on enterprise network architecture.
- ✓ Compare and contrast collapsed core and three-tier architectures.
- ✓ Compare and contrast network topologies.
- ✓ Select the appropriate cabling type based on implementation requirements.
- ✓ Apply troubleshooting methodologies to resolve problems.
- ✓ Configure, verify, and troubleshoot IPv4 addressing and subnetting.
- ✓ Compare and contrast IPv4 address types.
- ✓ Describe the need for private IPv4 addressing.
- ✓ Identify the appropriate IPv6 addressing scheme to satisfy addressing requirements in a LAN/WAN environment.
- ✓ Configure, verify, and troubleshoot IPv6 addressing.
- ✓ Configure and verify IPv6 Stateless Address Auto Configuration.
- ✓ Compare and contrast IPv6 address types.

In this chapter, I will review the basics of internetworking and what an internetwork is. I will go over some of the components that make up a network as well as some applications used in networking. I will also go over the OSI and TCP/IP models and, finally, explain how data flows across a network as well as discuss the various connectors used in a network.

Compare and contrast OSI and TCP/IP models

A *reference model* is a conceptual blueprint of how communications should take place. It addresses all the processes required for effective communication and divides them into logical groupings called layers. When a communication system is designed in this manner, it's known as a hierarchical or *layered architecture*. In this section two models covered on the exam are compared and contrasted.

The OSI Reference Model

The OSI model is hierarchical, and there are many advantages that can be applied to any layered model, but as I said, the OSI model's primary purpose is to allow different vendors' networks to interoperate.

Here's a list of some of the more important benefits of using the OSI layered model:

- It divides the network communication process into smaller and simpler components, facilitating component development, design, and troubleshooting.
- It allows multiple-vendor development through the standardization of network components.
- It encourages industry standardization by clearly defining what functions occur at each layer of the model.
- It allows various types of network hardware and software to communicate.
- It prevents changes in one layer from affecting other layers to expedite development.

The OSI has seven different layers, divided into two groups. The top three layers define how the applications within the end stations will communicate with each other as well as with users. The bottom four layers define how data is transmitted end to end.

Figure 1.1 shows the three upper layers and their functions.

FIGURE 1.1 The upper layers

Application	• Provides a user interface
Presentation	• Presents data • Handles processing such as encryption
Session	• Keeps different applications' data separate

When looking at Figure 1.1, understand that users interact with the computer at the Application layer and also that the upper layers are responsible for applications communicating between hosts. None of the upper layers knows anything about networking or network addresses because that's the responsibility of the four bottom layers.

In Figure 1.2, which shows the four lower layers and their functions, you can see that it's these four bottom layers that define how data is transferred through physical media like wire, cable, fiber optics, switches, and routers. These bottom layers also determine how to rebuild a data stream from a transmitting host to a destination host's application.

FIGURE 1.2 The lower layers

Transport	• Provides reliable or unreliable delivery • Performs error correction before retransmit
Network	• Provides logical addressing, which routers use for path determination
Data Link	• Combines packets into bytes and bytes into frames • Provides access to media using MAC address • Performs error detection not correction
Physical	• Moves bits between devices • Specifies voltage, wire speed, and pinout of cables

The following network devices operate at all seven layers of the OSI model:

- Network management stations (NMSs)
- Web and application servers
- Gateways (not default gateways)
- Servers
- Network hosts

The OSI reference model has the following seven layers:

- Application layer (layer 7)
- Presentation layer (layer 6)
- Session layer (layer 5)
- Transport layer (layer 4)

- Network layer (layer 3)
- Data Link layer (layer 2)
- Physical layer (layer 1)

Some people like to use a mnemonic to remember the seven layers, such as **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing. Figure 1.3 shows a summary of the functions defined at each layer of the OSI model.

FIGURE 1.3 OSI layer functions

Application	• File, print, message, database, and application services
Presentation	• Data encryption, compression, and translation services
Session	• Dialog control
Transport	• End-to-end connection
Network	• Routing
Data Link	• Framing
Physical	• Physical topology

I've separated the seven-layer model into three different functions: the upper layers, the middle layers, and the bottom layers. The upper layers communicate with the user interface and application, the middle layers do reliable communication and routing to a remote network, and the bottom layers communicate to the local network.

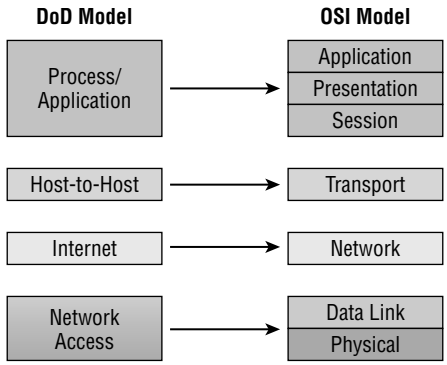
TCP/IP and the DoD Model

The DoD model is basically a condensed version of the OSI model that comprises four instead of seven layers:

- Process/Application layer
- Host-to-Host layer or Transport layer
- Internet layer
- Network Access layer or Link layer

Figure 1.4 offers a comparison of the DoD model and the OSI reference model. As you can see, the two are similar in concept, but each has a different number of layers with different names. Cisco may at times use different names for the same layer, such as both "Host-to-Host" and Transport" at the layer above the Internet layer, as well as "Network Access" and "Link" used to describe the bottom layer.

FIGURE 1.4 The DoD and OSI models



Exam Essentials

List the layers of the OSI and TCP/IP models. List the layers in order, and describe the function of each layer.

Compare and contrast the layers of the TCP/IP and OSI models. Identify the layers in each model that perform like functions.

Compare and contrast TCP and UDP protocols

The main purpose of the Host-to-Host layer is to shield the upper-layer applications from the complexities of the network. Coming up, I'll introduce you to the two protocols at this layer:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP) takes large blocks of information from an application and breaks them into segments. It numbers and sequences each segment so that the destination's TCP stack can put the segments back into the order the application intended. After these segments are sent on the transmitting host, TCP waits for an acknowledgment of the receiving end's TCP virtual circuit session, retransmitting any segments that aren't acknowledged.

Before a transmitting host starts to send segments down the model, the sender's TCP stack contacts the destination's TCP stack to establish a connection. This creates a *virtual circuit*, and this type of communication is known as *connection-oriented*. During this initial handshake, the two TCP layers also agree on the amount of information that's going to be sent before the recipient's TCP sends back an acknowledgment. With everything agreed upon in advance, the path is paved for reliable communication to take place.

TCP is a full-duplex, connection-oriented, reliable, and accurate protocol, but establishing all these terms and conditions, in addition to error checking, is no small task. TCP is very complicated, and so not surprisingly, it's costly in terms of network overhead. And since today's networks are much more reliable than those of yore, this added reliability is often unnecessary. Most programmers use TCP because it removes a lot of programming work, but for real-time video and VoIP, *User Datagram Protocol (UDP)* is often better because using it results in less overhead.

TCP Segment Format

Since the upper layers just send a data stream to the protocols in the Transport layers, I'll use Figure 1.5 to demonstrate how TCP segments a data stream and prepares it for the Internet layer. When the Internet layer receives the data stream, it routes the segments as packets through an internetwork. The segments are handed to the receiving host's Host-to-Host layer protocol, which rebuilds the data stream for the upper-layer applications or protocols.

FIGURE 1.5 TCP segment format

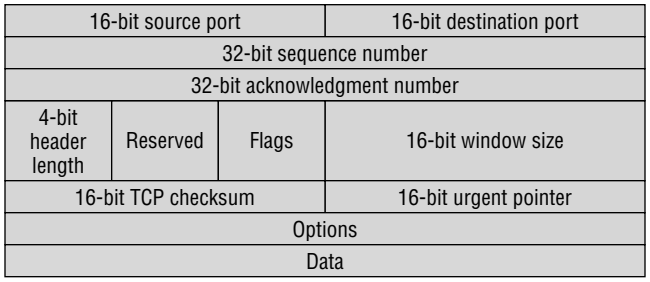


Figure 1.5 shows the TCP segment format and shows the different fields within the TCP header. This isn't important to memorize for the Cisco exam objectives, but you need to understand it well because it's really good foundational information.

The TCP header is 20 bytes long, or up to 24 bytes with options. You need to understand what each field in the TCP segment is in order to build a strong educational foundation:

Source port This is the port number of the application on the host sending the data, which I'll talk about more thoroughly a little later in this chapter.

Destination port This is the port number of the application requested on the destination host.

Sequence number A number used by TCP that puts the data back in the correct order or retransmits missing or damaged data during a process called sequencing

Acknowledgment number The value is the TCP octet that is expected next.

Header length The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits in length.

Reserved Always set to zero

Code bits/flags Controls functions used to set up and terminate a session

Window The window size the sender is willing to accept, in octets

Checksum The cyclic redundancy check (CRC), used because TCP doesn't trust the lower layers and checks everything. The CRC checks the header and data fields.

Urgent A valid field only if the Urgent pointer in the code bits is set. If so, this value indicates the offset from the current sequence number, in octets, where the segment of non-urgent data begins.

Options May be 0, meaning that no options have to be present, or a multiple of 32 bits. However, if any options are used that do not cause the option field to total a multiple of 32 bits, padding of 0s must be used to make sure the data begins on a 32-bit boundary. These boundaries are known as words.

Data Handed down to the TCP protocol at the Transport layer, which includes the upper-layer headers

Let's take a look at a TCP segment copied from a network analyzer:

```
TCP - Transport Control Protocol
 Source Port:      5973
 Destination Port: 23
 Sequence Number:  1456389907
 Ack Number:       1242056456
 Offset:           5
 Reserved:         %000000
 Code:             %011000
      Ack is valid
      Push Request
 Window:           61320
 Checksum:         0x61a6
 Urgent Pointer:   0
 No TCP Options
 TCP Data Area:
 vL.5.+.5.+.5.+.5  76 4c 19 35 11 2b 19 35 11 2b 19 35 11
  2b 19 35 +. 11 2b 19
Frame Check Sequence: 0x0d00000f
```

Did you notice that everything I talked about earlier is in the segment? As you can see from the number of fields in the header, TCP creates a lot of overhead. Again, this is why application developers may opt for efficiency over reliability to save overhead and go with UDP instead. It's also defined at the Transport layer as an alternative to TCP.

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is basically the scaled-down economy model of TCP, which is why UDP is sometimes referred to as a thin protocol. Like a thin person on a park bench, a thin protocol doesn't take up a lot of room—or in this case, require much bandwidth on a network.

UDP doesn't offer all the bells and whistles of TCP either, but it does do a fabulous job of transporting information that doesn't require reliable delivery, using far less network resources. (UDP is covered thoroughly in Request for Comments 768.)

So clearly, there are times that it's wise for developers to opt for UDP rather than TCP, one of them being when reliability is already taken care of at the Process/Application layer. Network File System (NFS) handles its own reliability issues, making the use of TCP both impractical and redundant. But ultimately, it's up to the application developer to opt for using UDP or TCP, not the user who wants to transfer data faster!

UDP does *not* sequence the segments and does not care about the order in which the segments arrive at the destination. UDP just sends the segments off and forgets about them. It doesn't follow through, check up on them, or even allow for an acknowledgment of safe arrival—complete abandonment. Because of this, it's referred to as an unreliable protocol. This does not mean that UDP is ineffective, only that it doesn't deal with reliability issues at all.

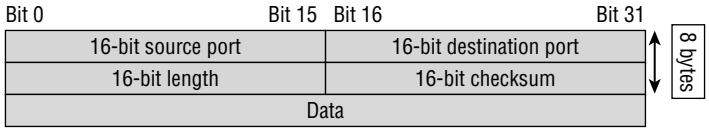
Furthermore, UDP doesn't create a virtual circuit, nor does it contact the destination before delivering information to it. Because of this, it's also considered a *connectionless* protocol. Since UDP assumes that the application will use its own reliability method, it doesn't use any itself. This presents an application developer with a choice when running the Internet Protocol stack: TCP for reliability or UDP for faster transfers.

It's important to know how this process works because if the segments arrive out of order, which is commonplace in IP networks, they'll simply be passed up to the next layer in whatever order they were received. This can result in some seriously garbled data! On the other hand, TCP sequences the segments so they get put back together in exactly the right order, which is something UDP just can't do.

UDP Segment Format

Figure 1.6 clearly illustrates UDP's markedly lean overhead as compared to TCP's hungry requirements. Look at the figure carefully—can you see that UDP doesn't use windowing or provide for acknowledgments in the UDP header?

FIGURE 1.6 UDP segment



It's important for you to understand what each field in the UDP segment is:

Source port Port number of the application on the host sending the data

Destination port Port number of the application requested on the destination host

Length Length of UDP header and UDP data

Checksum Checksum of both the UDP header and UDP data fields

Data Upper-layer data

UDP, like TCP, doesn't trust the lower layers and runs its own CRC. Remember that the Frame Check Sequence (FCS) is the field that houses the CRC, which is why you can see the FCS information.

The following shows a UDP segment caught on a network analyzer:

```
UDP - User Datagram Protocol
 Source Port:      1085
 Destination Port: 5136
 Length:           41
 Checksum:         0x7a3c
 UDP Data Area:
 ..Z......00 01 5a 96 00 01 00 00 00 00 00 11 0000 00
...C..2._C._C  2e 03 00 43 02 1e 32 0a 00 0a 00 80 43 00 80
Frame Check Sequence: 0x00000000
```

Notice that low overhead! Try to find the sequence number, ack number, and window size in the UDP segment. You can't because they just aren't there!

Key Concepts of Host-to-Host Protocols

Since you've now seen both a connection-oriented (TCP) and connectionless (UDP) protocol in action, it's a good time to summarize the two here. Table 1.1 highlights some of the key concepts about these two protocols for you to memorize.

TABLE 1.1 Key features of TCP and UDP

TCP	UDP
Sequenced	Unsequenced
Reliable	Unreliable
Connection-oriented	Connectionless
Virtual circuit	Low overhead
Acknowledgments	No acknowledgment
Windowing flow control	No windowing or flow control of any type

Exam Essentials

Compare and contrast UDP and TCP. Describe the differences in purpose and capability of the two transport layer protocols, including overhead and services offered. Also describe when each is used.

Describe the impact of infrastructure components in an enterprise network

Various internetworking devices offer services that are critical to the network. In this section, I will review three important components and the role each plays in making the network function in a secure fashion.

