

Chapter 1



Internetworking

THE CCNA EXAM TOPICS COVERED IN THIS CHAPTER INCLUDE THE FOLLOWING:

✓ **TECHNOLOGY**

- Describe network communications using layered models
- Compare and contrast key characteristics of LAN environments
- Describe the components of network devices
- Evaluate rules for packet control



Welcome to the exciting world of internetworking. This first chapter will really help you understand the basics of internetworking by focusing on how to connect networks together using Cisco routers and switches. First, you need to know exactly what an internetwork is, right? You create an internetwork when you take two or more LANs or WANs and connect them via a router, and configure a logical network addressing scheme with a protocol such as IP.

I'll be covering these four topics in this chapter:

- Internetworking basics
- Network segmentation
- How bridges, switches, and routers are used to physically segment a network
- How routers are employed to create an internetwork

I'm also going to dissect the Open Systems Interconnection (OSI) model and describe each part to you in detail, because you really need a good grasp of it for the solid foundation you'll build your networking knowledge upon. The OSI model has seven hierarchical layers that were developed to enable different networks to communicate reliably between disparate systems. Since this book is centering upon all things CCNA, it's crucial for you to understand the OSI model as Cisco sees it, so that's how I'll be presenting the seven layers of the OSI model to you.

Since there's a bunch of different types of devices specified at the different layers of the OSI model, it's also very important to understand the many types of cables and connectors used for connecting all those devices to a network. We'll go over cabling Cisco devices, discussing how to connect to a router or switch along with Ethernet LAN technologies, and even how to connect a router or switch with a console connection.

We'll finish the chapter by discussing the Cisco three-layer hierarchical model that was developed by Cisco to help you design, implement, and troubleshoot internetworks.

After you finish reading this chapter, you'll encounter 20 review questions and three written labs. These are given to you to really lock the information from this chapter into your memory. So don't skip them!

Internetworking Basics

Before we explore internetworking models and the specifications of the OSI reference model, you've got to understand the big picture and learn the answer to the key question, "Why is it so important to learn Cisco internetworking?"

Networks and networking have grown exponentially over the last 15 years—understandably so. They've had to evolve at light speed just to keep up with huge increases in basic mission-critical user needs such as sharing data and printers, as well as more advanced demands such as video conferencing. Unless everyone who needs to share network resources is located in the same office area (an increasingly uncommon situation), the challenge is to connect the sometimes many relevant networks together so all users can share the networks' wealth.

It's also likely that at some point, you'll have to break up one large network into a number of smaller ones because user response has dwindled to a trickle as the network grew and grew and LAN traffic congestion reached overwhelming proportions. Breaking up a larger network into a number of smaller ones is called *network segmentation*, and it's accomplished using *routers*, *switches*, and *bridges*.

Possible causes of LAN traffic congestion are

- Too many hosts in a broadcast domain
- Broadcast storms
- Multicasting
- Low bandwidth
- Adding hubs for connectivity to the network
- A large amount of ARP or IPX traffic (IPX is a Novell routing protocol that is like IP, but really, really chatty)

Routers are used to connect networks together and route packets of data from one network to another. Cisco became the de facto standard of routers because of their high-quality router products, great selection, and fantastic service. Routers, by default, break up a *broadcast domain*, which is the set of all devices on a network segment that hear all broadcasts sent on that segment. Breaking up a broadcast domain is important because when a host or server sends a network broadcast, every device on the network must read and process that broadcast—unless you've got a router. When the router's interface receives this broadcast, it can respond by basically saying "Thanks, but no thanks," and discard the broadcast without forwarding it on to other networks. Even though routers are known for breaking up broadcast domains by default, it's important to remember that they break up collision domains as well.

Two advantages of using routers in your network are

- They don't forward broadcasts by default.
- They can filter the network based on layer 3 (Network layer) information (i.e., IP address).

Four router functions in your network can be listed as

- Packet switching
- Packet filtering
- Internetwork communication
- Path selection

Remember that routers are really switches, but they're actually what we call layer 3 switches (we'll talk about layers later in this chapter). Unlike layer 2 switches that forward or filter frames,

4 Chapter 1 • Internetworking

routers (layer 3 switches) use logical addressing and provide what is called packet switching. Routers can also provide packet filtering by using access-lists (discussed in Chapter 10), and when routers connect two or more networks together and use logical addressing (IP), this is called an internetwork. Lastly, routers use a routing table (map of the internetwork) to make path selections and to forward packets to remote networks.

Conversely, switches aren't used to create internetworks, they're employed to add functionality to an internetwork LAN. The main purpose of a switch is to make a LAN work better—to optimize its performance—providing more bandwidth for the LAN's users. And switches don't forward packets to other networks as routers do. Instead, they only “switch” frames from one port to another within the switched network. Okay, you may be thinking, “Wait a minute, what are frames and packets?” I'll tell you all about them later in this chapter, I promise!

By default, switches break up *collision domains*. This is an Ethernet term used to describe a network scenario wherein one particular device sends a packet on a network segment, forcing every other device on that same segment to pay attention to it. At the same time, a different device tries to transmit, leading to a collision, after which both devices must retransmit, one at a time. Not very efficient! This situation is typically found in a hub environment where each host segment connects to a hub that represents only one collision domain and only one broadcast domain. By contrast, each and every port on a switch represents its own collision domain.



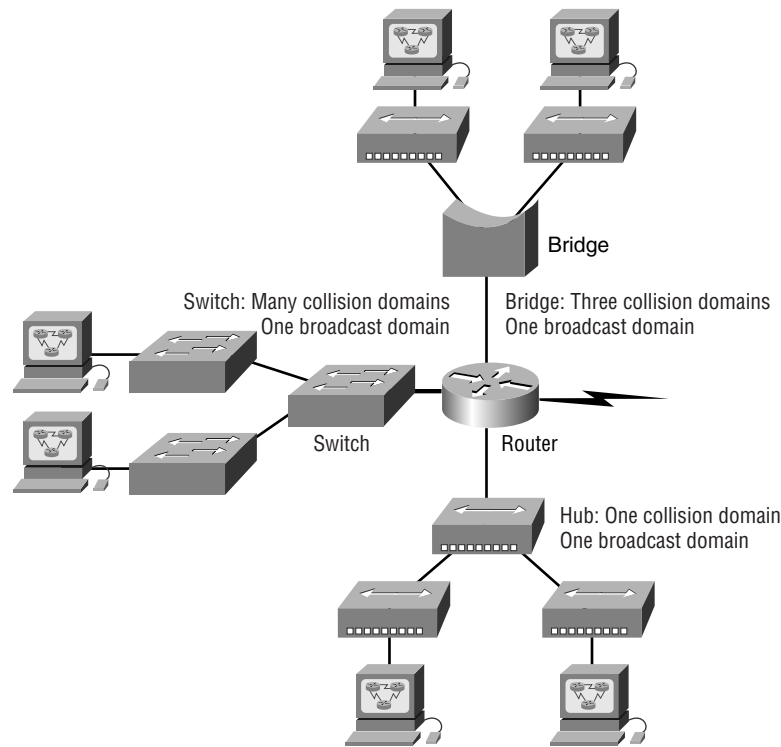
Switches create separate collision domains, but a single broadcast domain. Routers provide a separate broadcast domain for each interface.

The term *bridging* was introduced before routers and hubs were implemented, so it's pretty common to hear people referring to bridges as “switches.” That's because bridges and switches basically do the same thing—break up collision domains on a LAN. So what this means is that a switch is basically just a multiple-port bridge with more brainpower, right? Well, pretty much, but there are differences. Switches do provide this function, but they do so with greatly enhanced management ability and features. Plus, most of the time, bridges only had two or four ports. Yes, you could get your hands on a bridge with up to 16 ports, but that's nothing compared to the hundreds available on some switches!



You would use a bridge in a network to reduce collisions within broadcast domains and to increase the number of collision domains in your network. Doing this provides more bandwidth for users. And keep in mind that using hubs in your network can contribute to congestion on your Ethernet network. As always, plan your network design carefully!

Figure 1.1 shows how a network would look with all these internetwork devices in place. Remember that the router will not only break up broadcast domains for every LAN interface, but break up collision domains as well.

FIGURE 1.1 Internetworking devices

When you looked at Figure 1.1, did you notice that the router is found at center stage, and that it connects each physical network together? We have to use this layout because of the older technologies involved—bridges and hubs. Once we have only switches in our network, things change a lot! The LAN switches would then be placed at the center of the network world and the routers would be found connecting only logical networks together. If I've implemented this kind of setup, I've created virtual LANs (VLANs). Again, don't stress—I'll go over VLANs thoroughly with you in Chapter 8, "Virtual LANs (VLANs)."

On the top network in Figure 1.1, you'll notice that a bridge was used to connect the hubs to a router. The bridge breaks up collision domains, but all the hosts connected to both hubs are still crammed into the same broadcast domain. Also, the bridge only created two collision domains, so each device connected to a hub is in the same collision domain as every other device connected to that same hub. This is pretty lame, but it's still better than having one collision domain for all hosts.

Notice something else: the three hubs at the bottom that are connected also connect to the router, creating one humongous collision domain and one humongous broadcast domain. This makes the bridged network look much better indeed!



Although bridges are used to segment networks, they will not isolate broadcast or multicast packets.

The best network connected to the router is the LAN switch network on the left. Why? Because each port on that switch breaks up collision domains. But it's not all good—all devices are still in the same broadcast domain. Do you remember why this can be a really bad thing? Because all devices must listen to all broadcasts transmitted, that's why. And if your broadcast domains are too large, the users have less bandwidth and are required to process more broadcasts, and network response time will slow to a level that could cause office riots.

Obviously, the best network is one that's correctly configured to meet the business requirements of the company it serves. LAN switches with routers, correctly placed in the network, are the best network design. This book will help you understand the basics of routers and switches so you can make tight, informed decisions on a case-by-case basis.

Let's go back to Figure 1.1 again. Looking at the figure, how many collision domains and broadcast domains are in this network? Hopefully, you answered nine collision domains and three broadcast domains! The broadcast domains are definitely the easiest to see because only routers break up broadcast domains by default. And since there are three connections, that gives you three broadcast domains. But do you see the nine collision domains? Just in case that's a No, I'll explain. The all-hub network is one collision domain, the bridge network equals three collision domains. Add in the switch network of five collision domains—one for each switch port—and you've got a total of nine.

So now that you've gotten an introduction to internetworking, and the various devices that live in an internetwork, it's time to head into internetworking models.



Real World Scenario

Should I just replace all my hubs with switches?

You're a Network Administrator at a large company in San Jose. The boss comes to you and says that he got your requisition to buy a switch and is not sure about approving the expense; do you really need it?

Well, if you can, sure—why not? Switches really add a lot of functionality to a network that hubs just don't have. But most of us don't have an unlimited budget. Hubs still can create a nice network—that is, of course, if you design and implement the network correctly.

Let's say that you have 40 users plugged into four hubs, 10 users each. At this point, the hubs are all connected together so that you have one large collision domain and one large broadcast domain. If you can afford to buy just one switch and plug each hub into a switch port, as well as the servers into the switch, then you now have four collision domains and one broadcast domain. Not great, but for the price of one switch, your network is a much better thing.

So, go ahead! Put that requisition in to buy all new switches. What do you have to lose?

Internetworking Models

When networks first came into being, computers could typically communicate only with computers from the same manufacturer. For example, companies ran either a complete DECnet solution or an IBM solution—not both together. In the late 1970s, the *Open Systems Interconnection (OSI) reference model* was created by the International Organization for Standardization (ISO) to break this barrier.

The OSI model was meant to help vendors create interoperable network devices and software in the form of protocols so that different vendor networks could work with each other. Like world peace, it'll probably never happen completely, but it's still a great goal.

The OSI model is the primary architectural model for networks. It describes how data and network information are communicated from an application on one computer, through the network media, to an application on another computer. The OSI reference model breaks this approach into layers.

In the following section, I am going to explain the layered approach and how we can use this approach in helping us troubleshoot our internetworks.

The Layered Approach

A *reference model* is a conceptual blueprint of how communications should take place. It addresses all the processes required for effective communication and divides these processes into logical groupings called *layers*. When a communication system is designed in this manner, it's known as *layered architecture*.

Think of it like this: You and some friends want to start a company. One of the first things you'll do is sit down and think through what tasks must be done, who will do them, what order they will be done in, and how they relate to each other. Ultimately, you might group these tasks into departments. Let's say you decide to have an order-taking department, an inventory department, and a shipping department. Each of your departments has its own unique tasks, keeping its staff members busy and requiring them to focus on only their own duties.