Firewalls

Firewalls are hardware appliances or special software running on servers that control the flow of traffic between parts of the network. Routers can also be configured to perform this service.

These devices are network security systems that monitor and control the incoming and outgoing network traffic based on predetermined security rules, and they are usually intrusion protection systems (IPSs). The Cisco Adaptive Security Appliance (ASA) firewall typically establishes a barrier between a trusted, secure internal network and the Internet, which is not secure or trusted. Cisco's new acquisition of Sourcefire puts it at the top of the market with Next Generation Firewalls (NGFW) and Next Generation IPS (NGIPS), which Cisco now just calls Firepower. Cisco's new Firepower runs on dedicated appliances, Cisco ASAs, ISR routers, and even Meraki products.

Access Points

These devices allow wireless devices to connect to a wired network and extend a collision domain from a switch and are typically in their own broadcast domain, or what is referred to as a *virtual LAN* (VLAN). An AP can be a simple standalone device, but today they are usually managed by wireless controllers either in-house or through the Internet.

Wireless Controllers

These are the devices that network administrators or network operations centers use to manage access points in medium to large to extremely large quantities. The WLAN controller automatically handles the configuration of wireless access points and was typically used only in larger enterprise systems. However, with Cisco's acquisition of Meraki

systems, you can easily manage a small to medium-sized wireless network via the cloud using its simple-to-configure web controller system.

Exam Essentials

Describe the features of infrastructure components in an enterprise network. These include but are not limited to access points, WLAN controllers, and firewalls. Specific firewall solutions include the Cisco Adaptive Security Appliance (ASA), Next Generation Firewalls (NGFW), and Next Generation IPS (NGIPS), which Cisco now just calls Firepower.

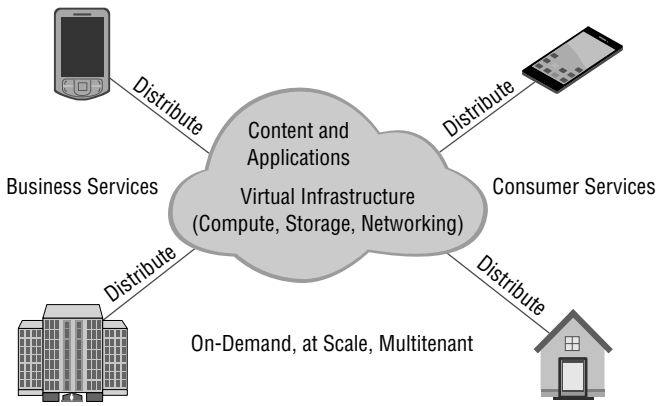
Describe the effects of cloud resources on enterprise network architecture

Cloud computing is by far one of the hottest topics in today's IT world. Basically, cloud computing can provide virtualized processing, storage, and computing resources to users remotely, making the resources transparently available regardless of the user connection. To put it simply, some people just refer to the cloud as "someone else's hard drive." This is true, of course, but the cloud is much more than just storage.

The history of the consolidation and virtualization of our servers tells us that this has become the de facto way of implementing servers because of basic resource efficiency. Two physical servers will use twice the amount of electricity as one server, but through virtualization, one physical server can host two virtual machines, hence the main thrust toward virtualization. With it, network components can simply be shared more efficiently.

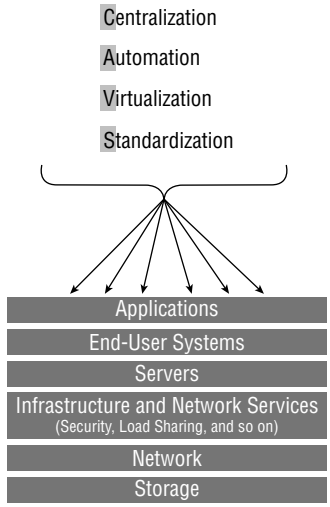
Users connecting to a cloud provider's network, whether it be for storage or applications, really don't care about the underlying infrastructure because as computing becomes a service rather than a product, it's then considered an on-demand resource, described in Figure 1.7.

FIGURE 1.7 Cloud computing is on-demand



Centralization/consolidation of resources, automation of services, virtualization, and standardization are just a few of the big benefits cloud services offer. Let's take a look in Figure 1.8.

FIGURE 1.8 Advantages of cloud computing



Traffic Path to Internal and External Cloud Services

Centralization/consolidation of resources, automation of services, virtualization, and standardization are just a few of the big benefits cloud services offer as shown in Figure 1.8.

Cloud computing has several advantages over the traditional use of computer resources. Following are advantages to the provider and to the cloud user.

Here are the advantages to a cloud service builder or provider:

- Cost reduction, standardization, and automation
- High utilization through virtualized, shared resources
- Easier administration
- Fall-in-place operations model

Here are the advantages to cloud users:

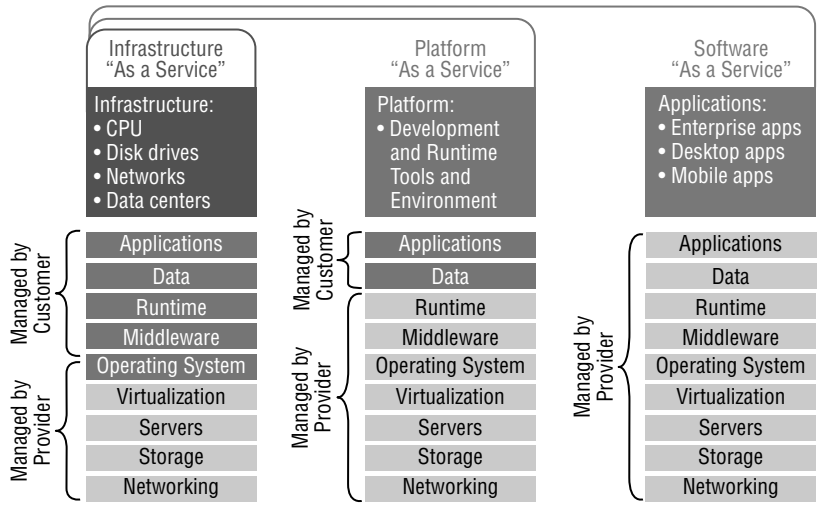
- On-demand, self-service resource provisioning
- Fast deployment cycles
- Cost effective
- Centralized appearance of resources
- Highly available, horizontally scaled application architectures
- No local backups

Virtual Services

Cloud providers can offer you different available resources based on your needs and budget. You can choose just a vitalized network platform or go all in with the network, OS, and application resources.

Figure 1.9 shows the three service models available depending on the type of service you choose to get from a cloud.

FIGURE 1.9 Cloud computing service



You can see that IaaS allows the customer to manage most of the network, whereas SaaS doesn't allow any management by the customer, and PaaS is somewhere in the middle of the two. Clearly, choices can be cost driven, so the most important thing is that the customer pays only for the services or infrastructure they use.

Let's take a look at each service:

Infrastructure as a Service (IaaS): Provides only the network Delivers computer infrastructure—a platform virtualization environment—where the customer has the most control and management capability.

Platform as a Service (PaaS): Provides the operating system and the network Delivers a computing platform and solution stack, allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an application. An example is Windows Azure.

Software as a Service (SaaS): Provides the required software, operating system, and network SaaS is common application software such as databases, web servers, and email software that's hosted by the SaaS vendor. The customer accesses this software over the

Internet. Instead of having users install software on their computers or servers, the SaaS vendor owns the software and runs it on computers in its data center. Microsoft Office 365 and many Amazon Web Services (AWS) offerings are perfect examples of SaaS.

So depending on your business requirements and budget, cloud service providers market a very broad offering of cloud computing products from highly specialized offerings to a large selection of services.

What's nice here is that you're is offered a fixed price for each service that you use, which allows you to easily budget wisely for the future. It's true—at first, you'll have to spend a little cash on staff training, but with automation you can do more with less staff because administration will be easier and less complex. All of this works to free up the company resources to work on new business requirements and be more agile and innovative in the long run.

Basic Virtual Network Infrastructure

Having centralized resources is critical for today's workforce. For example, if you have your documents stored locally on your laptop and your laptop gets stolen, you're pretty much screwed unless you're doing constant local backups. That is so 2005!

After I lost my laptop and all the files for the book I was writing at the time, I swore (yes, I did that too) to never have my files stored locally again. I started using only Google Drive, OneDrive, and Dropbox for all my files, and they became my best backup friends. If I lose my laptop now, I just need to log in from any computer from anywhere to my service provider's logical drives and presto, I have all my files again. This is clearly a simple example of using cloud computing, specifically SaaS, and it's wonderful!

So cloud computing provides for the sharing of resources, lower cost operations passed to the cloud consumer, computing scaling, and the ability to dynamically add new servers without going through the procurement and deployment process.

Exam Essentials

Understand basic cloud technology. Understand cloud services such as SaaS and others and how virtualization works.

Compare and contrast collapsed core and three-tier architectures

When arranging the infrastructure devices in the network there are a number of different models that can aid in defining these relationships. In this section we'll look at two such models.

The Cisco Three-Layer Hierarchical Model

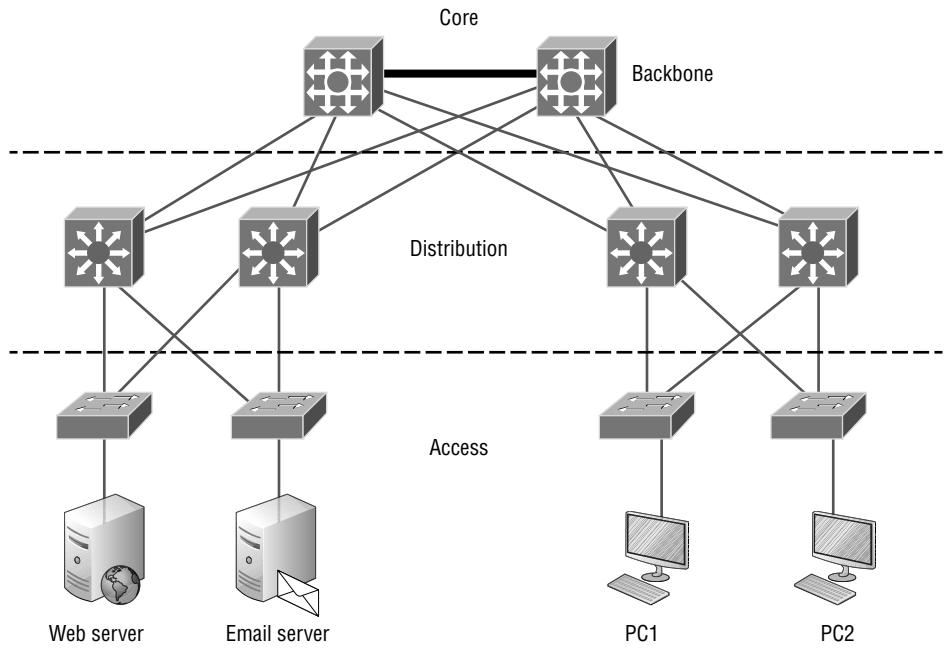
Most of us were exposed to hierarchy early in life. Anyone with older siblings learned what it was like to be at the bottom of the hierarchy. Regardless of where you first discovered the concept of hierarchy, most of us experience it in many aspects of our lives. It's *hierarchy* that helps us understand where things belong, how things fit together, and what functions go where. It brings order to otherwise complex models. If you want a pay raise, for instance, hierarchy dictates that you ask your boss, not your subordinate, because that's the person whose role it is to grant or deny your request. So basically, understanding hierarchy helps us discern where we should go to get what we need.

Hierarchy has many of the same benefits in network design that it does in other areas of life. When used properly, it makes networks more predictable and helps us define which areas should perform certain functions. Likewise, you can use tools such as access lists at certain levels in hierarchical networks and avoid them at others.

Let's face it: Large networks can be extremely complicated, with multiple protocols, detailed configurations, and diverse technologies. Hierarchy helps us summarize a complex collection of details into an understandable model, bringing order from the chaos. Then, as specific configurations are needed, the model dictates the appropriate manner in which to apply them.

The Cisco hierarchical model can help you design, implement, and maintain a scalable, reliable, cost-effective hierarchical internetwork. Cisco defines three layers of hierarchy, as shown in Figure 1.10, each with specific functions.

FIGURE 1.10 The Cisco hierarchical model



Each layer has specific responsibilities. Keep in mind that the three layers are logical and are not necessarily physical devices. Consider the OSI model, another logical hierarchy. Its seven layers describe functions but not necessarily protocols, right? Sometimes a protocol maps to more than one layer of the OSI model, and sometimes multiple protocols communicate within a single layer. In the same way, when we build physical implementations of hierarchical networks, we may have many devices in a single layer, or there may be a single device performing functions at two layers. Just remember that the definition of the layers is logical, not physical!

So let's take a closer look at each of the layers now.

The Core Layer

The *core layer* is literally the core of the network. At the top of the hierarchy, the core layer is responsible for transporting large amounts of traffic both reliably and quickly. The only purpose of the network's core layer is to switch traffic as fast as possible. The traffic transported across the core is common to a majority of users. But remember that user data is processed at the distribution layer, which forwards the requests to the core if needed.

If there's a failure in the core, *every single user* can be affected! This is why fault tolerance at this layer is so important. The core is likely to see large volumes of traffic, so speed and latency are driving concerns here. Given the function of the core, we can now consider some design specifics. Let's start with some things we don't want to do:

- We don't want 24/7 connectivity.
- Never do anything to slow down traffic. This includes making sure you don't use access lists, perform routing between virtual local area networks, or implement packet filtering.
- Don't support workgroup access here.
- Avoid expanding the core (e.g., adding routers when the internetwork grows). If performance becomes an issue in the core, give preference to upgrades over expansion.

Here's a list of things that we want to achieve as we design the core:

- Design the core for high reliability. Consider data-link technologies that facilitate both speed and redundancy, like Gigabit Ethernet with redundant links or even 10 Gigabit Ethernet.
- Design with speed in mind. The core should have very little latency.
- Select routing protocols with lower convergence times. Fast and redundant data-link connectivity is no help if your routing tables are shot!

The Distribution Layer

The *distribution layer* is sometimes referred to as the *workgroup layer* and is the communication point between the access layer and the core. The primary functions of the distribution layer are to provide routing, filtering, and WAN access and to determine how packets can

access the core, if needed. The distribution layer must determine the fastest way that network service requests are handled—for example, how a file request is forwarded to a server. After the distribution layer determines the best path, it forwards the request to the core layer if necessary. The core layer then quickly transports the request to the correct service.

The distribution layer is where we want to implement policies for the network because we are allowed a lot of flexibility in defining network operation here. There are several things that should generally be handled at the distribution layer:

- Routing
- Implementing tools (such as access lists), packet filtering, and queuing
- Implementing security and network policies, including address translation and firewalls
- Redistributing between routing protocols, including static routing
- Routing between VLANs and other workgroup support functions
- Defining broadcast and multicast domains

Key things to avoid at the distribution layer are those that are limited to functions that exclusively belong to one of the other layers!

The Access Layer

The *access layer* controls user and workgroup access to internetwork resources. The access layer is sometimes referred to as the *desktop layer*. The network resources most users need will be available locally because the distribution layer handles any traffic for remote services.

The following are some of the functions to be included at the access layer:

- Continued (from distribution layer) use of access control and policies
- Creation of separate collision domains (microsegmentation/switches)
- Workgroup connectivity into the distribution layer
- Device connectivity
- Resiliency and security services
- Advanced technology capabilities (voice/video, etc.)

Technologies like Gigabit or Fast Ethernet switching are frequently seen in the access layer.

I can't stress this enough—just because there are three separate levels does not imply three separate devices! There could be fewer or there could be more. After all, this is a *layered* approach.

Collapsed Core

In the collapsed core approach the distribution layer and the core layer are combined into a single layer, thus the name collapsed core. When using this design it is critical that

the devices operating as both distribution and core devices must exhibit the following characteristics:

- High speed paths connecting to the network
- Must be a Layer-2 aggregation point
- Must enforce routing and network access policies
- Must be capable of Intelligent network services such as QoS, and network virtualization.

The benefits are reduced cost in equipment, while the drawbacks can be slower performance and reduced network availability as compared to the three tier model.

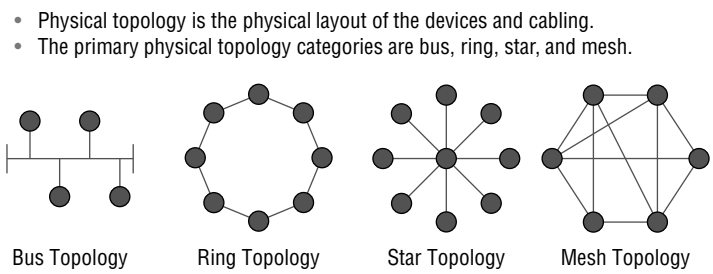
Exam Essentials

Identify the layers in the Cisco three-layer model, and describe the ideal function of each layer. The three layers in the Cisco hierarchical model are the core (responsible for transporting large amounts of traffic both reliably and quickly), distribution (provides routing, filtering, and WAN access), and access (workgroup connectivity into the distribution layer).

Compare and contrast network topologies

Understand that every type of network has both a physical and a logical topology. The physical topology of a network refers to the physical layout of the devices, but mostly the cabling and cabling layout. The logical topology defines the logical path on which the signal will travel on the physical topology. Figure 1.11 shows the four types of topologies:

FIGURE 1.11 Physical vs. Logical Topolgies



Here are the topology types, although the most common, and pretty much only network we use today is a physical star, logical bus technology, which is considered a hybrid topology (think Ethernet):

- Bus: In a bus topology, every workstation is connected to a single cable, meaning every host is directly connected to every other workstation in the network.
- Ring: In a ring topology, computers and other network devices are cabled together in a way that the last device is connected to the first to form a circle or ring.
- Star: The most common physical topology is a star topology, which is your Ethernet switching physical layout. A central cabling device (switch) connects the computers and other network devices together. This category includes star and extended star topologies. Physical connection is commonly made using twisted-pair wiring.
- Mesh: In a mesh topology, every network device is cabled together with a connection to each other. Redundant links increase reliability and self-healing. The physical connection is commonly made using fiber or twisted-pair wiring.
- Hybrid: Ethernet uses a physical star layout (cables come from all directions), and the signal travels end-to-end, like a bus route.

Exam Essentials

Describe the major physical topologies in use. Identify the differences between a physical and logical topology. List the distinguishing features of the star, ring, bus, mesh, and hybrid topologies

Select the appropriate cabling type based on implementation requirements

The *EIA/TIA* (Electronic Industries Alliance and the newer Telecommunications Industry Association) is the standards body that creates the Physical layer specifications for Ethernet. The EIA/TIA specifies that Ethernet use a *registered jack (RJ) connector* on *unshielded twisted-pair (UTP)* cabling (RJ45). But the industry is moving toward simply calling this an 8-pin modular connector.

Every Ethernet cable type that's specified by the EIA/TIA has inherent attenuation, which is defined as the loss of signal strength as it travels the length of a cable and is measured in decibels (dB). The cabling used in corporate and home markets is measured in categories. A higher-quality cable will have a higher-rated category and lower attenuation.

For example, category 5 is better than category 3 because category 5 cables have more wire twists per foot and therefore less crosstalk. Crosstalk is the unwanted signal interference from adjacent pairs in the cable.

Here is a list of some of the most common IEEE Ethernet standards, starting with 10 Mbps Ethernet:

10Base-T (IEEE 802.3) 10 Mbps using category 3 unshielded twisted pair (UTP) wiring for runs up to 100 meters. Unlike with the 10Base-2 and 10Base-5 networks, each device must connect into a hub or switch, and you can have only one host per segment or wire. It uses an RJ45 connector (8-pin modular connector) with a physical star topology and a logical bus.

100Base-TX (IEEE 802.3u) 100Base-TX, most commonly known as Fast Ethernet, uses EIA/TIA category 5, 5E, or 6 UTP two-pair wiring. One user per segment; up to 100 meters long. It uses an RJ45 connector with a physical star topology and a logical bus.

100Base-FX (IEEE 802.3u) Uses fiber cabling 62.5/125-micron multimode fiber. Point-to-point topology; up to 412 meters long. It uses ST and SC connectors, which are media-interface connectors.

1000Base-CX (IEEE 802.3z) Copper twisted-pair, called twinax, is a balanced coaxial pair that can run only up to 25 meters and uses a special 9-pin connector known as the High Speed Serial Data Connector (HSSDC). This is used in Cisco's new Data Center technologies.

1000Base-T (IEEE 802.3ab) Category 5, four-pair UTP wiring up to 100 meters long and up to 1 Gbps

1000Base-SX (IEEE 802.3z) The implementation of 1 Gigabit Ethernet running over multimode fiber-optic cable instead of copper twisted-pair cable, using short wavelength laser. Multimode fiber (MMF) using 62.5- and 50-micron core; uses an 850 nanometer (nm) laser and can go up to 220 meters with 62.5-micron, 550 meters with 50-micron

1000Base-LX (IEEE 802.3z) Single-mode fiber that uses a 9-micron core and 1300 nm laser and can go from 3 kilometers up to 10 kilometers

1000Base-ZX (Cisco standard) 1000BaseZX, or 1000Base-ZX, is a Cisco specified standard for Gigabit Ethernet communication. 1000BaseZX operates on ordinary single-mode fiber-optic links with spans up to 43.5 miles (70 km).

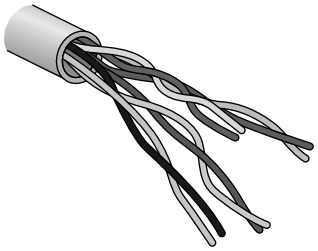
10GBase-T (802.3.an) 10GBase-T is a standard proposed by the IEEE 802.3an committee to provide 10 Gbps connections over conventional UTP cables, (category 5e, 6, or 7 cables). 10GBase-T allows the conventional RJ45 used for Ethernet LANs and can support signal transmission at the full 100-meter distance specified for LAN wiring.

A discussion about Ethernet cabling is an important one, especially if you are planning on taking the Cisco exams. You need to really understand the following three types of cables:

- Straight-through cable
- Crossover cable
- Rolled cable

We will look at each in the following sections, but first, let's take a look at the most common Ethernet cable used today, the category 5 Enhanced Unshielded Twisted Pair (UTP), shown in Figure 1.12.

FIGURE 1.12 Category 5 Enhanced UTP cable



The category 5 Enhanced UTP cable can handle speeds up to a gigabit with a distance of up to 100 meters. Typically we'd use this cable for 100 Mbps and category 6 for a gigabit, but the category 5 Enhanced is rated for gigabit speeds and category 6 is rated for 10 Gbps!

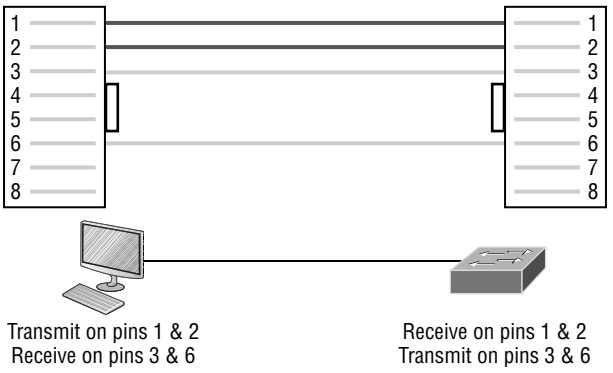
Straight-Through Cable

The *straight-through cable* is used to connect the following devices:

- Host to switch or hub
- Router to switch or hub

Four wires are used in a straight-through cable to connect Ethernet devices. It's relatively simple to create this type, and Figure 1.13 shows the four wires used in a straight-through Ethernet cable.

FIGURE 1.13 Straight-through Ethernet cable



Notice that only pins 1, 2, 3, and 6 are used. Just connect 1 to 1, 2 to 2, 3 to 3, and 6 to 6 and you'll be up and networking in no time. However, remember that this would be a 10/100 Mbps Ethernet-only cable and wouldn't work with gigabit, voice, or other LAN or WAN technology.

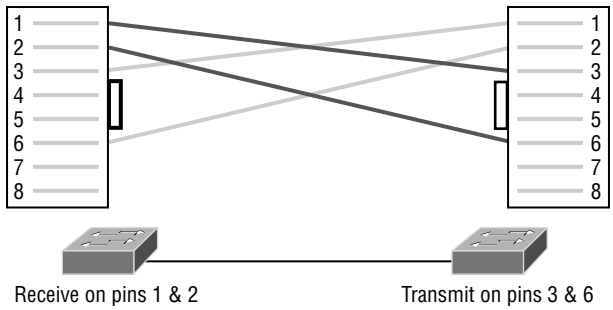
Crossover Cable

The *crossover cable* can be used to connect the following devices:

- Switch to switch
- Hub to hub
- Host to host
- Hub to switch
- Router direct to host
- Router to router

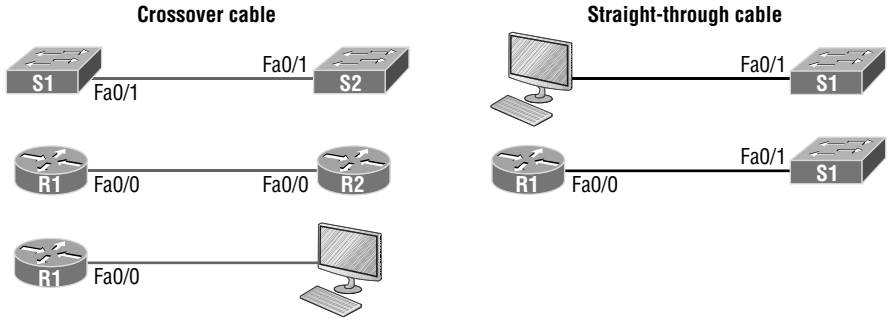
The same four wires used in the straight-through cable are used in this cable—we just connect different pins together. Figure 1.14 shows how the four wires are used in a cross-over Ethernet cable.

FIGURE 1.14 Crossover Ethernet cable



Notice that instead of connecting 1 to 1, 2 to 2, and so on, here we connect pins 1 to 3 and 2 to 6 on each side of the cable. Figure 1.15 shows some typical uses of straight-through and crossover cables.

FIGURE 1.15 Typical uses for straight-through and cross-over Ethernet cables



The crossover examples in Figure 1.15 are switch port to switch port, router Ethernet port to router Ethernet port, and router Ethernet port to PC Ethernet port. For the straight-through examples I used PC Ethernet to switch port and router Ethernet port to switch port.

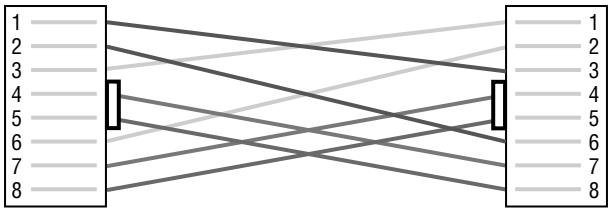
It's very possible to connect a straight-through cable between two switches, and it will start working because of autodetect mechanisms called auto-mdix. But be advised that the CCNA objectives do not typically consider autodetect mechanisms valid between devices!

UTP Gigabit Wiring (1000Base-T)

In the previous examples of 10Base-T and 100Base-T UTP wiring, only two wire pairs were used, but that is not good enough for Gigabit UTP transmission.

1000Base-T UTP wiring (Figure 1.16) requires four wire pairs and uses more advanced electronics so that each and every pair in the cable can transmit simultaneously. Even so, gigabit wiring is almost identical to my earlier 10/100 example, except that we'll use the other two pairs in the cable.

FIGURE 1.16 UTP Gigabit crossover Ethernet cable



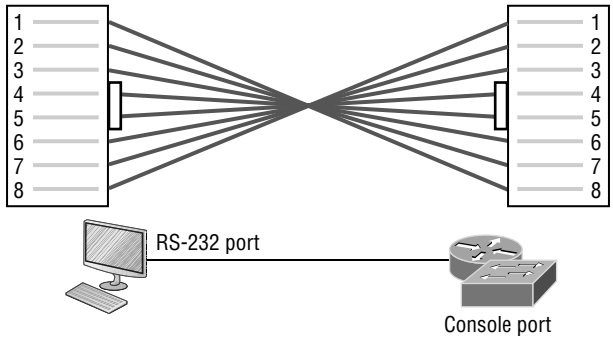
For a straight-through cable it's still 1 to 1, 2 to 2, and so on up to pin 8. And in creating the gigabit crossover cable, you'd still cross 1 to 3 and 2 to 6, but you would add 4 to 7 and 5 to 8—pretty straightforward!