In this scenario, I'm using departments as a metaphor for the layers in a communication system. For things to run smoothly, the staff of each department will have to trust and rely heavily upon the others to do their jobs and competently handle their unique responsibilities. In your planning sessions, you would probably take notes, recording the entire process to facilitate later discussions about standards of operation that will serve as your business blueprint, or reference model.

Once your business is launched, your department heads, armed with the part of the blueprint relating to their department, will need to develop practical methods to implement their assigned tasks. These practical methods, or protocols, will need to be compiled into a standard operating procedures manual and followed closely. Each of the various procedures in your manual will have been included for different reasons and have varying degrees of importance and implementation. If you form a partnership or acquire another company, it will be imperative that its business protocols—its business blueprint—match yours (or at least be compatible with it).

Similarly, software developers can use a reference model to understand computer communication processes and see what types of functions need to be accomplished on any one layer. If they are developing a protocol for a certain layer, all they need to concern themselves with is the

8 Chapter 1 • Internetworking

specific layer's functions, not those of any other layer. Another layer and protocol will handle the other functions. The technical term for this idea is *binding*. The communication processes that are related to each other are bound, or grouped together, at a particular layer.

Advantages of Reference Models

The OSI model is hierarchical, and the same benefits and advantages can apply to any layered model. The primary purpose of all such models, especially the OSI model, is to allow different vendors' networks to interoperate.

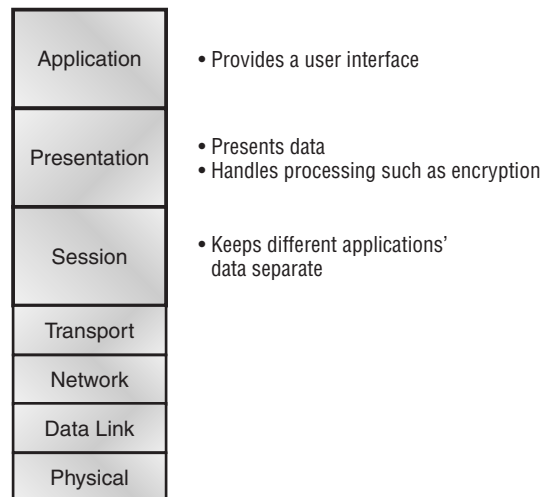
Advantages of using the OSI layered model include, but are not limited to, the following:

- It divides the network communication process into smaller and simpler components, thus aiding component development, design, and troubleshooting.
- It allows multiple-vendor development through standardization of network components.
- It encourages industry standardization by 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, so it does not hamper development.

The OSI Reference Model

One of the greatest functions of the OSI specifications is to assist in data transfer between disparate hosts—meaning, for example, that they enable us to transfer data between a Unix host and a PC or a Mac.

FIGURE 1.2 The upper layers



The OSI isn't a physical model, though. Rather, it's a set of guidelines that application developers can use to create and implement applications that run on a network. It also provides a framework for creating and implementing networking standards, devices, and inter-networking schemes.

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 and with users. The bottom four layers define how data is transmitted end-to-end. Figure 1.2 shows the three upper layers and their functions, and Figure 1.3 shows the four lower layers and their functions.

When you study Figure 1.2, understand that the user interfaces with the computer at the Application layer, and also that the upper layers are responsible for applications communicating between hosts. Remember that none of the upper layers knows anything about networking or network addresses. That's the responsibility of the four bottom layers.

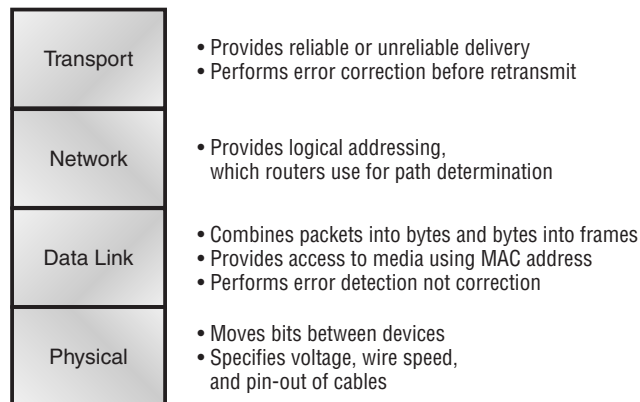
In Figure 1.3, you can see that it's the four bottom layers that define how data is transferred through a physical wire or through switches and routers. These bottom layers also determine how to rebuild a data stream from a transmitting host to a destination host's application.

Network devices that operate at all seven layers of the OSI model include

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

Basically, the ISO is pretty much the Emily Post of the network protocol world. Just like Ms. Post, who wrote the book setting the standards—or protocols—for human social interaction, the ISO developed the OSI reference model as the precedent and guide for an open network protocol set. Defining the etiquette of communication models, it remains today the most popular means of comparison for protocol suites.

FIGURE 1.3 The lower layers



10 Chapter 1 • Internetworking

The OSI reference model has 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)

Figure 1.4 shows the functions defined at each layer of the OSI model. With this in hand, you're now ready to explore each layer's function in detail.

FIGURE 1.4 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

The Application Layer

The *Application layer* of the OSI model marks the spot where users actually communicate to the computer. This layer only comes into play when it's apparent that access to the network is going to be needed soon. Take the case of Internet Explorer (IE). You could uninstall every trace of networking components from a system, such as TCP/IP, NIC card, etc., and you could still use IE to view a local HTML document—no problem. But things would definitely get messy if you tried to do something like view an HTML document that must be retrieved using HTTP, or nab a file with FTP. That's because IE will respond to requests such as those by attempting to access the Application layer. And what's happening is that the Application layer is acting as an interface between the actual application program—which isn't at all a part of the layered structure—and the next layer down, by providing ways for the application to send information down through the protocol stack. In other words, IE doesn't truly reside within the Application layer—it interfaces with Application-layer protocols when it needs to deal with remote resources.

The Application layer is also responsible for identifying and establishing the availability of the intended communication partner, and determining whether sufficient resources for the intended communication exist.

These tasks are important because computer applications sometimes require more than only desktop resources. Often, they'll unite communicating components from more than one network application. Prime examples are file transfers and e-mail, as well as enabling remote access, network management activities, client/server processes, and information location. Many network applications provide services for communication over enterprise networks, but for present and future internetworking, the need is fast developing to reach beyond the limits of current physical networking. Today, transactions and information exchanges between organizations are broadening to require internetworking applications such as the following:

World Wide Web (WWW) Connects countless servers (the number seems to grow with each passing day) presenting diverse formats. Most are multimedia and can include graphics, text, video, and sound. (And as pressure to keep up the pace mounts, websites are only getting slicker and snappier. Keep in mind, the snazzier the site, the more resources it requires. You'll see why I mention this later.) Netscape Navigator and IE simplify both accessing and viewing websites.

E-mail gateways Versatile; can use Simple Mail Transfer Protocol (SMTP) or the X.400 standard to deliver messages between different e-mail applications.

Electronic data interchange (EDI) A composite of specialized standards and processes that facilitates the flow of tasks such as accounting, shipping/receiving, and order and inventory tracking between businesses.

Special interest bulletin boards Include the many Internet chat rooms where people can "meet" (connect) and communicate with each other either by posting messages or by typing a live conversation. They can also share public-domain software.

Internet navigation utilities Include applications such as Gopher and WAIS, as well as search engines such as Google and Yahoo!, which help users locate the resources and information they need on the Internet.

Financial transaction services Target the financial community. They gather and sell information pertaining to investments, market trading, commodities, currency exchange rates, and credit data to their subscribers.

The Presentation Layer

The *Presentation layer* gets its name from its purpose: It presents data to the Application layer and is responsible for data translation and code formatting.

This layer is essentially a translator and provides coding and conversion functions. A successful data-transfer technique is to adapt the data into a standard format before transmission. Computers are configured to receive this generically formatted data and then convert the data back into its native format for actual reading (for example, EBCDIC to ASCII). By providing translation services, the Presentation layer ensures that data transferred from the Application layer of one system can be read by the Application layer of another one.

12 Chapter 1 • Internetworking

The OSI has protocol standards that define how standard data should be formatted. Tasks like data compression, decompression, encryption, and decryption are associated with this layer. Some Presentation layer standards are involved in multimedia operations too. The following serve to direct graphic and visual image presentation:

PICT A picture format used by Macintosh programs for transferring QuickDraw graphics.

TIFF Tagged Image File Format; a standard graphics format for high-resolution, bit-mapped images.

JPEG Photo standards brought to us by the Joint Photographic Experts Group.

Other standards guide movies and sound:

MIDI Musical Instrument Digital Interface (sometimes called Musical Instrument Device Interface), used for digitized music.

MPEG Increasingly popular Moving Picture Experts Group standard for the compression and coding of motion video for CDs. It provides digital storage and bit rates up to 1.5Mbps.

QuickTime For use with Macintosh programs; manages audio and video applications.

RTF Rich Text Format, a file format that lets you exchange text files between different word processors, even in different operating systems.

The Session Layer

The *Session layer* is responsible for setting up, managing, and then tearing down sessions between Presentation layer entities. This layer also provides dialogue control between devices, or nodes. It coordinates communication between systems, and serves to organize their communication by offering three different modes: *simplex*, *half duplex*, and *full duplex*. To sum up, the Session layer basically keeps different applications' data separate from other applications' data.

The following are some examples of Session layer protocols and interfaces (according to Cisco):

Network File System (NFS) Developed by Sun Microsystems and used with TCP/IP and Unix workstations to allow transparent access to remote resources.

Structured Query Language (SQL) Developed by IBM to provide users with a simpler way to define their information requirements on both local and remote systems.

Remote Procedure Call (RPC) A broad client/server redirection tool used for disparate service environments. Its procedures are created on clients and performed on servers.

X Window Widely used by intelligent terminals for communicating with remote Unix computers, allowing them to operate as though they were locally attached monitors.

AppleTalk Session Protocol (ASP) Another client/server mechanism, which both establishes and maintains sessions between AppleTalk client and server machines.

Digital Network Architecture Session Control Protocol (DNA SCP) A DECnet Session layer protocol.

The Transport Layer

The *Transport layer* segments and reassembles data into a data stream. Services located in the Transport layer both segment and reassemble data from upper-layer applications and unite it onto the same data stream. They provide end-to-end data transport services and can establish a logical connection between the sending host and destination host on an internetwork.

Some of you are probably familiar with TCP and UDP already. (But if you're not, no worries—I'll tell you all about them in Chapter 2, "Internet Protocols.") If so, you know that both work at the Transport layer, and that TCP is a reliable service and UDP is not. This means that application developers have more options because they have a choice between the two protocols when working with TCP/IP protocols.

The Transport layer is responsible for providing mechanisms for multiplexing upper-layer applications, establishing sessions, and tearing down virtual circuits. It also hides details of any network-dependent information from the higher layers by providing transparent data transfer.



The term "reliable networking" can be used at the Transport layer. It means that acknowledgments, sequencing, and flow control will be used.

The Transport layer can be connectionless or connection-oriented. However, Cisco is mostly concerned with you understanding the connection-oriented portion of the Transport layer. The following sections will provide the skinny on the connection-oriented (reliable) protocol of the Transport layer.

Flow Control

Data integrity is ensured at the Transport layer by maintaining *flow control* and by allowing users to request reliable data transport between systems. Flow control prevents a sending host on one side of the connection from overflowing the buffers in the receiving host—an event that can result in lost data. Reliable data transport employs a connection-oriented communications session between systems, and the protocols involved ensure that the following will be achieved:

- The segments delivered are acknowledged back to the sender upon their reception.
- Any segments not acknowledged are retransmitted.
- Segments are sequenced back into their proper order upon arrival at their destination.
- A manageable data flow is maintained in order to avoid congestion, overloading, and data loss.

Connection-Oriented Communication

In reliable transport operation, a device that wants to transmit sets up a connection-oriented communication with a remote device by creating a session. The transmitting device first establishes a connection-oriented session with its peer system, which is called a *call setup*, or a *three-way handshake*. Data is then transferred; when finished, a call termination takes place to tear down the virtual circuit.