Rolled Cable

Although *rolled cable* isn't used to connect any Ethernet connections together, you can use a rolled Ethernet cable to connect a host EIA-TIA 232 interface to a router console serial communication (COM) port.

If you have a Cisco router or switch, you would use this cable to connect your PC, Mac, or a device like an iPad to the Cisco hardware. Eight wires are used in this cable to connect serial devices, although not all eight are used to send information, just as in Ethernet networking. Figure 1.17 shows the eight wires used in a rolled cable.

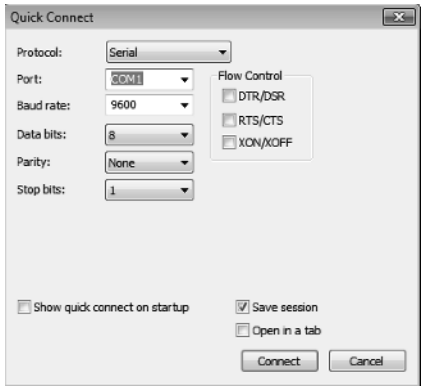
FIGURE 1.17 Rolled Ethernet cable



These are probably the easiest cables to make because you just cut the end off on one side of a straight-through cable, turn it over, and put it back on—with a new connector, of course!

Okay, once you have the correct cable connected from your PC to the Cisco router or switch console port, you can start your emulation program such as putty or SecureCRT to create a console connection and configure the device. Set the configuration as shown in Figure 1.18.

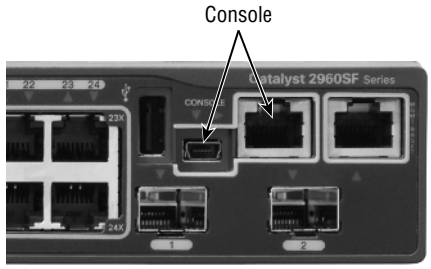
FIGURE 1.18 Configuring your console emulation program



Notice that Baud Rate is set to 9600, Data Bits to 8, Parity to None, and no Flow Control options are set. At this point, you can click Connect and press the Enter key and you should be connected to your Cisco device console port.

Figure 1.19 shows a nice new 2960 switch with two console ports.

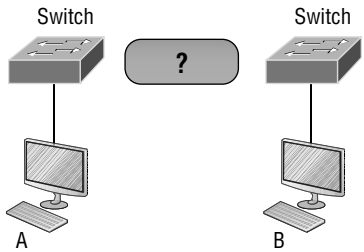
FIGURE 1.19 A Cisco 2960 console connections



Notice there are two console connections on this new switch—a typical original RJ45 connection and the newer mini type-B USB console. Remember that the new USB port supersedes the RJ45 port if you just happen to plug into both at the same time, and the USB port can have speeds up to 115,200 Kbps, which is awesome if you have to use Xmodem to update a Cisco Internal Operating System (IOS). I've even seen some cables that work on iPhones and iPads and allow them to connect to these mini USB ports!

Now that you've seen the various RJ45 unshielded twisted-pair (UTP) cables, what type of cable is used between the switches in Figure 1.20?

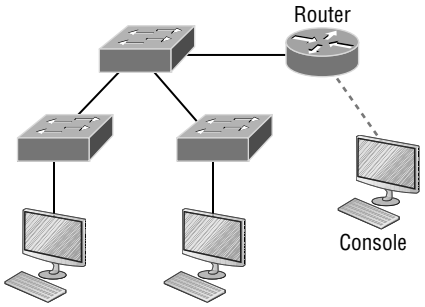
FIGURE 1.20 RJ45 UTP cable question #1



In order for host A to ping host B, you need a crossover cable to connect the two switches together. But what types of cables are used in the network shown in Figure 1.21?

In Figure 1.21, there's a whole menu of cables in use. For the connection between the switches, we'd obviously use a crossover cable like we saw in Figure 1.14. The trouble is that you must understand that we have a console connection that uses a rolled cable. Plus, the connection from the router to the switch is a straight-through cable, as is true for the hosts to the switches. Keep in mind that if we had a serial connection, which we don't, we would use a V.35 to connect us to a WAN.

FIGURE 1.21 RJ45 UTP cable question #2



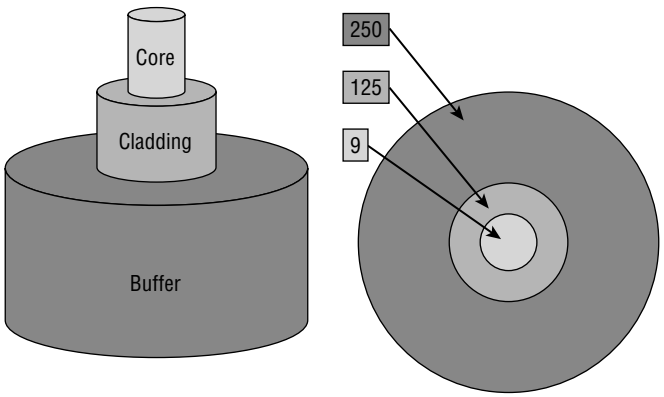
Fiber Optic

Fiber-optic cabling has been around for a long time and has some solid standards. The cable allows for very fast transmission of data, is made of glass (or even plastic!), is very thin, and works as a waveguide to transmit light between two ends of the fiber. Fiber optics has been used to go very long distances, as in intercontinental connections, but it is becoming more and more popular in Ethernet LAN networks due to the fast speeds available and because, unlike UTP, it's immune to interference like crosstalk.

Some main components of this cable are the core and the cladding. The core will hold the light and the cladding confines the light in the core. The tighter the cladding, the smaller the core, and when the core is small, less light will be sent, but it can go faster and farther!

In Figure 1.22 you can see that there is a 9-micron core, which is very small and can be measured against a human hair, which is 50 microns.

FIGURE 1.22 Typical fiber cable

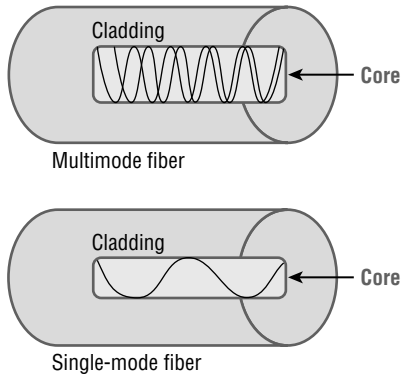


Dimensions are in um (10^{-6} meters). Not to scale.

The cladding is 125 microns, which is actually a fiber standard that allows manufacturers to make connectors for all fiber cables. The last piece of this cable is the buffer, which is there to protect the delicate glass.

There are two major types of fiber optics: single-mode and multimode. Figure 1.23 shows the differences between multimode and single-mode fibers.

FIGURE 1.23 Multimode and single-mode fibers



Single-mode is more expensive, has a tighter cladding, and can go much farther distances than multimode. The difference comes in the tightness of the cladding, which makes a smaller core, meaning that only one mode of light will propagate down the fiber. Multimode is looser and has a larger core so it allows multiple light particles to travel down the glass. These particles have to be put back together at the receiving end, so distance is less than that with single-mode fiber, which allows only very few light particles to travel down the fiber.

There are about 70 different connectors for fiber, and Cisco uses a few different types. Looking back at Figure 1.19, the two bottom ports are referred to as Small Form-Factor Pluggables, or SFPs.

Exam Essentials

Identify the major Ethernet cable types. Describe Ethernet cable types, their limitations, connectors, and application.

Apply cable types to the proper scenario. This includes straight through, crossover, and rollover cables.

Identify the major Fiber cable types. Describe Fiber cable types, their limitations, connectors, and application.

Apply troubleshooting methodologies to resolve problems

When attempting to troubleshoot, any issue following a troubleshooting process developed over years by experienced technicians can speed the solution of the issue. The Cisco troubleshooting methodology uses the following steps:

- Define the problem.
- Gather information.
- Analyze information.
- Eliminate potential causes.
- Propose hypothesis.
- Test hypothesis.
- Solve problem and document solution.

In the process of working this list, issues can be approached in three ways:

- **Top-down method:** Start at the top of the OSI model.
- **Bottom-up method:** Start at the bottom of the OSI model.
- **Divide and conquer:** Start in the middle of the OSI model.

As you identify individual issues you can use the following methods to track down the source:

- Perform comparison method: Compare the configuration of the device with an issue with a like device that is functioning properly.
- Follow the path method: Trace the path of the problematic communication, checking each device in the path as you go.
- Swap components method: Replace components with like components to see if the issue follows the component.

Exam Essentials

List the steps in the Cisco troubleshooting methodology. Know the 7 steps in this process.

Identify potential approaches to solving issues. These include top-down, bottom-up, and divide and conquer.

Describe methods to locate communication issues. These include follow the path, swap components, and perform comparison methods.

Configure, verify, and troubleshoot IPv4 addressing and subnetting

It is critical to be able to configure IP addresses on devices in such a way that those devices that should be in the same network are in the same network. This requires a complete understanding of IP addressing and subnetting. That will be the topic of this section.

The Hierarchical IP Addressing Scheme

An IP address consists of 32 bits of information. These bits are divided into four sections, referred to as octets or bytes, with each containing 1 byte (8 bits). You can depict an IP address using one of three methods:

- Dotted-decimal, as in 172.16.30.56
- Binary, as in 10101100.00010000.00011110.00111000
- Hexadecimal, as in AC.10.1E.38

All these examples represent the same IP address. Pertaining to IP addressing, hexadecimal isn't used as often as dotted-decimal or binary, but you still might find an IP address stored in hexadecimal in some programs.

The 32-bit IP address is a structured or hierarchical address, as opposed to a flat or nonhierarchical address. Although either type of addressing scheme could have been used, *hierarchical addressing* was chosen for a good reason. The advantage of this scheme is that it can handle a large number of addresses, namely 4.3 billion (a 32-bit address space with two possible values for each position—either 0 or 1—gives you 2^{32}, or 4,294,967,296). The disadvantage of the flat addressing scheme, and the reason it's not used for IP addressing, relates to routing. If every address were unique, all routers on the Internet would need to store the address of each and every machine on the Internet. This would make efficient routing impossible, even if only a fraction of the possible addresses were used!

The solution to this problem is to use a two- or three-level hierarchical addressing scheme that is structured by network and host or by network, subnet, and host.

This two- or three-level scheme can also be compared to a telephone number. The first section, the area code, designates a very large area. The second section, the prefix, narrows the scope to a local calling area. The final segment, the customer number, zooms in on the specific connection. IP addresses use the same type of layered structure. Rather than all 32 bits being treated as a unique identifier, as in flat addressing, a part of the address is designated as the network address and the other part is designated as either the subnet and host or just the node address.

Next, we'll cover IP network addressing and the different classes of address we can use to address our networks.

Network Addressing

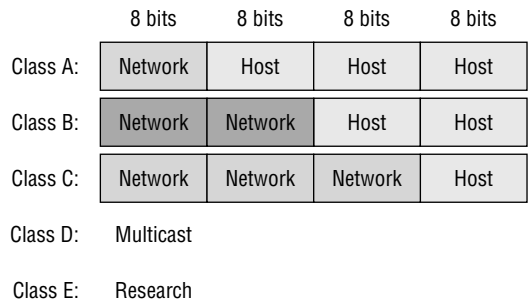
The *network address* (which can also be called the network number) uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. For example, in the IP address 172.16.30.56, 172.16 is the network address.

The *node address* is assigned to, and uniquely identifies, each machine on a network. This part of the address must be unique because it identifies a particular machine—an individual—as opposed to a network, which is a group. This number can also be referred to as a *host address*. In the sample IP address 172.16.30.56, the 30.56 specifies the node address.

The designers of the Internet decided to create classes of networks based on network size. For the small number of networks possessing a very large number of nodes, they created the rank *Class A network*. At the other extreme is the *Class C network*, which is reserved for the numerous networks with a small number of nodes. The class distinction for networks between very large and very small is predictably called the *Class B network*.

Subdividing an IP address into a network and node address is determined by the class designation of one's network. Figure 1.24 summarizes the three classes of networks used to address hosts—a subject I'll explain in much greater detail throughout this chapter.

FIGURE 1.24 Summary of the three classes of networks



To ensure efficient routing, Internet designers defined a mandate for the leading-bits section of the address for each different network class. For example, since a router knows that a Class A network address always starts with a 0, the router might be able to speed a packet on its way after reading only the first bit of its address. This is where the address schemes define the difference between a Class A, a Class B, and a Class C address. Coming up, I'll discuss the differences between these three classes, followed by a discussion of the Class D and Class E addresses. Classes A, B, and C are the only ranges that are used to address hosts in our networks.

Network Address Range: Class A

The designers of the IP address scheme decided that the first bit of the first byte in a Class A network address must always be off, or 0. This means a Class A address must be between 0 and 127 in the first byte, inclusive.

Consider the following network address:

```
0xxxxxxx
```

If we turn the other 7 bits all off and then turn them all on, we'll find the Class A range of network addresses:

```
00000000 = 0
01111111 = 127
```

So, a Class A network is defined in the first octet between 0 and 127, and it can't be less or more. Understand that 0 and 127 are not valid in a Class A network because they're reserved addresses, which I'll explain soon.

Network Address Range: Class B

In a Class B network, the RFCs state that the first bit of the first byte must always be turned on but the second bit must always be turned off. If you turn the other 6 bits all off and then all on, you will find the range for a Class B network:

```
10000000 = 128
10111111 = 191
```

As you can see, a Class B network is defined when the first byte is configured from 128 to 191.

Network Address Range: Class C

For Class C networks, the RFCs define the first 2 bits of the first octet as always turned on, but the third bit can never be on. Following the same process as the previous classes, convert from binary to decimal to find the range. Here's the range for a Class C network:

```
11000000 = 192
11011111 = 223
```

So, if you see an IP address that starts at 192 and goes to 223, you'll know it is a Class C IP address.

Network Address Ranges: Classes D and E

The addresses between 224 to 255 are reserved for Class D and E networks. Class D (224–239) is used for multicast addresses and Class E (240–255) for scientific purposes, but I'm not going into these types of addresses because they are beyond the scope of knowledge you need to gain from this book.