14 Chapter 1 • Internetworking

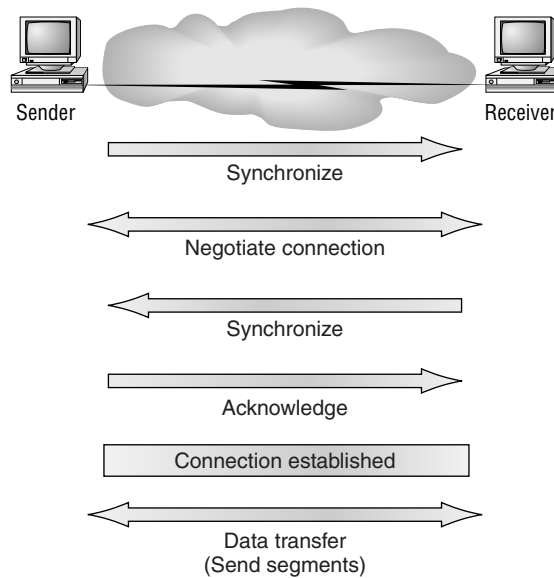
Figure 1.5 depicts a typical reliable session taking place between sending and receiving systems. Looking at it, you can see that both hosts' application programs begin by notifying their individual operating systems that a connection is about to be initiated. The two operating systems communicate by sending messages over the network confirming that the transfer is approved and that both sides are ready for it to take place. After all of this required synchronization takes place, a connection is fully established and the data transfer begins.

While the information is being transferred between hosts, the two machines periodically check in with each other, communicating through their protocol software to ensure that all is going well and that the data is being received properly.

Let me sum up the steps in the connection-oriented session—the three-way handshake—pictured in Figure 1.5:

- The first “connection agreement” segment is a request for synchronization.
- The second and third segments acknowledge the request and establish connection parameters—the rules—between hosts. The receiver's sequencing is also requested to be synchronized here, as well, so that a bidirectional connection is formed.
- The final segment is also an acknowledgment. It notifies the destination host that the connection agreement has been accepted and that the actual connection has been established. Data transfer can now begin.

FIGURE 1.5 Establishing a connection-oriented session



Sounds pretty simple, but things don't always flow so smoothly. Sometimes during a transfer, congestion can occur because a high-speed computer is generating data traffic a lot faster than the network can handle transferring. A bunch of computers simultaneously sending datagrams through

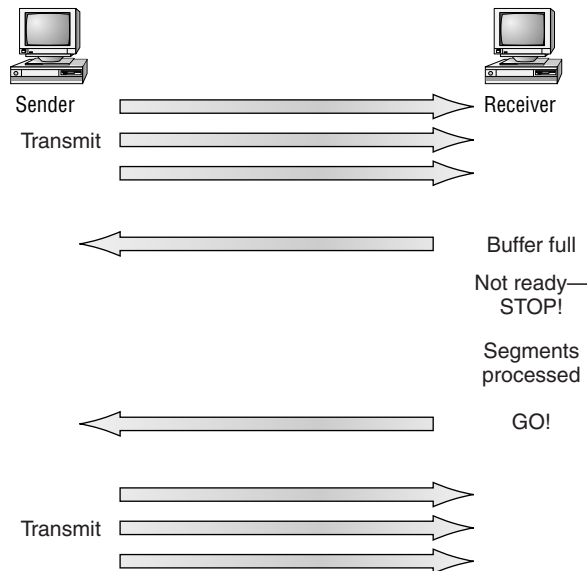
a single gateway or destination can also botch things up nicely. In the latter case, a gateway or destination can become congested even though no single source caused the problem. In either case, the problem is basically akin to a freeway bottleneck—too much traffic for too small a capacity. It's not usually one car that's the problem; there are simply too many cars on that freeway.

Okay, so what happens when a machine receives a flood of datagrams too quickly for it to process? It stores them in a memory section called a *buffer*. But this buffering action can only solve the problem if the datagrams are part of a small burst. If not, and the datagram deluge continues, a device's memory will eventually be exhausted, its flood capacity will be exceeded, and it will react by discarding any additional datagrams that arrive.

No huge worries here, though. Because of the transport function, network flood control systems really work quite well. Instead of dumping resources and allowing data to be lost, the transport can issue a "not ready" indicator to the sender, or source, of the flood (as shown in Figure 1.6). This mechanism works kind of like a stoplight, signaling the sending device to stop transmitting segment traffic to its overwhelmed peer. After the peer receiver processes the segments already in its memory reservoir—its buffer—it sends out a "ready" transport indicator. When the machine waiting to transmit the rest of its datagrams receives this "go" indicator, it resumes its transmission.

In fundamental, reliable, connection-oriented data transfer, datagrams are delivered to the receiving host in exactly the same sequence they're transmitted—and the transmission fails if this order is breached! If any data segments are lost, duplicated, or damaged along the way, a failure will transmit. This problem is solved by having the receiving host acknowledge that it has received each and every data segment.

FIGURE 1.6 Transmitting segments with flow control





Connectionless transfer is covered in Chapter 2.

A service is considered connection-oriented if it has the following characteristics:

- A virtual circuit is set up (e.g., a three-way handshake).
- It uses sequencing.
- It uses acknowledgments.
- It uses flow control.



The types of flow control are buffering, windowing, and congestion avoidance.

Windowing

Ideally, data throughput happens quickly and efficiently. And as you can imagine, it would be slow if the transmitting machine had to wait for an acknowledgment after sending each segment. But because there's time available *after* the sender transmits the data segment and *before* it finishes processing acknowledgments from the receiving machine, the sender uses the break as an opportunity to transmit more data. The quantity of data segments (measured in bytes) that the transmitting machine is allowed to send without receiving an acknowledgment for them is called a *window*.



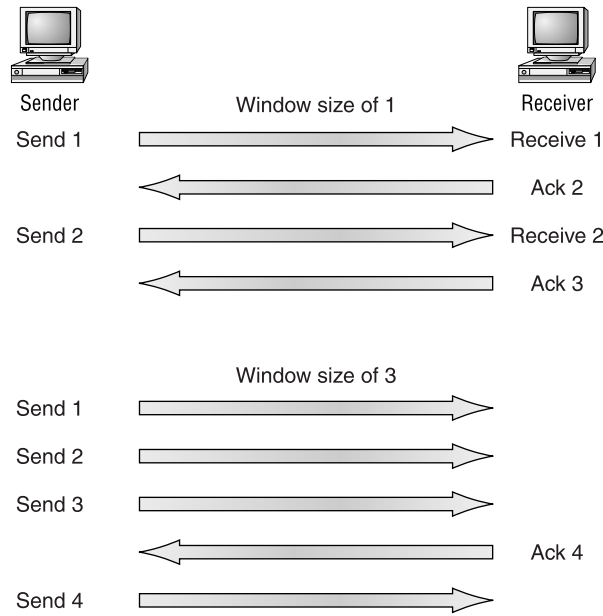
Windows are used to control the amount of outstanding, unacknowledged data segments.

So the size of the window controls how much information is transferred from one end to the other. While some protocols quantify information by observing the number of packets, TCP/IP measures it by counting the number of bytes.

As you can see in Figure 1.7, there are two window sizes—one set to 1 and one set to 3. When you've configured a window size of 1, the sending machine waits for an acknowledgment for each data segment it transmits before transmitting another. If you've configured a window size of 3, it's allowed to transmit three data segments before an acknowledgment is received. In our simplified example, both the sending and receiving machines are workstations. Reality is rarely that simple, and most often acknowledgments and packets will commingle as they travel over the network and pass through routers. Routing definitely complicates things! You'll learn about applied routing in Chapter 5, "IP Routing."



If a TCP session is set up with a window size of 2 bytes, and during the transfer stage of the session the window size changes from 2 bytes to 3 bytes, the sending host must then transmit 3 bytes before waiting for an acknowledgment instead of the 2 bytes originally set up in the virtual circuit.

FIGURE 1.7 Windowing

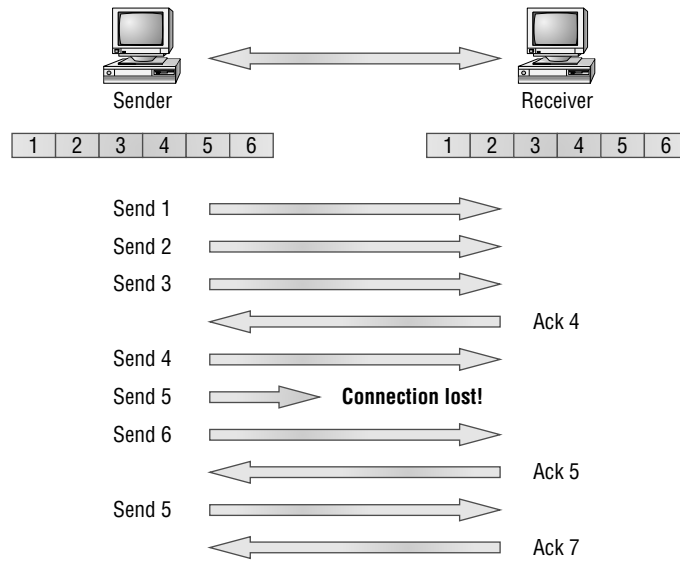
Acknowledgments

Reliable data delivery ensures the integrity of a stream of data sent from one machine to the other through a fully functional data link. It guarantees that the data won't be duplicated or lost. This is achieved through something called *positive acknowledgment with retransmission*—a technique that requires a receiving machine to communicate with the transmitting source by sending an acknowledgment message back to the sender when it receives data. The sender documents each segment it sends and waits for this acknowledgment before sending the next segment. When it sends a segment, the transmitting machine starts a timer and retransmits if it expires before an acknowledgment is returned from the receiving end.

In Figure 1.8, the sending machine transmits segments 1, 2, and 3. The receiving node acknowledges it has received them by requesting segment 4. When it receives the acknowledgment, the sender then transmits segments 4, 5, and 6. If segment 5 doesn't make it to the destination, the receiving node acknowledges that event with a request for the segment to be resent. The sending machine will then resend the lost segment and wait for an acknowledgment, which it must receive in order to move on to the transmission of segment 7.

The Network Layer

The *Network layer* (also called layer 3) manages device addressing, tracks the location of devices on the network, and determines the best way to move data, which means that the Network layer must transport traffic between devices that aren't locally attached. Routers (layer 3 devices) are specified at the Network layer and provide the routing services within an internetwork.

FIGURE 1.8 Transport layer reliable delivery

It happens like this: First, when a packet is received on a router interface, the destination IP address is checked. If the packet isn't destined for that particular router, it will look up the destination network address in the routing table. Once the router chooses an exit interface, the packet will be sent to that interface to be framed and sent out on the local network. If the router can't find an entry for the packet's destination network in the routing table, the router drops the packet.

Two types of packets are used at the Network layer: data and route updates.

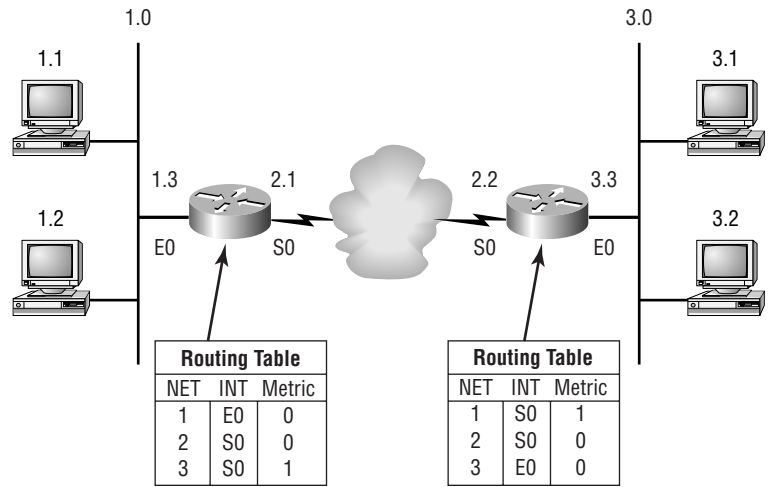
Data packets Used to transport user data through the internetwork. Protocols used to support data traffic are called *routed protocols*; examples of routed protocols are IP and IPX. You'll learn about IP addressing in Chapter 2 and Chapter 3, "IP Subnetting and Variable Length Subnet Masks (VLSMs)."

Route update packets Used to update neighboring routers about the networks connected to all routers within the internetwork. Protocols that send route update packets are called routing protocols; examples of some common ones are RIP, EIGRP, and OSPF. Route update packets are used to help build and maintain routing tables on each router.