Network Addresses: Special Purpose

Some IP addresses are reserved for special purposes, so network administrators can't ever assign these addresses to nodes. Table 1.2 lists the members of this exclusive little club and the reasons why they're included in it.

TABLE 1.2 Reserved IP addresses

Address	Function
Network address of all 0s	Interpreted to mean "this network or segment"
Network address of all 1s	Interpreted to mean "all networks"
Network 127.0.0.1	Reserved for loopback tests. Designates the local node and allows that node to send a test packet to itself without generating network traffic
Node address of all 0s	Interpreted to mean "network address" or any host on a specified network
Node address of all 1s	Interpreted to mean "all nodes" on the specified network; for example, 128.2.255.255 means "all nodes" on network 128.2 (Class B address)
Entire IP address set to all 0s	Used by Cisco routers to designate the default route. Could also mean "any network"
Entire IP address set to all 1s (same as 255.255.255.255)	Broadcast to all nodes on the current network; sometimes called an "all 1s broadcast" or local broadcast

Class A Addresses

In a Class A network address, the first byte is assigned to the network address and the three remaining bytes are used for the node addresses. The Class A format is as follows:

```
network.node.node.node
```

For example, in the IP address 49.22.102.70, the 49 is the network address and 22.102.70 is the node address. Every machine on this particular network would have the distinctive network address of 49.

Class A network addresses are 1 byte long, with the first bit of that byte reserved and the 7 remaining bits available for manipulation (addressing). As a result, the maximum number of Class A networks that can be created is 128. Why? Because each of the 7 bit positions can be either a 0 or a 1, thus 2^7, or 128.

To complicate matters further, the network address of all 0s (0000 0000) is reserved to designate the default route (see Table 1.2 in the previous section). Additionally, the address 127, which is reserved for diagnostics, can't be used either, which means that you can really only use the numbers 1 to 126 to designate Class A network addresses. This means the actual number of usable Class A network addresses is 128 minus 2, or 126.

The IP address 127.0.0.1 is used to test the IP stack on an individual node and cannot be used as a valid host address. However, the loopback address creates a shortcut method for TCP/IP applications and services that run on the same device to communicate with each other.

Each Class A address has 3 bytes (24-bit positions) for the node address of a machine. This means there are 2^{24}—or 16,777,216—unique combinations and, therefore, precisely that many possible unique node addresses for each Class A network. Because node addresses with the two patterns of all 0s and all 1s are reserved, the actual maximum usable number of nodes for a Class A network is 2^{24} minus 2, which equals 16,777,214. Either way, that's a huge number of hosts on a single network segment!

Class A Valid Host IDs

Here's an example of how to figure out the valid host IDs in a Class A network address:

- All host bits off is the network address: 10.0.0.0.
- All host bits on is the broadcast address: 10.255.255.255.

The valid hosts are the numbers in between the network address and the broadcast address: 10.0.0.1 through 10.255.255.254. Notice that 0s and 255s can be valid host IDs. All you need to remember when trying to find valid host addresses is that the host bits can't all be turned off or on at the same time.

Class B Addresses

In a Class B network address, the first 2 bytes are assigned to the network address and the remaining 2 bytes are used for node addresses. The format is as follows:

```
network.network.node.node
```

For example, in the IP address 172.16.30.56, the network address is 172.16 and the node address is 30.56.

With a network address being 2 bytes (8 bits each), you get 2^{16} unique combinations. But the Internet designers decided that all Class B network addresses should start with the binary digit 1, then 0. This leaves 14 bit positions to manipulate, therefore 16,384, or 2^{14} unique Class B network addresses.

A Class B address uses 2 bytes for node addresses. This is 2^{16} minus the two reserved patterns of all 0s and all 1s for a total of 65,534 possible node addresses for each Class B network.

Class B Valid Host IDs

Here's an example of how to find the valid hosts in a Class B network:

- All host bits turned off is the network address: 172.16.0.0.
- All host bits turned on is the broadcast address: 172.16.255.255.

The valid hosts would be the numbers in between the network address and the broadcast address: 172.16.0.1 through 172.16.255.254.

Class C Addresses

The first 3 bytes of a Class C network address are dedicated to the network portion of the address, with only 1 measly byte remaining for the node address. Here's the format:

```
network.network.network.node
```

Using the example IP address 192.168.100.102, the network address is 192.168.100 and the node address is 102.

In a Class C network address, the first three bit positions are always the binary 110. The calculation is as follows: 3 bytes, or 24 bits, minus 3 reserved positions leaves 21 positions. Hence, there are 2^{21}, or 2,097,152, possible Class C networks.

Each unique Class C network has 1 byte to use for node addresses. This leads to 2^8, or 256, minus the two reserved patterns of all 0s and all 1s, for a total of 254 node addresses for each Class C network.

Class C Valid Host IDs

Here's an example of how to find a valid host ID in a Class C network:

- All host bits turned off is the network ID: 192.168.100.0.
- All host bits turned on is the broadcast address: 192.168.100.255.

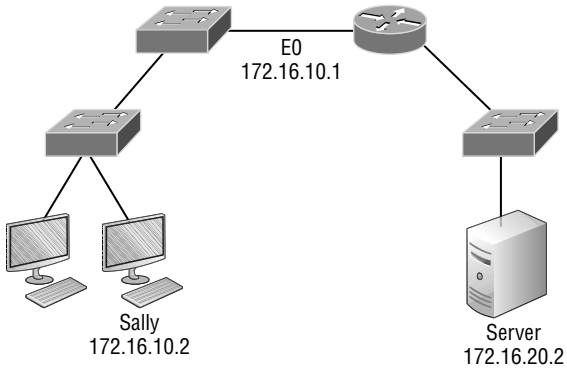
The valid hosts would be the numbers in between the network address and the broadcast address: 192.168.100.1 through 192.168.100.254.

Troubleshooting IPv4 Addressing and Subnetting

Because running into trouble now and then in networking is a given, being able to troubleshoot IP addressing is clearly a vital skill. I'm not being negative here—just realistic. The positive side to this is that if you're the one equipped with the tools to diagnose and clear up the inevitable trouble, you get to be the hero when you save the day! Even better? You can usually fix an IP network regardless of whether you're on site or at home!

So this is where I'm going to show you the "Cisco way" of troubleshooting IP addressing. Let's use Figure 1.25 as an example of your basic IP trouble—poor Sally can't log in to the Windows server. Do you deal with this by calling the Microsoft team to tell them their server is a pile of junk and causing all your problems? Though tempting, a better approach is to first double-check and verify your network instead.

FIGURE 1.25 Basic IP troubleshooting



Okay, let's get started by going through the troubleshooting steps that Cisco recommends. They're pretty simple, but important nonetheless. Pretend you're at a customer host and they're complaining that they can't communicate to a server that just happens to be on a remote network. Here are the four troubleshooting steps Cisco recommends:

1. Open a Command window and ping 127.0.0.1. This is the diagnostic, or loopback, address, and if you get a successful ping, your IP stack is considered initialized. If it fails, then you have an IP stack failure and need to reinstall TCP/IP on the host.

```
C:\>ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. From the Command window, ping the IP address of the local host (we'll assume correct configuration here, but always check the IP configuration too!). If that's successful, your network interface card (NIC) is functioning. If it fails, there is a problem with the NIC. Success here doesn't just mean that a cable is plugged into the NIC, only that the IP protocol stack on the host can communicate to the NIC via the LAN driver.

```
C:\>ping 172.16.10.2
Pinging 172.16.10.2 with 32 bytes of data:
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
```

```
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3. From the Command window, ping the default gateway (router). If the ping works, it means that the NIC is plugged into the network and can communicate on the local network. If it fails, you have a local physical network problem that could be anywhere from the NIC to the router.

```
C:\>ping 172.16.10.1
Pinging 172.16.10.1 with 32 bytes of data:
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

4. If steps 1 through 3 were successful, try to ping the remote server. If that works, then you know that you have IP communication between the local host and the remote server. You also know that the remote physical network is working.

```
C:\>ping 172.16.20.2
Pinging 172.16.20.2 with 32 bytes of data:
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

If the user still can't communicate with the server after steps 1 through 4 have been completed successfully, you probably have some type of name resolution problem and need to check your Domain Name System (DNS) settings. But if the ping to the remote server fails,

then you know you have some type of remote physical network problem and need to go to the server and work through steps 1 through 3 until you find the snag.

Before we move on to determining IP address problems and how to fix them, I just want to mention some basic commands that you can use to help troubleshoot your network from both a PC and a Cisco router. Keep in mind that though these commands may do the same thing, they're implemented differently.

`ping` Uses ICMP echo request and replies to test if a node IP stack is initialized and alive on the network

`traceroute` Displays the list of routers on a path to a network destination by using TTL time-outs and ICMP error messages. This command will not work from a command prompt.

`tracert` Same function as `traceroute`, but it's a Microsoft Windows command and will not work on a Cisco router

`arp -a` Displays IP-to-MAC-address mappings on a Windows PC

`show ip arp` Same function as `arp -a`, but displays the ARP table on a Cisco router. Like the commands `traceroute` and `tracert`, `arp -a` and `show ip arp` are not interchangeable through DOS and Cisco.

`ipconfig /all` Used only from a Windows command prompt; shows you the PC network configuration

Once you've gone through all these steps and, if necessary, used the appropriate commands, what do you do when you find a problem? How do you go about fixing an IP address configuration error? Time to cover the next step—determining and fixing the issue at hand!

Determining IP Address Problems

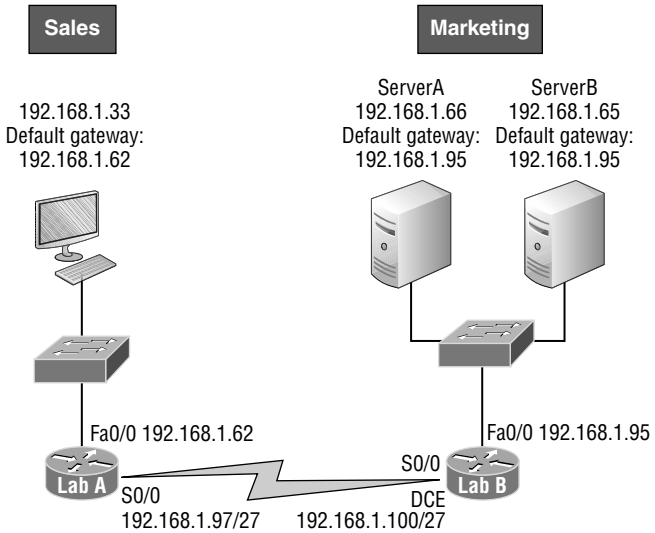
It's common for a host, router, or other network device to be configured with the wrong IP address, subnet mask, or default gateway. Because this happens way too often, you must know how to find and fix IP address configuration errors.

A good way to start is to draw out the network and IP addressing scheme. If that's already been done, consider yourself lucky because though sensible, it's rarely done. Even if it is, it's usually outdated or inaccurate anyway. So either way, it's a good idea to bite the bullet and start from scratch.

Once you have your network accurately drawn out, including the IP addressing scheme, you need to verify each host's IP address, mask, and default gateway address to establish the problem. Of course, this is assuming that you don't have a physical layer problem, or if you did, that you've already fixed it.

Let's check out the example illustrated in Figure 1.26.

FIGURE 1.26 IP address problem 1



A user in the sales department calls and tells you that she can't get to ServerA in the marketing department. You ask her if she can get to ServerB in the marketing department, but she doesn't know because she doesn't have rights to log on to that server. What do you do?

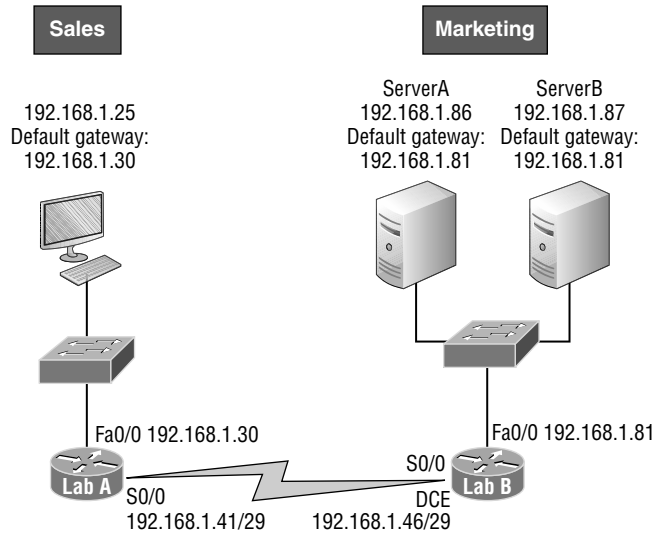
First, guide your user through the four troubleshooting steps you learned in the preceding section. Okay—let's say steps 1 through 3 work but step 4 fails. By looking at the figure, can you determine the problem? Look for clues in the network drawing. First, the WAN link between the Lab A router and the Lab B router shows the mask as a /27. You should already know that this mask is 255.255.255.224 and determine that all networks are using this mask. The network address is 192.168.1.0. What are our valid subnets and hosts? 256 – 224 = 32, so this makes our subnets 0, 32, 64, 96, 128, etc. So, by looking at the figure, you can see that subnet 32 is being used by the sales department. The WAN link is using subnet 96, and the marketing department is using subnet 64.

Now you've got to establish what the valid host ranges are for each subnet. From what you learned at the beginning of this chapter, you should now be able to easily determine the subnet address, broadcast addresses, and valid host ranges. The valid hosts for the Sales LAN are 33 through 62, and the broadcast address is 63 because the next subnet is 64, right? For the Marketing LAN, the valid hosts are 65 through 94 (broadcast 95), and for the WAN link, 97 through 126 (broadcast 127). By closely examining the figure, you can determine that the default gateway on the Lab B router is incorrect. That address is the broadcast address for subnet 64, so there's no way it could be a valid host!

If you tried to configure that address on the Lab B router interface, you'd receive a `bad mask` error. Cisco routers don't let you type in subnet and broadcast addresses as valid hosts!

Did you get all that? Let's try another one to make sure. Figure 1.27 shows a network problem.

FIGURE 1.27 IP address problem 2



A user in the Sales LAN can't get to ServerB. You have the user run through the four basic troubleshooting steps and find that the host can communicate to the local network but not to the remote network. Find and define the IP addressing problem.