In Figure 1.9, I've given you an example of a routing table. The routing table used in a router includes the following information:

Network addresses Protocol-specific network addresses. A router must maintain a routing table for individual routing protocols because each routing protocol keeps track of a network with a different addressing scheme. Think of it as a street sign in each of the different languages spoken by the residents that live on a particular street. So, if there were American, Spanish, and French folks on a street named "Cat," the sign would read: Cat/Gato/Chat.

FIGURE 1.9 Routing table used in a router

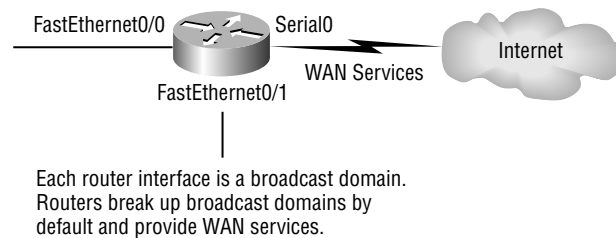


Interface The exit interface a packet will take when destined for a specific network.

Metric The distance to the remote network. Different routing protocols use different ways of computing this distance. I'm going to cover routing protocols in Chapter 5, but for now, know that some routing protocols use something called a *hop count* (the number of routers a packet passes through en route to a remote network), while others use bandwidth, delay of the line, or even tick count (1/18 of a second).

And as I mentioned earlier, routers break up broadcast domains, which means that by default, broadcasts aren't forwarded through a router. Do you remember why this is a good thing? Routers also break up collision domains, but you can also do that using layer 2 (Data Link layer) switches. Because each interface in a router represents a separate network, it must be assigned unique network identification numbers, and each host on the network connected to that router must use the same network number. Figure 1.10 shows how a router works in an internetwork.

FIGURE 1.10 A Router in an internetwork



20 Chapter 1 • Internetworking

Here are some points about routers that you should really commit to memory:

- Routers, by default, will not forward any broadcast or multicast packets.
- Routers use the logical address in a Network layer header to determine the next hop router to forward the packet to.
- Routers can use access lists, created by an administrator, to control security on the types of packets that are allowed to enter or exit an interface.
- Routers can provide layer 2 bridging functions if needed and can simultaneously route through the same interface.
- Layer 3 devices (routers in this case) provide connections between virtual LANs (VLANs).
- Routers can provide quality of service (QoS) for specific types of network traffic.



Switching and VLANs are covered in Chapter 7, “Layer 2 Switching,” and Chapter 8.

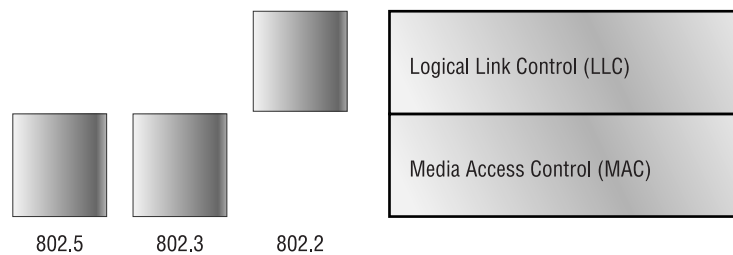
The Data Link Layer

The *Data Link layer* provides the physical transmission of the data and handles error notification, network topology, and flow control. This means that the Data Link layer will ensure that messages are delivered to the proper device on a LAN using hardware addresses, and translates messages from the Network layer into bits for the Physical layer to transmit.

The Data Link layer formats the message into pieces, each called a *data frame*, and adds a customized header containing the hardware destination and source address. This added information forms a sort of capsule that surrounds the original message in much the same way that engines, navigational devices, and other tools were attached to the lunar modules of the Apollo project. These various pieces of equipment were useful only during certain stages of space flight and were stripped off the module and discarded when their designated stage was complete. Data traveling through networks is similar.

Figure 1.11 shows the Data Link layer with the Ethernet and IEEE specifications. When you check it out, notice that the IEEE 802.2 standard is used in conjunction with and adds functionality to the other IEEE standards.

FIGURE 1.11 Data Link layer



It's important for you to understand that routers, which work at the Network layer, don't care at all about where a particular host is located. They're only concerned about where networks are located, and the best way to reach them—including remote ones. Routers are totally obsessive when it comes to networks. And for once, this is a good thing! It's the Data Link layer that's responsible for the actual unique identification of each device that resides on a local network.

For a host to send packets to individual hosts on a local network as well as transmitting packets between routers, the Data Link layer uses hardware addressing. Each time a packet is sent between routers, it's framed with control information at the Data Link layer, but that information is stripped off at the receiving router and only the original packet is left completely intact. This framing of the packet continues for each hop until the packet is finally delivered to the correct receiving host. It's really important to understand that the packet itself is never altered along the route; it's only encapsulated with the type of control information required for it to be properly passed on to the different media types.

The IEEE Ethernet Data Link layer has two sublayers:

Media Access Control (MAC) 802.3 Defines how packets are placed on the media. Contention media access is “first come/first served” access where everyone shares the same bandwidth—hence the name. Physical addressing is defined here, as well as logical topologies. What's a logical topology? It's the signal path through a physical topology. Line discipline, error notification (not correction), ordered delivery of frames, and optional flow control can also be used at this sublayer.

Logical Link Control (LLC) 802.2 Responsible for identifying Network layer protocols and then encapsulating them. An LLC header tells the Data Link layer what to do with a packet once a frame is received. It works like this: A host will receive a frame and look in the LLC header to find out where the packet is destined for—say, the IP protocol at the Network layer. The LLC can also provide flow control and sequencing of control bits.

The switches and bridges I talked about near the beginning of the chapter both work at the Data Link layer and filter the network using hardware (MAC) addresses. We will look at these in the following section.

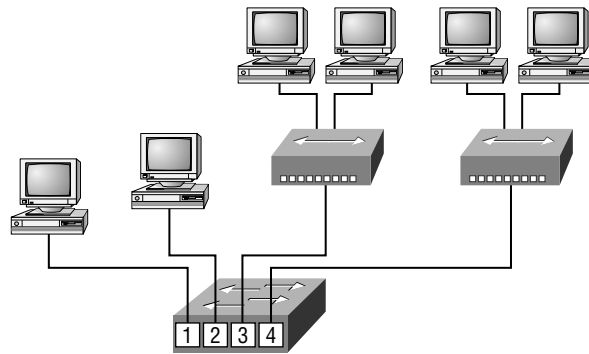
Switches and Bridges at the Data Link Layer

Layer 2 switching is considered hardware-based bridging because it uses specialized hardware called an *application-specific integrated circuit (ASIC)*. ASICs can run up to gigabit speeds with very low latency rates.



Latency is the time measured from when a frame enters a port to the time it exits a port.

Bridges and switches read each frame as it passes through the network. The layer 2 device then puts the source hardware address in a filter table and keeps track of which port the frame was received on. This information (logged in the bridge's or switch's filter table) is what helps the machine determine the location of the specific sending device. Figure 1.12 shows a switch in an internetwork.

FIGURE 1.12 A switch in an internetwork

Each segment has its own collision domain.
All segments are in the same broadcast domain.

The real estate business is all about location, location, location, and it's the same way for both layer 2 and layer 3 devices. Though both need to be able to negotiate the network, it's crucial to remember that they're concerned with very different parts of it. Primarily, layer 3 machines (such as routers) need to locate specific networks, whereas layer 2 machines (switches and bridges) need to eventually locate specific devices. So, networks are to routers as individual devices are to switches and bridges. And routing tables that "map" the internetwork are for routers, as filter tables that "map" individual devices are for switches and bridges.

After a filter table is built on the layer 2 device, it will only forward frames to the segment where the destination hardware address is located. If the destination device is on the same segment as the frame, the layer 2 device will block the frame from going to any other segments. If the destination is on a different segment, the frame can only be transmitted to that segment. This is called *transparent bridging*.

When a switch interface receives a frame with a destination hardware address that isn't found in the device's filter table, it will forward the frame to all connected segments. If the unknown device that was sent the "mystery frame" replies to this forwarding action, the switch updates its filter table regarding that device's location. But in the event the destination address of the transmitting frame is a broadcast address, the switch will forward all broadcasts to every connected segment by default.

All devices that the broadcast is forwarded to are considered to be in the same broadcast domain. This can be a problem; layer 2 devices propagate layer 2 broadcast storms that choke performance, and the only way to stop a broadcast storm from propagating through an internetwork is with a layer 3 device—a router.

The biggest benefit of using switches instead of hubs in your internetwork is that each switch port is actually its own collision domain. (Conversely, a hub creates one large collision domain.) But even armed with a switch, you still can't break up broadcast domains. Neither switches nor bridges will do that. They'll typically simply forward all broadcasts instead.

Another benefit of LAN switching over hub-centered implementations is that each device on every segment plugged into a switch can transmit simultaneously. At least, they can as long as there is only one host on each port and a hub isn't plugged into a switch port. As you might have guessed, hubs only allow one device per network segment to communicate at a time.

Each network segment connected to the switch must have the same type of devices attached. What this means to you and me is that you can connect an Ethernet hub into a switch port and then connect multiple Ethernet hosts into the hub, but you can't mix Token Ring hosts in with the Ethernet gang on the same segment. Mixing hosts in this manner is called *media translation*, and Cisco says you've just got to have a router around if you need to provide this service, although I have found this not to be true in reality—but remember, we're studying for the CCNA exam here, right?

The Physical Layer

Finally arriving at the bottom, we find that the *Physical layer* does two things: It sends bits and receives bits. Bits come only in values of 1 or 0—a Morse code with numerical values. The Physical layer communicates directly with the various types of actual communication media. Different kinds of media represent these bit values in different ways. Some use audio tones, while others employ *state transitions*—changes in voltage from high to low and low to high. Specific protocols are needed for each type of media to describe the proper bit patterns to be used, how data is encoded into media signals, and the various qualities of the physical media's attachment interface.

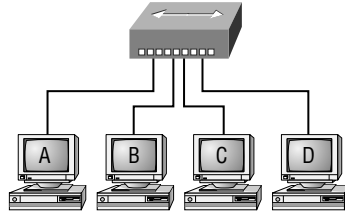
The Physical layer specifies the electrical, mechanical, procedural, and functional requirements for activating, maintaining, and deactivating a physical link between end systems. This layer is also where you identify the interface between the *data terminal equipment (DTE)* and the *data communication equipment (DCE)*. Some old-phone-company employees still call DCE data circuit-terminating equipment. The DCE is usually located at the service provider, while the DTE is the attached device. The services available to the DTE are most often accessed via a modem or *channel service unit/data service unit (CSU/DSU)*.

The Physical layer's connectors and different physical topologies are defined by the OSI as standards, allowing disparate systems to communicate. The CCNA exam is only interested in the IEEE Ethernet standards.

Hubs at the Physical Layer

A *hub* is really a multiple-port repeater. A repeater receives a digital signal and reamplifies or regenerates that signal, and then forwards the digital signal out all active ports without looking at any data. An active hub does the same thing. Any digital signal received from a segment on a hub port is regenerated or reamplified and transmitted out all ports on the hub. This means all devices plugged into a hub are in the same collision domain as well as in the same broadcast domain. Figure 1.13 shows a hub in a network.

Hubs, like repeaters, don't examine any of the traffic as it enters and is then transmitted out to the other parts of the physical media. Every device connected to the hub, or hubs, must listen if a device transmits. A physical star network—where the hub is a central device and cables extend in all directions out from it—is the type of topology a hub creates. Visually, the design really does resemble a star, whereas Ethernet networks run a logical bus topology, meaning that the signal has to run from end to end of the network.

FIGURE 1.13 A hub in a network

All devices in the same collision domain
All devices in the same broadcast domain
Devices share the same bandwidth

Ethernet Networking

Ethernet is a contention media access method that allows all hosts on a network to share the same bandwidth of a link. Ethernet is popular because it's readily scalable, meaning that it's comparatively easy to integrate new technologies, such as Fast Ethernet and Gigabit Ethernet, into an existing network infrastructure. It's also relatively simple to implement in the first place, and with it, troubleshooting is reasonably straightforward. Ethernet uses both Data Link and Physical layer specifications, and this section of the chapter will give you both the Data Link and Physical layer information you need to effectively implement, troubleshoot, and maintain an Ethernet network.

Ethernet networking uses *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*, a protocol that helps devices share the bandwidth evenly without having two devices transmit at the same time on the network medium. CSMA/CD was created to overcome the problem of those collisions that occur when packets are transmitted simultaneously from different nodes. And trust me—good collision management is crucial, because when a node transmits in a CSMA/CD network, all the other nodes on the network receive and examine that transmission. Only bridges and routers can effectively prevent a transmission from propagating throughout the entire network!