If you went through the same steps used to solve the last problem, you can see that first, the WAN link again provides the subnet mask to use— /29, or 255.255.255.248. Assuming classful addressing, you need to determine what the valid subnets, broadcast addresses, and valid host ranges are to solve this problem.

The 248 mask is a block size of 8 (256 – 248 = 8, as discussed in Chapter 4), so the subnets both start and increment in multiples of 8. By looking at the figure, you see that the Sales LAN is in the 24 subnet, the WAN is in the 40 subnet, and the Marketing LAN is in the 80 subnet. Can you see the problem yet? The valid host range for the Sales LAN is 25–30, and the configuration appears correct. The valid host range for the WAN link is 41–46, and this also appears correct. The valid host range for the 80 subnet is 81–86, with a broadcast address of 87 because the next subnet is 88. ServerB has been configured with the broadcast address of the subnet.

Okay, now that you can figure out misconfigured IP addresses on hosts, what do you do if a host doesn't have an IP address and you need to assign one? What you need to do is scrutinize the other hosts on the LAN and figure out the network, mask, and default gateway. Let's take a look at a couple of examples of how to find and apply valid IP addresses to hosts.

You need to assign a server and router IP addresses on a LAN. The subnet assigned on that segment is 192.168.20.24/29. The router needs to be assigned the first usable address

and the server needs the last valid host ID. What is the IP address, mask, and default gateway assigned to the server?

To answer this, you must know that a /29 is a 255.255.255.248 mask, which provides a block size of 8. The subnet is known as 24, the next subnet in a block of 8 is 32, so the broadcast address of the 24 subnet is 31 and the valid host range is 25–30.

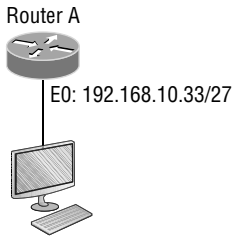
Server IP address: 192.168.20.30

Server mask: 255.255.255.248

Default gateway: 192.168.20.25 (router's IP address)

Take a look at Figure 1.28 and solve this problem.

FIGURE 1.28 Find the valid host #1.



Look at the router's IP address on Ethernet0. What IP address, subnet mask, and valid host range could be assigned to the host?

The IP address of the router's Ethernet0 is 192.168.10.33/27. As you already know, a /27 is a 224 mask with a block size of 32. The router's interface is in the 32 subnet. The next subnet is 64, so that makes the broadcast address of the 32 subnet 63 and the valid host range 33–62.

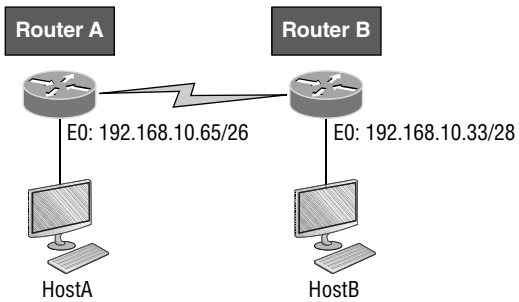
Host IP address: 192.168.10.34–62 (any address in the range except for 33, which is assigned to the router)

Mask: 255.255.255.224

Default gateway: 192.168.10.33

Figure 1.29 shows two routers with Ethernet configurations already assigned. What are the host addresses and subnet masks of HostA and HostB?

FIGURE 1.29 Find the valid host #2.



Router A has an IP address of 192.168.10.65/26 and Router B has an IP address of 192.168.10.33/28. What are the host configurations? Router A Ethernet0 is in the 192.168.10.64 subnet and Router B Ethernet0 is in the 192.168.10.32 network.

Host A IP address: 192.168.10.66–126

Host A mask: 255.255.255.192

Host A default gateway: 192.168.10.65

Host B IP address: 192.168.10.34–46

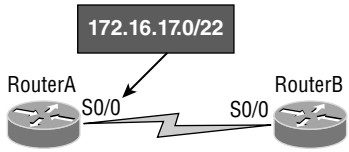
Host B mask: 255.255.255.240

Host B default gateway: 192.168.10.33

Just a couple more examples before you can put this chapter behind you—hang in there!

Figure 1.30 shows two routers. You need to configure the S0/0 interface on RouterA. The network assigned to the serial link is 172.16.17.0/22. What IP address can be assigned?

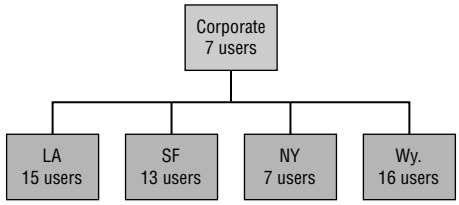
FIGURE 1.30 Find the valid host address #3.



First, know that a /22 CIDR is 255.255.252.0, which makes a block size of 4 in the third octet. Since 17 is listed, the available range is 16.1 through 19.254, so in this example, the IP address S0/0 could be 172.16.18.255 since that's within the range.

Okay, last one! You need to find a classful network address that has one Class C network ID and you need to provide one usable subnet per city while allowing enough usable host addresses for each city specified in Figure 1.31. What is your mask?

FIGURE 1.31 Find the valid subnet mask.



Actually, this is probably the easiest thing you've done all day! I count 5 subnets needed, and the Wyoming office needs 16 users—always look for the network that needs the most hosts! What block size is needed for the Wyoming office? Your answer is 32. You can't use a block size of 16 because you always have to subtract 2. What mask provides you with a block size of 32? 224 is your answer because this provides 8 subnets, each with 30 hosts.

Exam Essentials

Define the Class A IP address range. The IP range for a Class A network is 1–126. This provides 8 bits of network addressing and 24 bits of host addressing by default.

Define the Class B IP address range. The IP range for a Class B network is 128–191. Class B addressing provides 16 bits of network addressing and 16 bits of host addressing by default.

Define the Class C IP address range. The IP range for a Class C network is 192 through 223. Class C addressing provides 24 bits of network addressing and 8 bits of host addressing by default.

Remember the four diagnostic steps. The four simple steps that Cisco recommends for troubleshooting are ping the loopback address, ping the NIC, ping the default gateway, and ping the remote device.

Identify and mitigate an IP addressing problem. Once you go through the four troubleshooting steps that Cisco recommends, you must be able to determine the IP addressing problem by drawing out the network and finding the valid and invalid hosts addressed in your network.

Understand the troubleshooting tools that you can use from your host and a Cisco router. The `ping 127.0.0.1` command tests your local IP stack, and `tracert` is a Windows command to track the path a packet takes through an internetwork to a destination. Cisco routers use the command `traceroute`, or just `trace` for short. Don't confuse the Windows and Cisco commands. Although they produce the same output, they don't work from the same prompts. The command `ipconfig /all` will display your PC network configuration from a DOS prompt, and `arp -a` (again from a DOS prompt) will display IP-to-MAC-address mapping on a Windows PC.

Compare and contrast IPv4 address types

There are a number of different IPv4 address types with which you must be familiar for the exam. Included in these are the following:

Loopback (localhost) Used to test the IP stack on the local computer. Can be any address from 127.0.0.1 through 127.255.255.254.

Layer 2 broadcasts These are sent to all nodes on a LAN.

Broadcasts (layer 3) These are sent to all nodes on the network.

Unicast This is an address for a single interface, and these are used to send packets to a single destination host.

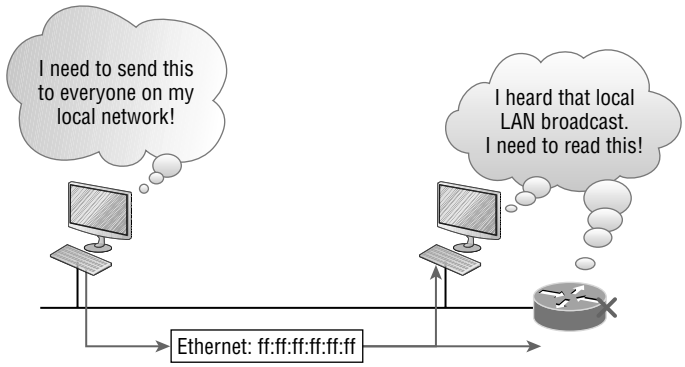
Multicast These are packets sent from a single source and transmitted to many devices on different networks. Referred to as "one-to-many."

Layer 2 Broadcasts

First, understand that layer 2 broadcasts are also known as hardware broadcasts—they only go out on a LAN, but they don't go past the LAN boundary (router).

The typical hardware address is 6 bytes (48 bits) and looks something like 45:AC:24:E3:60:A5. The broadcast would be all 1s in binary, which would be all *F*s in hexadecimal, as in ff:ff:ff:ff:ff:ff and shown in Figure 1.32.

FIGURE 1.32 Local layer 2 broadcasts



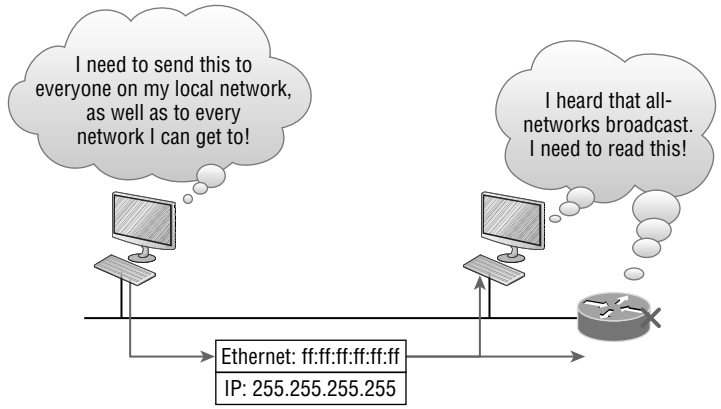
Every network interface card (NIC) will receive and read the frame, including the router, since this was a layer 2 broadcast, but the router would never, ever forward this!

Layer 3 Broadcasts

Then there are the plain old broadcast addresses at layer 3. Broadcast messages are meant to reach all hosts on a broadcast domain. These are the network broadcasts that have all host bits on.

Here's an example that you're already familiar with: The network address of 172.16.0.0 255.255.0.0 would have a broadcast address of 172.16.255.255—all host bits on. Broadcasts can also be "any network and all hosts," as indicated by 255.255.255.255, and shown in Figure 1.33.

FIGURE 1.33 Layer 3 broadcasts



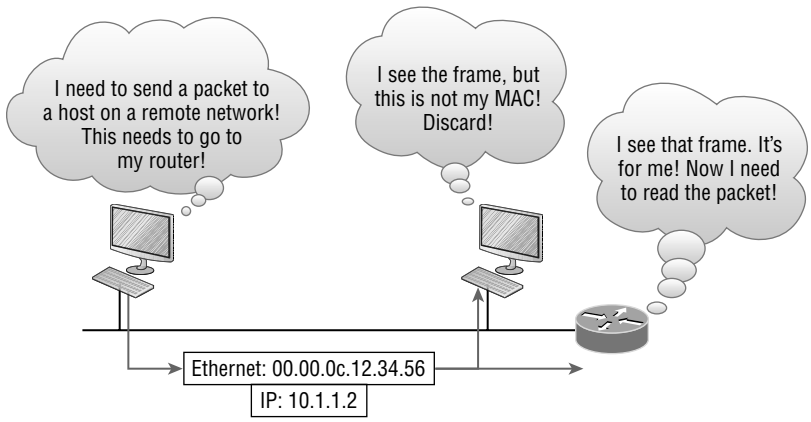
In Figure 1.33, all hosts on the LAN will get this broadcast on their NIC, including the router, but by default the router would never forward this packet.

Unicast Address

A unicast is defined as a single IP address that's assigned to a network interface card and is the destination IP address in a packet—in other words, it's used for directing packets to a specific host.

In Figure 1.34, both the MAC address and the destination IP address are for a single NIC on the network. All hosts on the broadcast domain would receive this frame and accept it. Only the destination NIC of 10.1.1.2 would accept the packet; the other NICs would discard the packet.

FIGURE 1.34 Unicast address

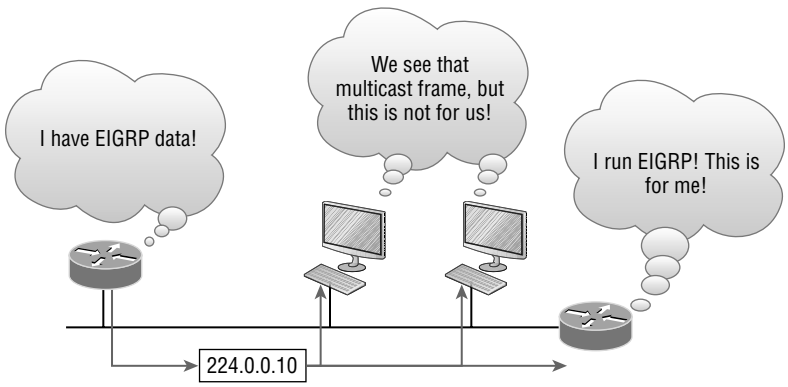


Multicast Address

Multicast is a different beast entirely. At first glance, it appears to be a hybrid of unicast and broadcast communication, but that isn't quite the case. Multicast does allow point-to-multipoint communication, which is similar to broadcasts, but it happens in a different manner. The crux of *multicast* is that it enables multiple recipients to receive messages without flooding the messages to all hosts on a broadcast domain. However, this is not the default behavior—it's what we *can* do with multicasting if it's configured correctly!

Multicast works by sending messages or data to IP *multicast group* addresses. Unlike with broadcasts, which aren't forwarded, routers then forward copies of the packet out to every interface that has hosts *subscribed* to that group address. This is where multicast differs from broadcast messages—with multicast communication, copies of packets, in theory, are sent only to subscribed hosts. For example, when I say in theory, I mean that the hosts will receive a multicast packet destined for 224.0.0.10. This is an EIGRP packet, and only a router running the EIGRP protocol will read these. All hosts on the broadcast LAN, and Ethernet is a broadcast multi-access LAN technology, will pick up the frame, read the destination address, then immediately discard the frame unless they're in the multicast group. This saves PC processing, not LAN bandwidth. Be warned though—multicasting can cause some serious LAN congestion if it's not implemented carefully! Figure 1.35 shows a Cisco router sending an EIGRP multicast packet on the local LAN and only the other Cisco router will accept and read this packet.

FIGURE 1.35 EIGRP multicast example



There are several different groups that users or applications can subscribe to. The range of multicast addresses starts with 224.0.0.0 and goes through 239.255.255.255. As you can see, this range of addresses falls within IP Class D address space based on classful IP assignment.

Exam Essentials

Understand the difference between a broadcast, unicast, and multicast address. A broadcast is to all devices in a subnet, a unicast is to one device, and a multicast is to some but not all devices.

Describe the need for private IPv4 addressing

The people who created the IP addressing scheme also created private IP addresses. These addresses can be used on a private network, but they're not routable through the Internet. This is designed for the purpose of creating a measure of well-needed security, but it also conveniently saves valuable IP address space.

If every host on every network was required to have real routable IP addresses, we would have run out of IP addresses to hand out years ago. But by using private IP addresses, ISPs, corporations, and home users only need a relatively tiny group of bona fide IP addresses to connect their networks to the Internet. This is economical because they can use private IP addresses on their inside networks and get along just fine.

To accomplish this task, the ISP and the corporation—the end user, no matter who they are—need to use something called *Network Address Translation (NAT)*, which basically takes a private IP address and converts it for use on the Internet. Many people can use the same real IP address to transmit out onto the Internet. Doing things this way saves megatons of address space—good for us all!

The reserved private addresses are listed in Table 1.3.

TABLE 1.3 Reserved IP address space

Address Class	Reserved Address Space
Class A	10.0.0.0 through 10.255.255.255
Class B	172.16.0.0 through 172.31.255.255
Class C	192.168.0.0 through 192.168.255.255

Exam Essentials

Identify the private IP ranges. The Class A private address range is 10.0.0.0 through 10.255.255.255. The Class B private address range is 172.16.0.0 through 172.31.255.255. The Class C private address range is 192.168.0.0 through 192.168.255.255.

Identify the appropriate IPv6 addressing scheme to satisfy addressing requirements in a LAN/WAN environment

There are a number of different approaches to applying an IPv6 address in the network. In this section we'll identify those methods and their characteristics.

Manual Address Assignment

In order to enable IPv6 on a router, you have to use the `ipv6 unicast-routing` global configuration command:

```
Corp(config)#ipv6 unicast-routing
```

By default, IPv6 traffic forwarding is disabled, so using this command enables it. Also, as you've probably guessed, IPv6 isn't enabled by default on any interfaces either, so we have to go to each interface individually and enable it.

There are a few different ways to do this, but a really easy way is to just add an address to the interface. You use the interface configuration command `ipv6 address <ipv6prefix>/<prefix-length> [eui-64]`to get this done.

Here's an example:

```
Corp(config-if)#ipv6 address 2001:db8:3c4d:1:0260:d6FF.FE73:1987/64
```

You can specify the entire 128-bit global IPv6 address as I just demonstrated with the preceding command, or you can use the EUI-64 option. Remember, the EUI-64 (extended unique identifier) format allows the device to use its MAC address and pad it to make the interface ID. Check it out:

```
Corp(config-if)#ipv6 address 2001:db8:3c4d:1::/64 eui-64
```

As an alternative to typing in an IPv6 address on a router, you can enable the interface instead to permit the application of an automatic link-local address.

To configure a router so that it uses only link-local addresses, use the `ipv6 enable` interface configuration command:

```
Corp(config-if)#ipv6 enable
```

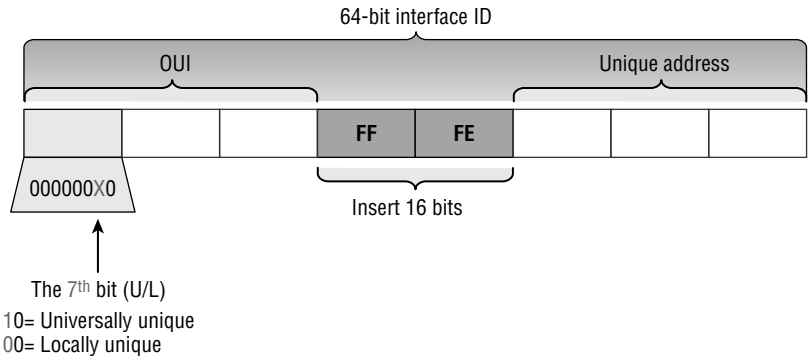
Remember, if you have only a link-local address, you will be able to communicate only on that local subnet.

Stateless Autoconfiguration (EUI-64)

Autoconfiguration is an especially useful solution because it allows devices on a network to address themselves with a link-local unicast address as well as with a global unicast address. This process happens through first learning the prefix information from the router and then appending the device's own interface address as the interface ID. But where does it get that interface ID? Well, you know every device on an Ethernet network has a physical MAC address, which is exactly what's used for the interface ID. But since the interface ID in an IPv6 address is 64 bits in length and a MAC address is only 48 bits, where do the extra 16 bits come from? The MAC address is padded in the middle with the extra bits—it's padded with FFFE.

For example, let's say I have a device with a MAC address that looks like this: 0060:d673:1987. After it's been padded, it would look like this: 0260:d6FF:FE73:1987. Figure 1.36 illustrates what an EUI-64 address looks like.

FIGURE 1.36 EUI-64 interface ID assignment



So where did that 2 in the beginning of the address come from? Another good question. You see that part of the process of padding, called modified EUI-64 format, changes a bit to specify if the address is locally unique or globally unique. And the bit that gets changed is the 7th bit in the address.

The reason for modifying the U/L bit is that, when using manually assigned addresses on an interface, it means you can simply assign the address 2001:db8:1:9::1/64 instead of the much longer 2001:db8:1:9:0200::1/64. Also, if you are going to manually assign a link-local address, you can assign the short address fe80::1 instead of the long fe80::0200:0:0:1 or fe80:0:0:0:0200::1. So, even though at first glance it seems the IETF made this harder for you to simply understand IPv6 addressing by flipping the 7th bit, in reality this made addressing much simpler. Also, since most people don't typically override the burned-in address, the U/L bit is a 0, which means that you'll see this inverted to a 1 most of the time. But because you're studying the Cisco exam objectives, you'll need to look at inverting it both ways.

Here are a few examples:

- MAC address 0090:2716:fd0f
- IPv6 EUI-64 address: 2001:0db8:0:1:0290:27ff:fe16:fd0f

That one was easy! Too easy for the Cisco exam, so let's do another:

- MAC address aa12:bcbc:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:a812:bcff:febc:1234

10101010 represents the first 8 bits of the MAC address (aa), which when inverting the 7th bit becomes 10101000. The answer becomes A8. I can't tell you how important this is for you to understand, so bear with me and work through a couple more!

- MAC address 0c0c:dede:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:0e0c:deff:fede:1234

0c is 00001100 in the first 8 bits of the MAC address, which then becomes 00001110 when flipping the 7th bit. The answer is then 0e. Let's practice one more:

- MAC address 0b34:ba12:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:0934:baff:fe12:1234

0b in binary is 00001011, the first 8 bits of the MAC address, which then becomes 00001001. The answer is 09.

Pay extra-special attention to this EUI-64 address assignment and be able to convert the 7th bit based on the EUI-64 rules!

DHCPv6 (Stateful)

DHCPv6 works pretty much the same way DHCP does in v4, with the obvious difference that it supports IPv6's new addressing scheme. And it might come as a surprise, but there are a couple of other options that DHCP still provides for us that autoconfiguration doesn't. And no, I'm not kidding— in autoconfiguration, there's absolutely no mention of DNS servers, domain names, or many of the other options that DHCP has always generously provided for us via IPv4. This is a big reason that the odds favor DHCP's continued use into the future in IPv6 at least partially—maybe even most of the time!

Upon booting up in IPv4, a client sends out a DHCP Discover message looking for a server to give it the information it needs. But remember, in IPv6, the RS and RA process happens first, so if there's a DHCPv6 server on the network, the RA that comes back to the client will tell it if DHCP is available for use. If a router isn't found, the client will respond by sending out a DHCP Solicit message, which is actually a multicast message addressed with a destination of ff02::1:2 that calls out, "All DHCP agents, both servers and relays."

It's good to know that there's some support for DHCPv6 in the Cisco IOS even though it's limited. This rather miserly support is reserved for stateless DHCP servers and tells us it

doesn't offer any address management of the pool or the options available for configuring that address pool other than the DNS, domain name, default gateway, and SIP servers.

This means that you're definitely going to need another server around to supply and dispense all the additional, required information—maybe to even manage the address assignment, if needed!

Exam Essentials

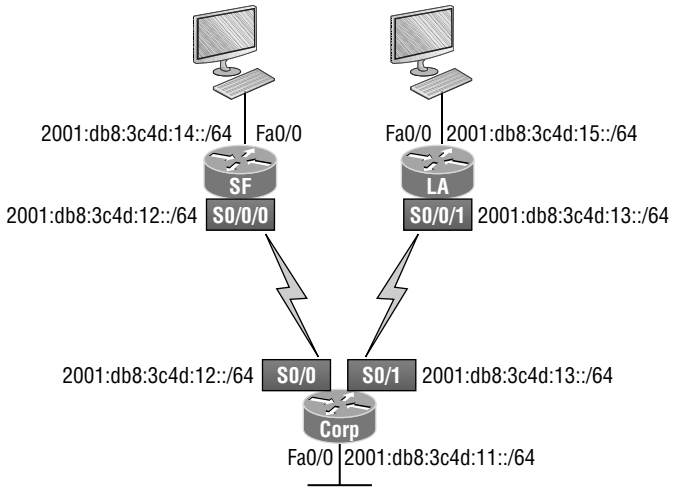
Describe the available methods to apply IPv6 addresses to the network. These include manual configuration, stateless autoconfiguration, and DHCPv6.

Understand and be able to read a EUI-64 address with the 7th bit inverted. Hosts can use autoconfiguration to obtain an IPv6 address, and one of the ways it can do that is through what is called EUI-64. This takes the unique MAC address of a host and inserts FF:FE in the middle of the address to change a 48-bit MAC address to a 64-bit interface ID. In addition to inserting the 16 bits into the interface ID, the 7th bit of the 1st byte is inverted, typically from a 0 to a 1.

Configure, verify, and troubleshoot IPv6 addressing

While covering the steps to confuse and troubleshoot IPv6, we'll be using the internetwork shown in Figure 1.37. This section will cover the process of both.

FIGURE 1.37 Our internetwork



Configuring IPv6 on Our Internetwork

Well start by using a simple subnet scheme of 11, 12, 13, 14, and 15. After that, we'll add the OSPFv3 routing protocol. Notice in Figure 1.37 how the subnet numbers are the same on each end of the WAN links. Keep in mind that we'll finish this chapter by running through some verification commands.

As usual, I'll start with the Corp router:

```
Corp#config t
Corp(config)#ipv6 unicast-routing
Corp(config)#int f0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:11::/64 eui-64
Corp(config-if)#int s0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
Corp(config-if)#int s0/1
Corp(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
Corp(config-if)#^Z
Corp#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
```

Pretty simple! In the previous configuration, I only changed the subnet address for each interface slightly. Let's take a look at the routing table now:

```
Corp(config-if)#do sho ipv6 route
C   2001:DB8:3C4D:11::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:DB8:3C4D:11:20D:BDFF:FE3B:D80/128 [0/0]
     via ::, FastEthernet0/0
C   2001:DB8:3C4D:12::/64 [0/0]
     via ::, Serial0/0
L   2001:DB8:3C4D:12:20D:BDFF:FE3B:D80/128 [0/0]
     via ::, Serial0/0
C   2001:DB8:3C4D:13::/64 [0/0]
     via ::, Serial0/1
L   2001:DB8:3C4D:13:20D:BDFF:FE3B:D80/128 [0/0]
     via ::, Serial0/1
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
Corp(config-if)#
```

Alright, but what's up with those two addresses for each interface? One shows C for connected, one shows L. The connected address indicates the IPv6 address I configured on each interface, and the L is the link-local that's been automatically assigned. Notice in the link-local address that the FF:FE is inserted into the address to create the EUI-64 address.

Let's configure the SF router now:

```
SF#config t
SF(config)#ipv6 unicast-routing
SF(config)#int s0/0/0
SF(config-if)#ipv6 address 2001:db8:3c4d:12::/64
% 2001:DB8:3C4D:12::/64 should not be configured on Serial0/0/0, a subnet router
anycast
SF(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
SF(config-if)#int fa0/0
SF(config-if)#ipv6 address 2001:db8:3c4d:14::/64 eui-64
SF(config-if)#^Z
SF#show ipv6 route
C   2001:DB8:3C4D:12::/64 [0/0]
     via ::, Serial0/0/0
L   2001:DB8:3C4D:12::/128 [0/0]
     via ::, Serial0/0/0
L   2001:DB8:3C4D:12:21A:2FFF:FEE7:4398/128 [0/0]
     via ::, Serial0/0/0
C   2001:DB8:3C4D:14::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:DB8:3C4D:14:21A:2FFF:FEE7:4398/128 [0/0]
     via ::, FastEthernet0/0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
```

Did you notice that I used the exact IPv6 subnet addresses on each side of the serial link? Good… but wait—what's with that anycast error I received when trying to configure the interfaces on the SF router? I didn't mean to create that error; it happened because I forgot to add the `eui-64` at the end of the address. Still, what's behind that error? An anycast address is a host address of all 0s, meaning the last 64 bits are all off, but by typing in `/64` without the `eui-64`, I was telling the interface that the unique identifier would be nothing but zeros, and that's not allowed!

Let's configure the LA router now:

```
SF#config t
SF(config)#ipv6 unicast-routing
```

```
SF(config)#int s0/0/1
SF(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
SF(config-if)#int f0/0
SF(config-if)#ipv6 address 2001:db8:3c4d:15::/64 eui-64
SF(config-if)#do show ipv6 route
C   2001:DB8:3C4D:13::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:13:21A:6CFF:FEA1:1F48/128 [0/0]
     via ::, Serial0/0/1
C   2001:DB8:3C4D:15::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:DB8:3C4D:15:21A:6CFF:FEA1:1F48/128 [0/0]
     via ::, FastEthernet0/0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
```

This looks good, but I want you to notice that I used the exact same IPv6 subnet addresses on each side of the links from the Corp router to the SF router as well as from the Corp to the LA router.