So, how does the CSMA/CD protocol work? Like this: When a host wants to transmit over the network, it first checks for the presence of a digital signal on the wire. If all is clear (no other host is transmitting), the host will then proceed with its transmission. But it doesn't stop there. The transmitting host constantly monitors the wire to make sure no other hosts begin transmitting. If the host detects another signal on the wire, it sends out an extended jam signal that causes all nodes on the segment to stop sending data (think busy signal). The nodes respond to that jam signal by waiting a while before attempting to transmit again. Backoff algorithms determine when the colliding stations can retransmit. If collisions keep occurring after 15 tries, the nodes attempting to transmit will then time out. Pretty clean!

When a collision occurs on an Ethernet LAN

- A jam signal informs all devices that a collision occurred.
- The collision invokes a random backoff algorithm.
- Each device on the Ethernet segment stops transmitting for a short time until the timers expire.

The effects of having a CSMA/CD network sustaining heavy collisions include

- Delay
- Low throughput
- Congestion



Backoff on an 802.3 network is the retransmission delay that's enforced when a collision occurs. When a collision occurs, a host will resume transmission after the forced time delay has expired.

In the following sections, I am going to cover Ethernet in detail at both the Data Link layer (layer 2) and the Physical layer (layer 1).

Half- and Full-Duplex Ethernet

Half-duplex Ethernet is defined in the original 802.3 Ethernet; Cisco says it uses only one wire pair with a digital signal running in both directions on the wire. Certainly, the IEEE specifications discuss the process of half duplex somewhat differently, but what Cisco is talking about is a general sense of what is happening here with Ethernet.

It also uses the CSMA/CD protocol to help prevent collisions and to permit retransmitting if a collision does occur. If a hub is attached to a switch, it must operate in half-duplex mode because the end stations must be able to detect collisions. Half-duplex Ethernet—typically 10BaseT—is only about 30 to 40 percent efficient as Cisco sees it, because a large 10BaseT network will usually only give you 3 to 4Mbps—at most.

But full-duplex Ethernet uses two pairs of wires, instead of one wire pair like half duplex. And full duplex uses a point-to-point connection between the transmitter of the transmitting device and the receiver of the receiving device. This means that with full-duplex data transfer, you get a faster data transfer compared to half duplex. And because the transmitted data is sent on a different set of wires than the received data, no collisions will occur.

The reason you don't need to worry about collisions is because now it's like a freeway with multiple lanes instead of the single-lane road provided by half duplex. Full-duplex Ethernet is supposed to offer 100 percent efficiency in both directions—e.g., you can get 20Mbps with a 10Mbps Ethernet running full duplex, or 200Mbps for Fast Ethernet. But this rate is something known as an aggregate rate, which translates as “You're supposed to get” 100 percent efficiency. No guarantees, in networking as in life.



Full-duplex Ethernet requires a point-to-point connection when only two nodes are present. You can run full duplex with just about any device except a hub.

26 Chapter 1 • Internetworking

Full-duplex Ethernet can be used in three situations:

- With a connection from a switch to a host
- With a connection from a switch to a switch
- With a connection from a host to a host using a crossover cable

Now, if it's capable of all that speed, why wouldn't it deliver? Well, when a full-duplex Ethernet port is powered on, it first connects to the remote end, and then negotiates with the other end of the Fast Ethernet link. This is called an *auto-detect mechanism*. This mechanism first decides on the exchange capability, which means it checks to see if it can run at 10 or 100Mbps. It then checks to see if it can run full duplex, and if it can't, it will run half duplex.



Remember that half-duplex Ethernet shares a collision domain and provides a lower effective throughput than full-duplex Ethernet, which typically has a private collision domain and a higher effective throughput.

Ethernet at the Data Link Layer

Ethernet at the Data Link layer is responsible for Ethernet addressing, commonly referred to as hardware addressing or MAC addressing. Ethernet is also responsible for framing packets received from the Network layer and preparing them for transmission on the local network through the Ethernet contention media access method. There are four different types of Ethernet frames available:

- Ethernet_II
- IEEE 802.3
- IEEE 802.2
- SNAP

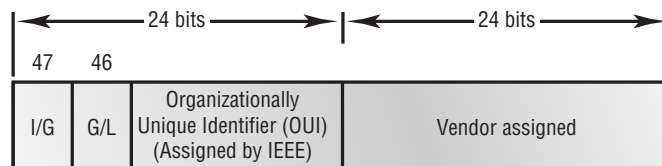
I'll go over all four of the available Ethernet frames in the upcoming sections.

Ethernet Addressing

Here's where we get into how Ethernet addressing works. It uses the *Media Access Control (MAC) address* burned into each and every Ethernet Network Interface Card (NIC). The MAC, or hardware address, is a 48-bit (6-byte) address written in a hexadecimal format.

Figure 1.14 shows the 48-bit MAC addresses and how the bits are divided.

FIGURE 1.14 Ethernet addressing using MAC addresses



The *organizationally unique identifier (OUI)* is assigned by the IEEE to an organization. It's composed of 24 bits, or 3 bytes. The organization, in turn, assigns a globally administered address (24 bits, or 3 bytes) that is unique (supposedly, again—no guarantees) to each and every adapter they manufacture. Look closely at the figure. The high-order bit is the Individual/Group (I/G) bit. When it has a value of 0, we can assume that the address is the MAC address of a device and may well appear in the source portion of the MAC header. When it is a 1, we can assume that the address represents either a broadcast or multicast address in Ethernet, or a broadcast or functional address in TR and FDDI (who really knows about FDDI?). The next bit is the G/L bit (also known as U/L, where U means universal). When set to 0, this bit represents a globally administered address (as by the IEEE). When the bit is a 1, it represents a locally governed and administered address (as in DECnet). The low-order 24 bits of an Ethernet address represent a locally administered or manufacturer-assigned code. This portion commonly starts with 24 0s for the first card made and continues in order until there are 24 1s for the last (16,777,216th) card made. You'll find that many manufacturers use these same six hex digits as the last six characters of their serial number on the same card.



MAC addresses are also part of an IPX/SPX configuration of a host—or were when IPX/SPX was still used, of course.

Ethernet Frames

The Data Link layer is responsible for combining bits into bytes and bytes into frames. Frames are used at the Data Link layer to encapsulate packets handed down from the Network layer for transmission on a type of media access. There are three types of media access methods: contention (Ethernet), token passing (Token Ring and FDDI), and polling (IBM mainframes and 100VG-AnyLAN).

The function of Ethernet stations is to pass data frames between each other using a group of bits known as a MAC frame format. This provides error detection from a cyclic redundancy check (CRC). But remember—this is error detection, not error correction. The 802.3 frames and Ethernet frame are shown in Figure 1.15.

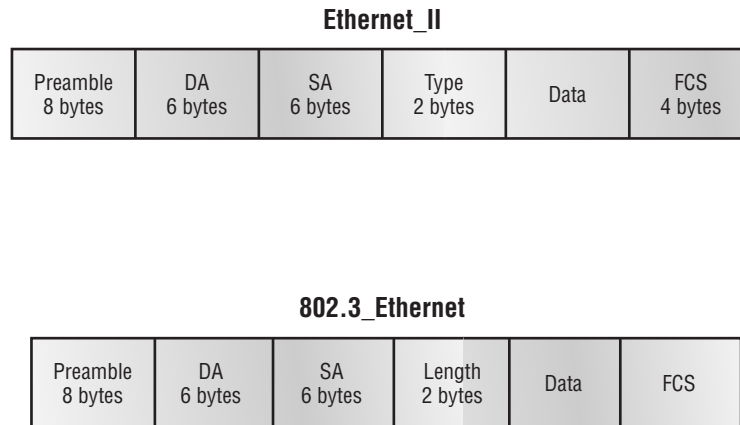


Encapsulating a frame within a different type of frame is called *tunneling*.

Following are the details of the different fields in the 802.3 and Ethernet frame types:

Preamble An alternating 1,0 pattern provides a 5MHz clock at the start of each packet, which allows the receiving devices to lock the incoming bit stream.

Start Frame Delimiter (SFD)/Synch The preamble is seven octets and the SFD is one octet (synch). The SFD is 10101011, where the last pair of 1s allows the receiver to come into the alternating 1,0 pattern somewhere in the middle and still sync up and detect the beginning of the data.

FIGURE 1.15 802.3 and Ethernet frame formats

Destination Address (DA) This transmits a 48-bit value using the least significant bit (LSB) first. The DA is used by receiving stations to determine whether an incoming packet is addressed to a particular node. The destination address can be an individual address, or a broadcast or multicast MAC address. Remember that a broadcast is all 1s (or Fs in hex) and is sent to all devices, but a multicast is sent only to a similar subset of nodes on a network.



Hex is short for hexadecimal, which is a numbering system that uses the first six letters of the alphabet (A through F) to extend beyond the available 10 digits in the decimal system. Hexadecimal has a total of 16 digits.

Source Address (SA) The SA is a 48-bit MAC address used to identify the transmitting device, and it uses the LSB first. Broadcast and multicast address formats are illegal within the SA field.

Length or Type 802.3 uses a Length field, but the Ethernet frame uses a Type field to identify the Network layer protocol. 802.3 cannot identify the upper-layer protocol and must be used with a proprietary LAN—IPX, for example.

Data This is a packet sent down to the Data Link layer from the Network layer. The size can vary from 64 to 1500 bytes.

Frame Check Sequence (FCS) FCS is a field at the end of the frame that's used to store the CRC.

Let's pause here for a minute and take a look at some frames caught on our trusty Etherpeek network analyzer. You can see that the frame below has only three fields: Destination, Source, and Type (shown as Protocol Type on this analyzer):

```
Destination: 00:60:f5:00:1f:27
Source:      00:60:f5:00:1f:2c
Protocol Type: 08-00 IP
```

This is an Ethernet_II frame. Notice that the type field is IP, or 08-00 in hexadecimal.

The next frame has the same fields, so it must be an Ethernet_II frame too:

```
Destination: ff:ff:ff:ff:ff:ff Ethernet Broadcast
Source:      02:07:01:22:de:a4
Protocol Type: 81-37 NetWare
```

I included this one so you could see that the frame can carry more than just IP—it can also carry IPX, or 81-37h. Did you notice that this frame was a broadcast? You can tell because the destination hardware address is all 1s in binary, or all Fs in hexadecimal.

Now, pay special attention to the length field in the next frame; this must be an 802.3 frame:

```
Flags:      0x80 802.3
Status:     0x00
Packet Length: 64
Timestamp:  12:45:45.192000 06/26/1998
Destination: ff:ff:ff:ff:ff:ff Ethernet Broadcast
Source:     08:00:11:07:57:28
Length:     34
```

The problem with this frame is this: How do you know which protocol this packet is going to be handed to at the Network layer? It doesn't specify in the frame, so it must be IPX. Why? Because when Novell created the 802.3 frame type (before the IEEE did and called it 802.3 Raw), Novell was pretty much the only LAN server out there. So, Novell assumed that if you were running a LAN, it must be IPX, and they didn't include any Network layer protocol field information in the 802.3 frame.

802.2 and SNAP

Since the 802.3 Ethernet frame cannot by itself identify the upper-layer (Network) protocol, it obviously needs some help. The IEEE defined the 802.2 LLC specifications to provide this function and more. Figure 1.16 shows the IEEE 802.3 with LLC (802.2) and the Subnetwork Access Protocol (SNAP) frame types. The LLC header information is added to the data portion of the frame. Now let's take a look at an 802.2 frame and SNAP captured from our analyzer.

802.2 Frame

The following is an 802.2 frame captured with a protocol analyzer:

```
Flags:      0x80 802.3
Status:     0x02 Truncated
Packet Length:64
Slice Length: 51
Timestamp:  12:42:00.592000 03/26/1998
Destination: ff:ff:ff:ff:ff:ff Ethernet Broadcast
Source:     00:80:c7:a8:f0:3d
```

30 Chapter 1 • Internetworking

```

LLC Length: 37
Dest. SAP: 0xe0 NetWare
Source SAP: 0xe0 NetWare Individual LLC
             SublayerManagement Function
Command: 0x03 Unnumbered Information

```

You can see that the first frame has a Length field, so it's probably an 802.3, right? Maybe. Look again. It also has a DSAP and an SSAP, so it's not an 802.3. It has to be an 802.2 frame. (Remember—an 802.2 frame is an 802.3 frame with the LLC information in the data field of the header so we know what the upper-layer protocol is.)