Exam Essentials

Describe the steps required to apply IPv6 addresses to a network. These steps include selecting subnet numbers and configuring IPv6 addresses on both the router's interfaces and the host.

Configure and verify IPv6 Stateless Address Auto Configuration

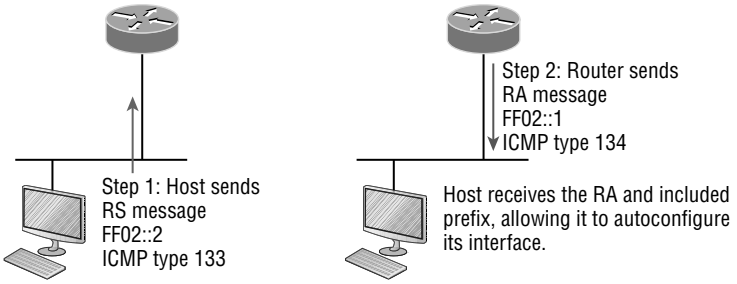
To perform autoconfiguration, a host goes through a basic two-step process:

1. First, the host needs the prefix information, similar to the network portion of an IPv4 address, to configure its interface, so it sends a router solicitation (RS) request for it. This RS is then sent out as a multicast to all routers (FF02::2). The actual information being sent is a type of ICMP message, and like everything in networking, this ICMP message has a number that identifies it. The RS message is ICMP type 133.
2. The router answers back with the required prefix information via a router advertisement (RA). An RA message also happens to be a multicast packet that's sent to the

all-nodes multicast address (FF02::1) and is ICMP type 134. RA messages are sent on a periodic basis, but the host sends the RS for an immediate response so it doesn't have to wait until the next scheduled RA to get what it needs.

These two steps are shown in Figure 1.38.

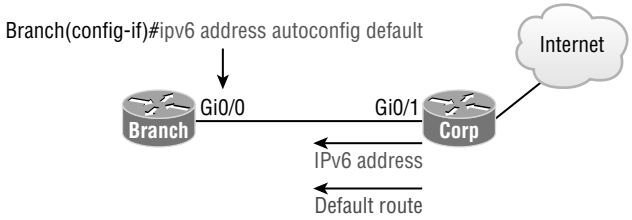
FIGURE 1.38 Two steps to IPv6 autoconfiguration



By the way, this type of autoconfiguration is also known as stateless autoconfiguration because it doesn't contact or connect to and receive any further information from the other device. We'll get to stateful configuration when we talk about DHCPv6 next.

But before we do that, first take a look at Figure 1.39. In this figure, the Branch router needs to be configured, but I just don't feel like typing in an IPv6 address on the interface connecting to the Corp router. I also don't feel like typing in any routing commands, but I need more than a link-local address on that interface, so I'm going to have to do something! So basically, I want to have the Branch router work with IPv6 on the internetwork with the least amount of effort from me. Let's see if I can get away with that.

FIGURE 1.39 IPv6 autoconfiguration example



Aha—there is an easy way! I love IPv6 because it allows me to be relatively lazy when dealing with some parts of my network, yet it still works really well. By using the command `ipv6 address autoconfig`, the interface will listen for RAs and then, via the EUI-64 format, it will assign itself a global address—sweet!

Exam Essentials

Describe the autoconfiguration process. List the steps a device uses to create an IPv6 address. Understand the roles played by both the router and the device.

Configure stateless autoconfiguration. Identify the commands necessary to configure the devices and the router for stateless autoconfiguration.

Compare and contrast IPv6 address types

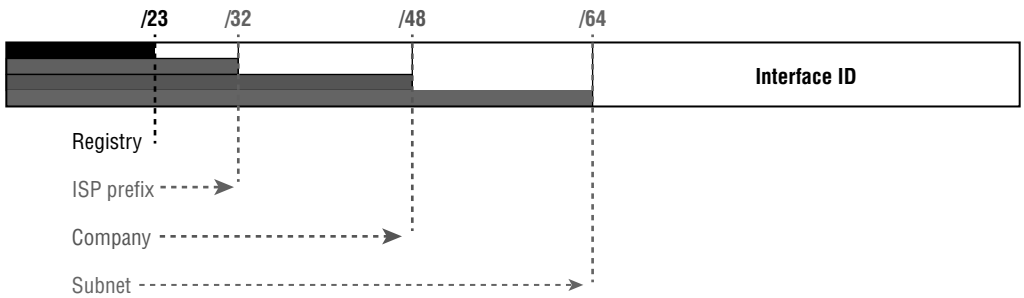
We're all familiar with IPv4's unicast, broadcast, and multicast addresses that basically define who or at least how many other devices we're talking to. But as I mentioned, IPv6 modifies that trio and introduces the anycast. Broadcasts, as we know them, have been eliminated in IPv6 because of their cumbersome inefficiency and basic tendency to drive us insane!

So let's find out what each of these types of IPv6 addressing and communication methods do for us:

Unicast Packets addressed to a unicast address are delivered to a single interface. For load balancing, multiple interfaces across several devices can use the same address, but we'll call that an anycast address. There are a few different types of unicast addresses, but we don't need to get further into that here.

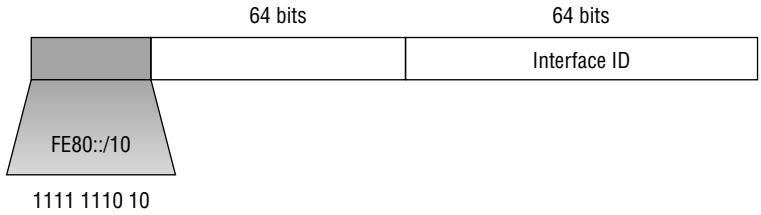
Global unicast addresses (2000::/3) These are your typical publicly routable addresses and they're the same as in IPv4. Global addresses start at 2000::/3. Figure 1.40 shows how a unicast address breaks down. The ISP can provide you with a minimum /48 network ID, which in turn provides you with 16-bits to create a unique 64-bit router interface address. The last 64-bits are the unique host ID.

FIGURE 1.40 IPv6 global unicast addresses



Link-local addresses (FE80::/10) These are like the Automatic Private IP Address (APIPA) addresses that Microsoft uses to automatically provide addresses in IPv4 in that they're not meant to be routed. In IPv6 they start with FE80::/10, as shown in Figure 1.41. Think of these addresses as handy tools that give you the ability to throw a temporary LAN together for meetings or create a small LAN that's not going to be routed but still needs to share and access files and services locally.

FIGURE 1.41 IPv6 link local FE80::/10: The first 10 bits define the address type.



Unique local addresses (FC00::/7) These addresses are also intended for nonrouting purposes over the Internet, but they are nearly globally unique, so it's unlikely you'll ever have one of them overlap. Unique local addresses were designed to replace site-local addresses, so they basically do almost exactly what IPv4 private addresses do: Allow communication throughout a site while being routable to multiple local networks. Site-local addresses were deprecated as of September 2004.

Multicast (FF00::/8) Again, as in IPv4, packets addressed to a multicast address are delivered to all interfaces tuned into the multicast address. Sometimes people call them "one-to-many" addresses. It's really easy to spot a multicast address in IPv6 because they always start with *FF*.

Anycast Like multicast addresses, an anycast address identifies multiple interfaces on multiple devices. But there's a big difference: the anycast packet is delivered to only one device—actually, to the closest one it finds defined in terms of routing distance. And again, this address is special because you can apply a single address to more than one host. These are referred to as "one-to-nearest" addresses. Anycast addresses are typically only configured on routers, never hosts, and a source address could never be an anycast address. Of note is that the IETF did reserve the top 128 addresses for each /64 for use with anycast addresses.

You're probably wondering if there are any special, reserved addresses in IPv6 because you know they're there in IPv4. Well there are—plenty of them! Let's go over those now.

Special Addresses

I'm going to list some of the addresses and address ranges (in Table 1.4) that you should definitely make sure to remember because you'll eventually use them. They're all special or

reserved for a specific use, but unlike IPv4, IPv6 gives us a galaxy of addresses, so reserving a few here and there doesn't hurt at all!

TABLE 1.4 Special IPv6 addresses

Address	Meaning
0:0:0:0:0:0:0:0	Equals ::. This is the equivalent of IPv4's 0.0.0.0 and is typically the source address of a host before the host receives an IP address when you're using DHCP-driven stateful configuration.
0:0:0:0:0:0:0:1	Equals ::1. The equivalent of 127.0.0.1 in IPv4
0:0:0:0:0:0:192.168.100.1	This is how an IPv4 address would be written in a mixed IPv6/IPv4 network environment.
2000::/3	The global unicast address range
FC00::/7	The unique local unicast range
FE80::/10	The link-local unicast range
FF00::/8	The multicast range
3FFF:FFFF::/32	Reserved for examples and documentation
2001:0DB8::/32	Also reserved for examples and documentation
2002::/16	Used with 6-to-4 tunneling, which is an IPv4-to-IPv6 transition system. The structure allows IPv6 packets to be transmitted over an IPv4 network without the need to configure explicit tunnels.

When you run IPv4 and IPv6 on a router, you have what is called "dual-stack."

Exam Essentials

Understand link-local. Link-local is like an IPv4 private IP address, but it can't be routed at all, not even in your organization.

Understand unique local. This, like link-local, is like a private IP address in IPv4 and cannot be routed to the Internet. However, the difference between link-local and unique local is that unique local can be routed within your organization or company.

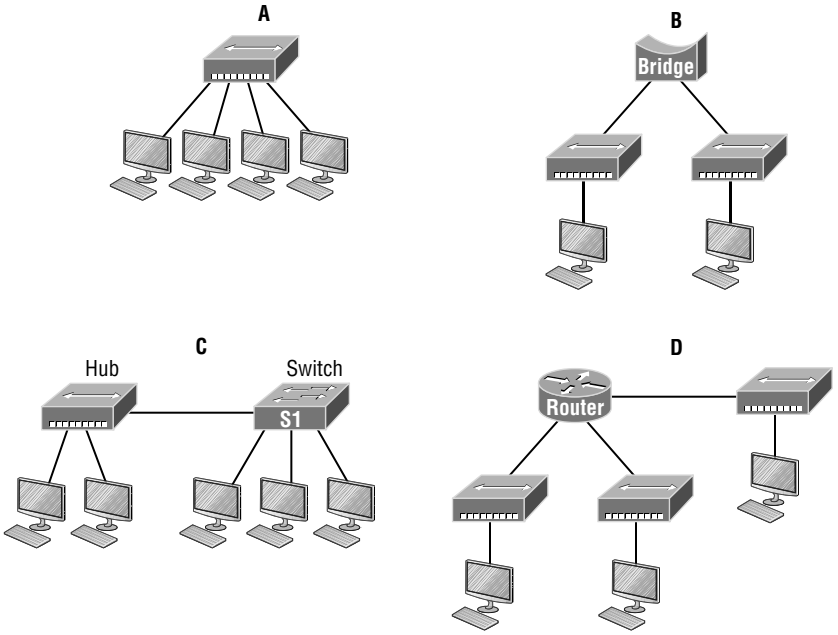
Review Questions

You can find the answers in the Appendix.

1. What cable type is shown in the following image?

A. Fiber optic
B. Rollover
C. Coaxial
D. Full-duplex

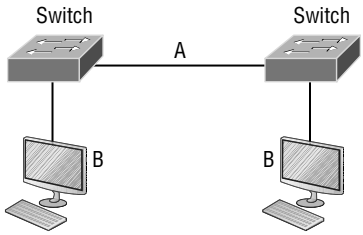
2. Which of the following statements is/are true with regard to the device shown below?



A. It includes one collision domain and one broadcast domain.
B. It includes no collision domain and one broadcast domains.
C. It includes 10 collision domains and one broadcast domain.
D. It includes one collision domain and 10 broadcast domains.
E. It includes 10 collision domains and 10 broadcast domains.

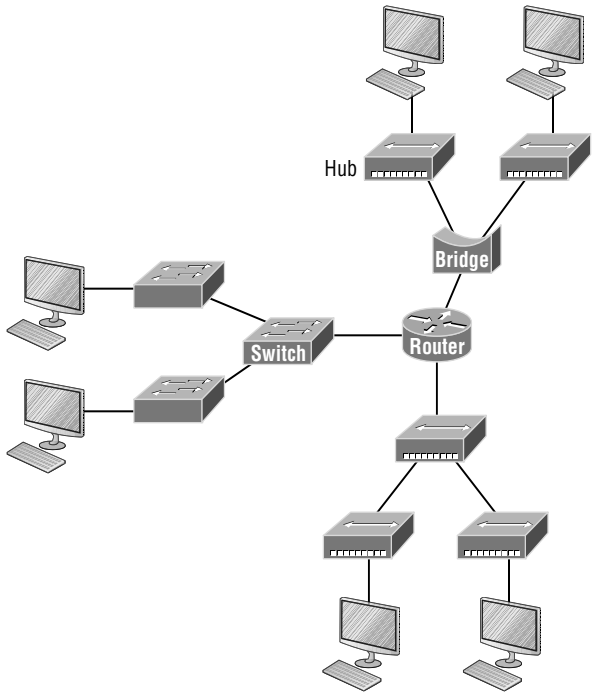
3. Which of the following Application layer protocols sets up a secure session that's similar to Telnet?
   - **A.** FTP
   - **B.** SSH
   - **C.** DNS
   - **D.** DHCP

4. In the following diagram, identify the cable types required for connections A and B.



   - **A.** A crossover, B crossover
   - **B.** A crossover, B straight through
   - **C.** A straight through, B straight through
   - **D.** A straight through, B crossover

5. How many collision domains are present in the following diagram?



A. 8
B. 9
C. 10
D. 11

6. What type of cable is used between a host and switch?
A. crossover
B. rollover
C. straight through
D. console

7. How many octets are parts of the network portion of an IP address when the address is a Class B?
A. one
B. two
C. three
D. four

8. Which of the following layers of the OSI model was later subdivided into two layers?
A. Presentation
B. Transport
C. Data Link
D. Physical

9. What is a function of an access point (AP)?
A. To monitor and control the incoming and outgoing network traffic
B. To automatically handle the configuration of a wireless access point
C. To allow wireless devices to connect to a wired network
D. To connect networks and intelligently choose the best paths between networks

10. A _____ is an example of a device that operates only at the physical layer.
A. Hub
B. Switch
C. Router
D. Bridge