SNAP Frame

The SNAP frame has its own protocol field to identify the upper-layer protocol. This is really a way to allow an Ethernet_II Ether-Type field to be used in an 802.3 frame. Even though the following network trace shows a protocol field, it is actually an Ethernet_II type (Ether-Type) field:

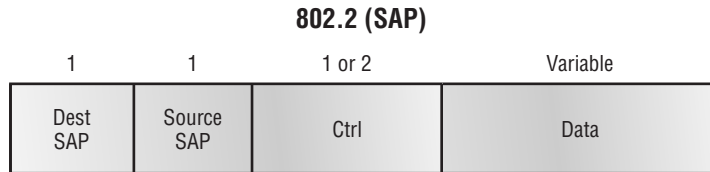
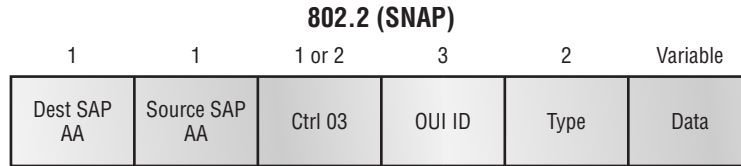
```

Flags: 0x80 802.3
Status: 0x00
Packet Length:78
Timestamp: 09:32:48.264000 01/04/2000
802.3 Header
Destination: 09:00:07:FF:FF:FF AT Ph 2 Broadcast
Source: 00:00:86:10:C1:6F
LLC Length: 60
802.2 Logical Link Control (LLC) Header
Dest. SAP: 0xAA SNAP
Source SAP: 0xAA SNAP
Command: 0x03 Unnumbered Information
Protocol: 0x080007809B AppleTalk

```

You can identify a SNAP frame because the DSAP and SSAP fields are always AA, and the Command field is always 3. This frame type was created because not all protocols worked well with the 802.3 Ethernet frame, which didn't have an Ether-Type field. To allow the proprietary protocols created by application developers to be used in the LLC frame, the IEEE defined the SNAP format that uses the exact same codes as Ethernet_II. Up until about 1997 or so, the SNAP frame was on its way out of the corporate market. However, the new 802.11 wireless LAN specification uses an Ethernet SNAP field to identify the Network layer protocol. Cisco also still uses a SNAP frame with their proprietary protocol Cisco Discovery Protocol (CDP)—something I'm going to talk about in Chapter 9, "Managing a Cisco Internetwork."

FIGURE 1.16 802.2 and SNAP



Ethernet at the Physical Layer

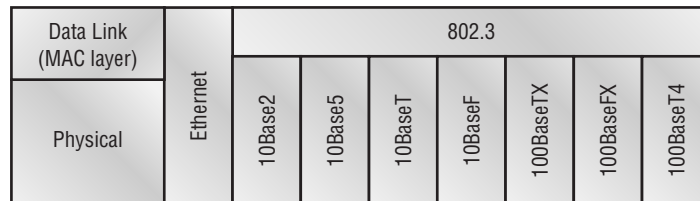
Ethernet was first implemented by a group called DIX (Digital, Intel, and Xerox). They created and implemented the first Ethernet LAN specification, which the IEEE used to create the IEEE 802.3 Committee. This was a 10Mbps network that ran on coax, and then eventually twisted-pair and fiber physical media.

The IEEE extended the 802.3 Committee to two new committees known as 802.3u (Fast Ethernet) and 802.3ab (Gigabit Ethernet on category 5) and then finally 802.3ae (10Gbps over fiber and coax).

Figure 1.17 shows the IEEE 802.3 and original Ethernet Physical layer specifications.

When designing your LAN, it's really important to understand the different types of Ethernet media available to you. Sure, it would be great to run Gigabit Ethernet to each desktop and 10Gbps between switches, and although this might happen one day, justifying the cost of that network today would be pretty difficult. But if you mix and match the different types of Ethernet media methods currently available, you can come up with a cost-effective network solution that works great.

FIGURE 1.17 Ethernet Physical layer specifications



32 Chapter 1 • Internetworking

The EIA/TIA (Electronic Industries Association and the newer Telecommunications Industry Alliance) is the standards body that creates the Physical layer specifications for Ethernet. The EIA/TIA specifies that Ethernet uses a *registered jack (RJ) connector* with a 4 5 wiring sequence on *unshielded twisted-pair (UTP)* cabling (RJ-45). However, the industry is moving toward calling this just an 8-pin modular connector.

Each Ethernet cable type that is 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 cable has more wire twists per foot and therefore less crosstalk. Crosstalk is the unwanted signal interference from adjacent pairs in the cable.

Near End Crosstalk (NEXT) is crosstalk measured at the transmitting end of the cable. Far End Crosstalk (FEXT) is measured at the far end from where the signal was injected into the cable. Power Sum NEXT (PSNEXT) is basically a mathematical calculation that simulates all four pairs being energized at the same time. PSNEXT calculations are used to ensure that a cable will not exceed crosstalk noise performance requirements when all pairs are operating simultaneously. PSNEXT is typically used in Gigabit Ethernet, rather than 10BaseT or 100BaseT.

Here are the original IEEE 802.3 standards:

10Base2 10Mbps, baseband technology, up to 185 meters in length. Known as *thinnet* and can support up to 30 workstations on a single segment. Uses a physical and logical bus with AUI connectors. The 10 means 10Mbps, Base means baseband technology, and the 2 means almost 200 meters. 10Base2 Ethernet cards use BNC (British Naval Connector, Bayonet Neill Concelman, or Bayonet Nut Connector) and T-connectors to connect to a network.

10Base5 10Mbps, baseband technology, up to 500 meters in length. Known as *thicknet*. Uses a physical and logical bus with AUI connectors. Up to 2500 meters with repeaters and 1024 users for all segments.

10BaseT 10Mbps using category 3 UTP wiring. Unlike the 10Base2 and 10Base5 networks, each device must connect into a hub or switch, and you can only have one host per segment or wire. Uses an RJ-45 connector (8-pin modular connector) with a physical star topology and a logical bus.

The “Base” in the preceding network standards means “baseband,” which is a signaling method for communication on the network.

Each of the 802.3 standards defines an Attachment Unit Interface (AUI), which allows a one-bit-at-a-time transfer to the Physical layer from the Data Link media access method. This allows the MAC to remain constant but means the Physical layer can support any existing and new technologies. The original AUI interface was a 15-pin connector, which allowed a transceiver (transmitter/receiver) that provided a 15-pin-to-twisted-pair conversion.

The thing is, the AUI interface cannot support 100Mbps Ethernet because of the high frequencies involved. So 100BaseT needed a new interface, and the 802.3u specifications created one called the Media Independent Interface (MII), which provides 100Mbps throughput. The MII uses a *nibble*, defined as 4 bits. Gigabit Ethernet uses a Gigabit Media Independent Interface (GMII) and transmits 8 bits at a time.

802.3u (Fast Ethernet) is compatible with 802.3 Ethernet because they share the same physical characteristics. Fast Ethernet and Ethernet use the same maximum transmission unit (MTU), use the same MAC mechanisms, and preserve the frame format that is used by 10BaseT Ethernet. Basically, Fast Ethernet is just based on an extension to the IEEE 802.3 specification, except that it offers a speed increase of 10 times that of 10BaseT.

Here are the expanded IEEE Ethernet 802.3 standards:

100BaseTX (IEEE 802.3u) EIA/TIA category 5, 6, or 7 UTP two-pair wiring. One user per segment; up to 100 meters long. It uses an RJ-45 connector with a physical star topology and a logical bus.

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

1000BaseCX (IEEE 802.3z) Copper twisted-pair called twinax (a balanced coaxial pair) that can only run up to 25 meters.

1000BaseT (IEEE 802.3ab) Category 5, four-pair UTP wiring up to 100 meters long.

1000BaseSX (IEEE 802.3z) MMF using 62.5- and 50-micron core; uses a 850 nanometer laser and can go up to 220 meters with 62.5-micron, 550 meters with 50-micron.

1000BaseLX (IEEE 802.3z) Single-mode fiber that uses a 9-micron core and 1300 nanometer laser, and can go from 3 kilometers up to 10 kilometers.

100VG-AnyLAN is a twisted-pair technology that was the first 100Mbps LAN. But since it was incompatible with Ethernet signaling techniques (it used a demand priority access method), it wasn't very popular, and is now essentially dead.

Ethernet Cabling

Ethernet cabling is an important discussion, especially if you are planning on taking the Cisco CCNA exam. The types of Ethernet cables available are:

- Straight-through cable
- Crossover cable
- Rolled cable

We will look at each in the following sections.

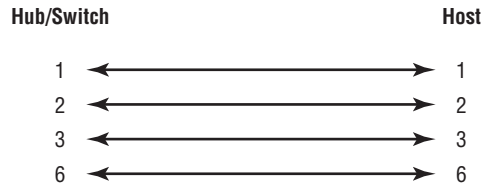
Straight-Through Cable

The *straight-through cable* is used to connect

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

Four wires are used in straight-through cable to connect Ethernet devices. It is relatively simple to create this type; Figure 1.18 shows the four wires used in a straight-through Ethernet cable.

FIGURE 1.18 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 an Ethernet-only cable and wouldn't work with Voice, Token Ring, ISDN, etc.

Crossover Cable

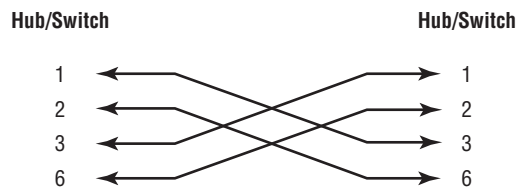
The *crossover cable* can be used to connect

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

The same four wires are used in this cable as in the straight-through cable; we just connect different pins together. Figure 1.19 shows how the four wires are used in a crossover Ethernet cable.

Notice that instead of connecting 1 to 1, etc., here we connect pins 1 to 3 and 2 to 6 on each side of the cable.

FIGURE 1.19 Crossover Ethernet cable

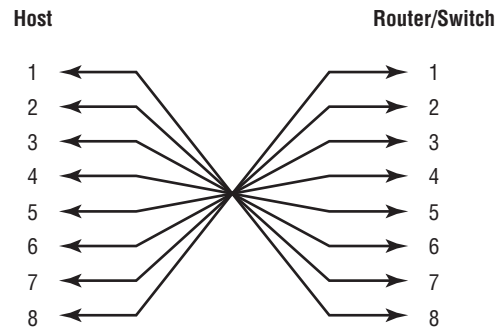


Rolled Cable

Although *rolled cable* isn't used to connect any Ethernet connections together, you can use a rolled Ethernet cable to connect a host 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 running HyperTerminal 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 1.20 shows the eight wires used in a rolled cable.

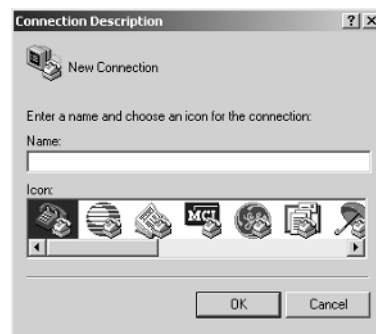
FIGURE 1.20 Rolled Ethernet cable



These are probably the easiest cables to make, because you just cut the end off on one side of a straight-through cable and reverse the end.

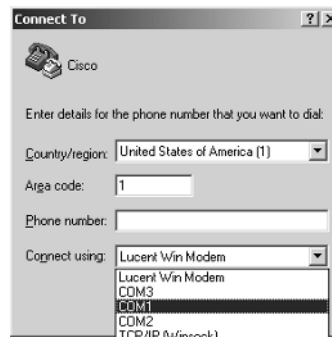
Once you have the correct cable connected from your PC to the Cisco router or switch, you can start HyperTerminal to create a console connection and configure the device. Set the configuration as follows:

1. Open HyperTerminal and enter a name for the connection. It is irrelevant what you name it, but I always just use "Cisco." Then click OK.



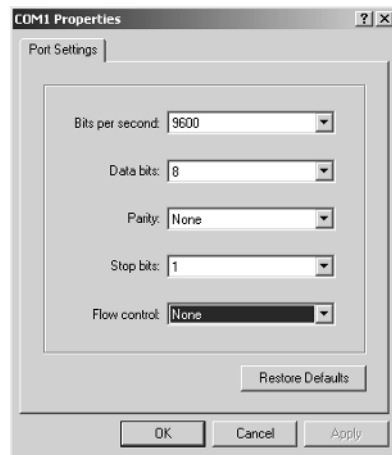
36 Chapter 1 • Internetworking

- Choose the communications port—either COM1 or COM2, whichever is open on your PC.



- Now set the port settings. The default values (2400bps and no flow control hardware) will not work; you must set the port settings as shown in Figure 1.21.

FIGURE 1.21 Port settings for a rolled cable connection



Notice that the bit rate is now set to 9600 and the flow control is set to none. At this point, you can click OK and press the Enter key, and you should be connected to your Cisco device console port.

Wireless Networking

No book on this subject today would be complete without mentioning wireless networking. That's because two years ago, it just wasn't all that common to find people using this technology—in 1996, a lot of people didn't even have an e-mail address. Sure, some did, but now everyone does, and the

same thing is happening in the wireless world. That's because wireless networking is just way too convenient not to use. I'm betting that some of you reading this probably have a wireless network at home. If not, you probably do at work. I do! For this reason, I'm now going to go over the various types of wireless networks as well as their speeds and distance limitations. Figure 1.22 shows some of the most popular types of wireless networks in use today.

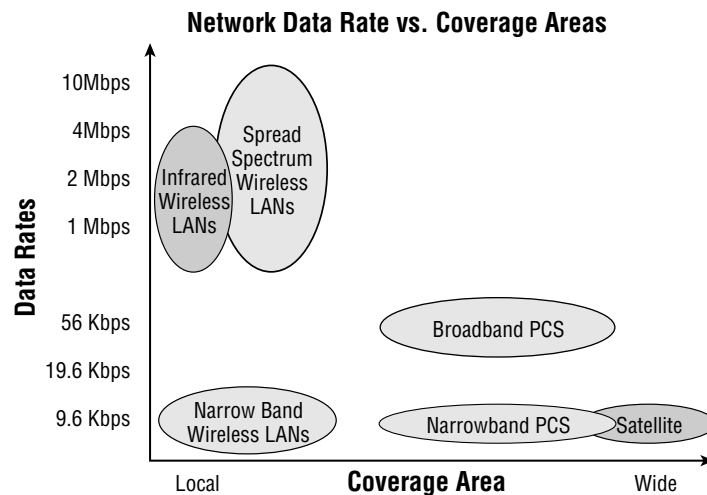
Let's discuss these various types of wireless networks plus the speed and distance of each one.

Narrowband Wireless LANs Narrowband radio does as its name suggests—it keeps the radio signal frequency as narrow as it possibly can while still being able to pass the information along. The problem of interference is avoided by directing different users onto different channel frequencies. The distance you get is decent, but the speeds available today just aren't adequate enough for corporate users. Plus, you've got to have proprietary equipment to run it on, as well as buy an FCC license to run the frequency at each site!

Personal Communication Services (PCS) Personal Communication Services (PCS) includes a whole bunch of mobile, portable, and auxiliary communications services for individuals and businesses. The Federal Communication Commission (FCC) roughly defined PCS as mobile and fixed communications options for both individuals and businesses that can be incorporated into various kinds of competing networks. Narrowband or broadband PCS is what's used today.

Narrowband PCS Again as the name implies, the narrowband PCS flavor requires a smaller serving size of the spectrum than broadband PCS does. With licenses for narrowband PCS, you get to access services such as two-way paging and/or text-based messaging. Think about people with PDAs and keyboard attachments, etc. getting and sending wireless e-mail—these subscribers are able to do this via microwave signals. With narrowband PCS you can also access cool services such as wireless telemetry—the monitoring of mobile or stationary equipment remotely. Such tasks as remotely monitoring utility meters of energy companies, commonly know as automatic meter reading, are accomplished using this technology.

FIGURE 1.22 Wireless Networks



Broadband PCS Broadband Personal Communications Service (PCS) is used for many kinds of wireless services, both mobile and fixed radio. The mobile broadband set includes both the voice and the advanced two-way data features usually available to us via small, mobile, multi-function devices such as digital camera/cell phones. In the industry, these services are commonly referred to as Mobile Telephone Services and Mobile Data Services. Sources of these services include companies that rule huge amounts of broadband PCS spectrum—AT&T Wireless, Verizon, Sprint PCS, etc.

Satellite With satellite services, the speed you get is sweet—it's up to around 1Mbps upload and 2Mbps download! But there's an annoying delay when connecting, so it doesn't work very well when you're dealing with bursty traffic. The good news is that speeds are increasing, but even so, they just can't compete with what you get via wireless LANs. The real upside to using a satellite-based network is that its geographic coverage area can be huge.

Infrared Wireless LANs Here we have pretty much the opposite. This technology works really well handling short, bursty traffic in the Personal Area Network (PAN) sector. And the speeds are increasing too, but the available range is still very short. It's commonly used for laptop-to-laptop and laptop-to-PDA transfers. The speed range we usually get is anywhere from 115Kbps to 4Mbps, but a new specification called Very Fast Infrared (VFIR) says we'll be getting speeds up to 16Mbps in the future—we'll see!

Spread Spectrum Wireless LANs Your typical wireless LANs (WLANs) use something called spread spectrum. It's a wideband radio frequency technique that the military came up with that's both reliable and secure (that's debatable). The most popular WLAN in use today is 802.11b that runs up to 11Mbps, but the new 802.11g specifications can bump that figure up to around 22Mbps (the specs say 54Mbps) and more, depending on who made your equipment. Plus, the new 802.11a lives in the 5GHz range and can run bandwidth around 50Mbps—and it's pledging over 100Mbps in the near future! But the distance is still less than what you get with the 802.11b and 802.11g 2.4GHz range models (which is about 300 feet or so). Basically, you usually find 802.11b/g used indoors and 802.11a in the shorter-distance outdoor market when more bandwidth is needed—but the market is still young and who knows what the future holds for these up-and-coming WLANs.



802.11b WLANs have a total bandwidth of up to 11Mbps and are considered "Wi-Fi" (Wireless Fidelity).

Data Encapsulation

When a host transmits data across a network to another device, the data goes through *encapsulation*: it is wrapped with protocol information at each layer of the OSI model. Each layer communicates only with its peer layer on the receiving device.

To communicate and exchange information, each layer uses *Protocol Data Units (PDUs)*. These hold the control information attached to the data at each layer of the model. They are usually attached to the header in front of the data field but can also be in the trailer, or end, of it.

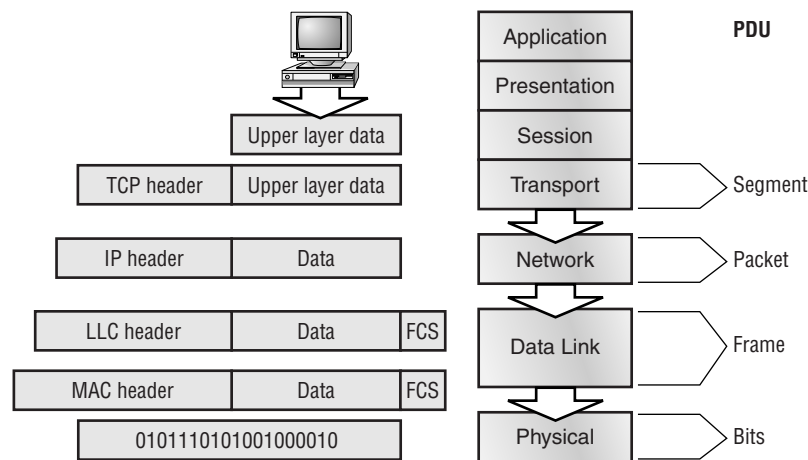
Each PDU is attached to the data by encapsulating it at each layer of the OSI model, and each has a specific name depending on the information provided in each header. This PDU information is read only by the peer layer on the receiving device. After it's read, it's stripped off, and the data is then handed to the next layer up.

Figure 1.23 shows the PDUs and how they attach control information to each layer. This figure demonstrates how the upper-layer user data is converted for transmission on the network. The data stream is then handed down to the Transport layer, which sets up a virtual circuit to the receiving device by sending over a synch packet. Next, the data stream is broken up into smaller pieces, and a Transport layer header (a PDU) is created and attached to the header of the data field; now the piece of data is called a segment. Each segment is sequenced so the data stream can be put back together on the receiving side exactly as it was transmitted.

Each segment is then handed to the Network layer for network addressing and routing through the internetwork. Logical addressing (for example, IP) is used to get each segment to the correct network. The Network layer protocol adds a control header to the segment handed down from the Transport layer, and what we have now is called a *packet* or *datagram*. Remember that the Transport and Network layers work together to rebuild a data stream on a receiving host, but it's not part of their work to place their PDUs on a local network segment—which is the only way to get the information to a router or host.

It's the Data Link layer that's responsible for taking packets from the Network layer and placing them on the network medium (cable or wireless). The Data Link layer encapsulates each packet in a *frame*, and the frame's header carries the hardware address of the source and destination hosts. If the destination device is on a remote network, then the frame is sent to a router to be routed through an internetwork. Once it gets to the destination network, a new frame is used to get the packet to the destination host.

FIGURE 1.23 Data encapsulation



40 Chapter 1 • Internetworking

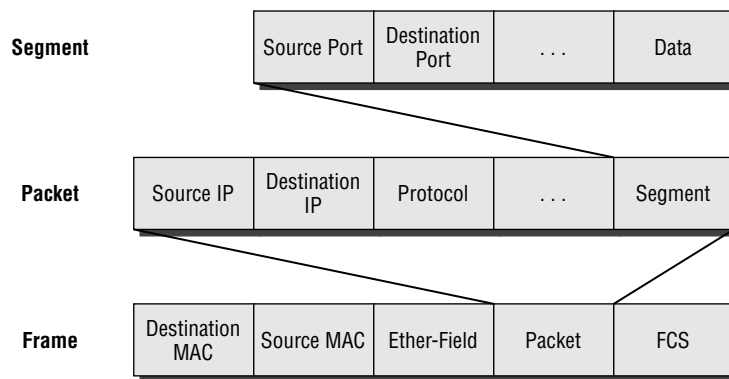
To put this frame on the network, it must first be put into a digital signal. Since a frame is really a logical group of 1s and 0s, the Physical layer is responsible for encoding these digits into a digital signal, which is read by devices on the same local network. The receiving devices will synchronize on the digital signal and extract (decode) the 1s and 0s from the digital signal. At this point the devices build the frames, run a CRC, and then check their answer against the answer in the frame's FCS field. If it matches, the packet is pulled from the frame, and what's left of the frame is discarded. This process is called *de-encapsulation*. The packet is handed to the Network layer, where the address is checked. If the address matches, the segment is pulled from the packet, and what's left of the packet is discarded. The segment is processed at the Transport layer, which rebuilds the data stream and acknowledges to the transmitting station that it received each piece. It then happily hands the data stream to the upper-layer application.

At a transmitting device, the data encapsulation method works like this:

1. User information is converted to data for transmission on the network.
2. Data is converted to segments and a reliable connection is set up between the transmitting and receiving hosts.
3. Segments are converted to packets or datagrams, and a logical address is placed in the header so each packet can be routed through an internetwork.
4. Packets or datagrams are converted to frames for transmission on the local network. Hardware (Ethernet) addresses are used to uniquely identify hosts on a local network segment.
5. Frames are converted to bits, and a digital encoding and clocking scheme is used.

To explain this in more detail using the layer addressing, I'll use Figure 1.24.

FIGURE 1.24 PDU and layer addressing



Bit 1011011100011110000

Remember that a data stream is handed down from the upper layer to the Transport layer. As technicians, we really don't care who the data stream comes from because that's really a programmer's problem. Our job is to rebuild the data stream reliably and hand it to the upper layers on the receiving device.

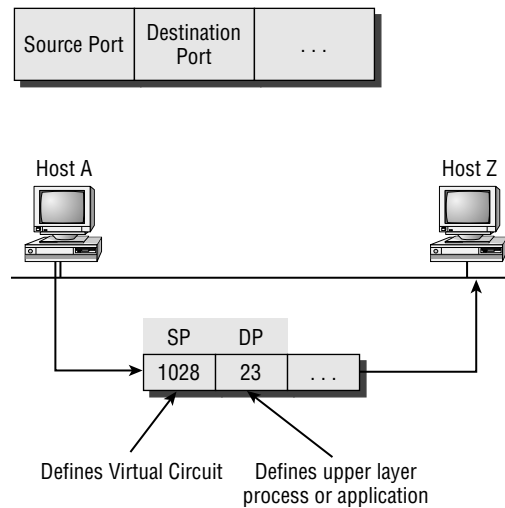
Before we go further in our discussion of Figure 1.24, let's discuss port numbers and make sure we understand them. The Transport layer uses port numbers to define both the virtual circuit and the upper layer process, as you can see from Figure 1.25.

The Transport layer takes the data stream, makes segments out of it, and establishes a reliable session by creating a virtual circuit. It then sequences (numbers) each segment and uses acknowledgments and flow control. If you're using TCP, the virtual circuit is defined by the source port number. Remember, the host just makes this up starting at port number 1024 (0 through 1023 are reserved for well-known port numbers). The destination port number defines the upper-layer process (application) that the data stream is handed to when the data stream is reliably rebuilt on the receiving host.

Now that we understand port numbers and how they are used at the Transport layer, let's go back to Figure 1.24. Once the Transport layer header information is added to the piece of data, it becomes a segment and is handed down to the Network layer, along with the destination IP address. (The destination IP address was handed down from the upper layers to the Transport layer with the data stream, and it was discovered through a name resolution method at the upper layers—probably DNS.)

The Network layer adds a header, and adds the logical addressing (IP addresses), to the front of each segment. The packet also has a protocol field that describes where it came from (either UDP or TCP)—or you can think about it as “who owns the segment.” This lets the Network layer hand the segment to the correct protocol at the Transport layer when the packet reaches the receiving host. It also has a protocol field that describes where it came from (either UDP or TCP), so it can hand it to the correct protocol at the Transport layer when it reaches the receiving host. The Network layer is responsible for finding the destination hardware address that dictates where the packet should be sent on the local network. It does this by using the Address Resolution Protocol (ARP)—something I'll talk about more in Chapter 2. IP at the Network layer looks at the destination IP address and compares that address to its own source IP address and subnet mask. If it turns out to be a local network request, the hardware address of the local host is requested via an ARP request. If the packet is destined for a remote host, IP will look for the IP address of the default gateway (router) instead.

The packet, along with the destination hardware address of either the local host or default gateway, is then handed down to the Data Link layer. The Data Link layer will add a header to the front of the packet and the piece of data then becomes a frame. (We call it a frame because both a header and a trailer are added to the packet, which makes the data resemble bookends or a frame, if you will.) At any rate, this is shown in Figure 1.24. The frame uses an Ether-Type field to describe which protocol the packet came from at the Network layer. Now, a cyclic redundancy check (CRC) is run on the frame, and the answer to the CRC is placed in the Frame Check Sequence field found in the trailer of the frame.

FIGURE 1.25 Port numbers at the Transport layer

The Frame is now ready to be handed down, one bit at a time, to the Physical layer, which will use bit timing rules to encode the data in a digital signal. Every device on the network segment will synchronize with the clock and extract the 1s and 0s from the digital signal and build a frame. After the frame is rebuilt, a CRC is run to make sure the frame is OK. If everything turns out to be all good, the hosts will check the destination address to see if the frame is for them.

If all this is making your eyes cross and your brain freeze, don't freak. I'll be going over exactly how data is encapsulated and routed through an internetwork in Chapter 5.

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 hierarchy, today most of us experience it in many aspects of our lives. It is *hierarchy* that helps us understand where things belong, how things fit together, and what functions go where. It brings order and understandability to otherwise complex models. If you want a pay raise, for instance, hierarchy dictates that you ask your boss, not your subordinate. That is 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. It 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. Then, as specific configurations are needed, the model dictates the appropriate manner 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 1.26, each with specific functions.

The following are the three layers and their typical functions:

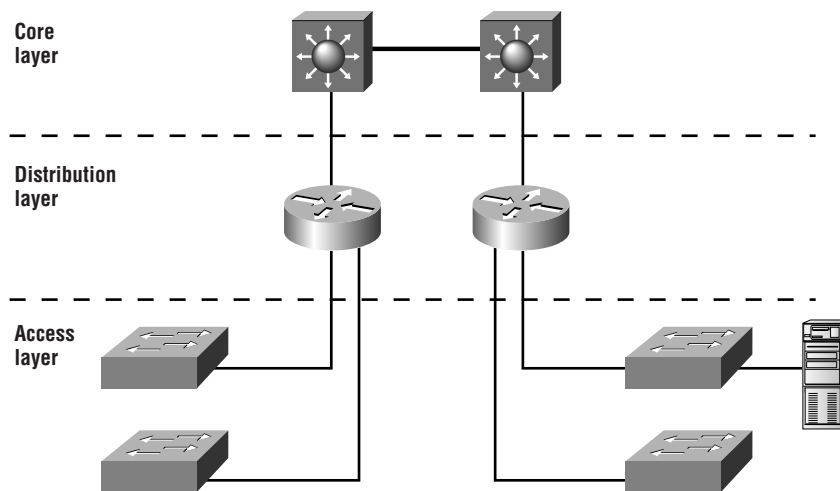
- The core layer: Backbone
- The distribution layer: Routing
- The access layer: Switching

Each layer has specific responsibilities. Remember, however, that the three layers are logical and are not necessarily physical devices. Consider the OSI model, another logical hierarchy. The 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 we might have a single device performing functions at two layers.

The definition of the layers is logical, not physical.

Now, let's take a closer look at each of the layers.

FIGURE 1.26 The Cisco hierarchical model



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. However, remember that user data is processed at the distribution layer, which forwards the requests to the core if needed.

If there is a failure in the core, *every single user* can be affected. Therefore, fault tolerance at this layer is an issue. 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:

- Don't do anything to slow down traffic. This includes using access lists, routing between virtual local area networks (VLANs), and packet filtering.
- Don't support workgroup access here.
- Avoid expanding the core (i.e., adding routers) when the internetwork grows. If performance becomes an issue in the core, give preference to upgrades over expansion.

Now, there are a few things that we want to do as we design the core. They include the following:

- Design the core for high reliability. Consider data-link technologies that facilitate both speed and redundancy, such as FDDI, Fast Ethernet (with redundant links), or even ATM.
- 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 needed. The core layer then quickly transports the request to the correct service.

The distribution layer is the place to implement policies for the network. Here you can exercise considerable flexibility in defining network operation. There are several actions that generally should be done at the distribution layer. They include the following:

- Routing
- Implementation of tools such as access lists, of packet filtering, and of queuing
- Implementation of security and network policies, including address translation and firewalls
- Redistribution between routing protocols, including static routing

- Routing between VLANs and other workgroup support functions
- Definitions of broadcast and multicast domains

Things to avoid at the distribution layer are limited to those 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. 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) access control and policies
- Creation of separate collision domains (segmentation)
- Workgroup connectivity into the distribution layer

Technologies such as DDR and Ethernet switching are frequently seen in the access layer. Static routing (instead of dynamic routing protocols) is seen here as well.

As already noted, three separate levels does not imply three separate routers. There could be fewer, or there could be more. Remember, this is a *layered* approach.

Summary

Whew! I know this seemed like the chapter that wouldn't end, but it did—and you made it! You're now armed with a ton of fundamental information; you are ready to build upon it, and are well on your way to certification.

This chapter began with a discussion of the OSI model—the seven-layer model used to help application developers design applications that can run on any type of system or network. Each layer has its special jobs and select responsibilities within the model to ensure that solid, effective communications do, in fact, occur. I provided you with complete details of each layer and discussed how Cisco views the specifications of the OSI model.

In addition, each layer in the OSI model specifies different types of devices. I described the different devices, cables, and connectors used at each layer. Remember that hubs are Physical layer devices and repeat the digital signal to all segments except the one it was received from. Switches segment the network using hardware addresses and break up collision domains. Routers break up broadcast domains (and collision domains) and use logical addressing to send packets through an internetwork.

Lastly, this chapter covered the Cisco three-layer hierarchical model. I described in detail the three layers and how each is used to help design and implement a Cisco internetwork. We are now going to move on to IP addressing in the next chapter.

Exam Essentials

Remember the possible causes of LAN traffic congestion. Too many hosts in a broadcast domain, broadcast storms, multicasting, and low bandwidth are all possible causes of LAN traffic congestion.

Understand the difference between a collision domain and a broadcast domain. A collision domain is an Ethernet term used to describe a network collection of devices in which one particular device sends a packet on a network segment, forcing every other device on that same segment to pay attention to it. A broadcast domain is where a set of all devices on a network segment hear all broadcasts sent on that segment.

Understand the difference between a hub, a bridge, a switch, and a router. Hubs create one collision domain and one broadcast domain. Bridges break up collision domains but create one large broadcast domain. They use hardware addresses to filter the network. Switches are really just multiple port bridges with more intelligence. They break up collision domains but create one large broadcast domain by default. Switches use hardware addresses to filter the network. Routers break up broadcast domains (and collision domains) and use logical addressing to filter the network.

Remember the Presentation layer protocols. PICT, TIFF, JPEG, MIDI, MPEG, QuickTime, and RTF are examples of Presentation layer protocols.

Remember the difference between connection-oriented and connectionless network services. Connection-oriented uses acknowledgments and flow control to create a reliable session. More overhead is used than in a connectionless network service. Connectionless services are used to send data with no acknowledgments or flow control. This is considered unreliable.

Remember the OSI layers. You must remember the seven layers of the OSI model and what function each layer provides. The Application, Presentation, and Session layers are upper layers and are responsible for communicating from a user interface to an application. The Transport layer provides segmentation, sequencing, and virtual circuits. The Network layer provides logical network addressing and routing through an internetwork. The Data Link layer provides framing and placing of data on the network medium. The Physical layer is responsible for taking 1s and 0s and encoding them into a digital signal for transmission on the network segment.

Remember the types of Ethernet cabling and when you would use them. The three types of cables that can be created from an Ethernet cable are: straight-through (to connect a PC's or a router's Ethernet interface to a hub or switch), crossover (to connect hub to hub, hub to switch, switch to switch, or PC to PC), and rolled (for a console connection from a PC to a router or switch).

Understand how to connect a console cable from a PC to a router and start HyperTerminal. Take a rolled cable and connect it from the COM port of the host to the console port of a router. Start HyperTerminal and set the BPS to 9600 and flow control to None.

Remember the three layers in the Cisco three-layer model. The three layers in the Cisco hierarchical model are the core, distribution, and access layers.

Written Lab 1

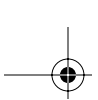
In this section, you'll complete the following labs to make sure you've got the information and concepts contained within them fully dialed in:

- Lab 1.1: OSI Questions
- Lab 1.2: Defining the OSI Layers and Devices
- Lab 1.3: Identifying Collision and Broadcast Domains

Written Lab 1.1: OSI Questions

Answer the following questions about the OSI model:

1. Which layer chooses and determines the availability of communicating partners, along with the resources necessary to make the connection; coordinates partnering applications; and forms a consensus on procedures for controlling data integrity and error recovery?
2. Which layer is responsible for converting data packets from the Data Link layer into electrical signals?
3. At which layer is routing implemented, enabling connections and path selection between two end systems?
4. Which layer defines how data is formatted, presented, encoded, and converted for use on the network?
5. Which layer is responsible for creating, managing, and terminating sessions between applications?
6. Which layer ensures the trustworthy transmission of data across a physical link and is primarily concerned with physical addressing, line discipline, network topology, error notification, ordered delivery of frames, and flow control?
7. Which layer is used for reliable communication between end nodes over the network and provides mechanisms for establishing, maintaining, and terminating virtual circuits; transport-fault detection and recovery; and controlling the flow of information?
8. Which layer provides logical addressing that routers will use for path determination?
9. Which layer specifies voltage, wire speed, and pinout cables and moves bits between devices?
10. Which layer combines bits into bytes and bytes into frames, uses MAC addressing, and provides error detection?
11. Which layer is responsible for keeping the data from different applications separate on the network?
12. Which layer is represented by frames?
13. Which layer is represented by segments?
14. Which layer is represented by packets?



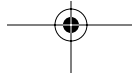
48 Chapter 1 • Internetworking

- 15. Which layer is represented by bits?
- 16. Put the following in order of encapsulation:
 - Packets
 - Frames
 - Bits
 - Segments
- 17. Which layer segments and reassembles data into a data stream?
- 18. Which layer provides the physical transmission of the data and handles error notification, network topology, and flow control?
- 19. Which layer manages device addressing, tracks the location of devices on the network, and determines the best way to move data?
- 20. What is the bit length and expression form of a MAC address?

Written Lab 1.2: Defining the OSI Layers and Devices

Fill in the blanks with the appropriate layer of the OSI or hub, switch, or router device.

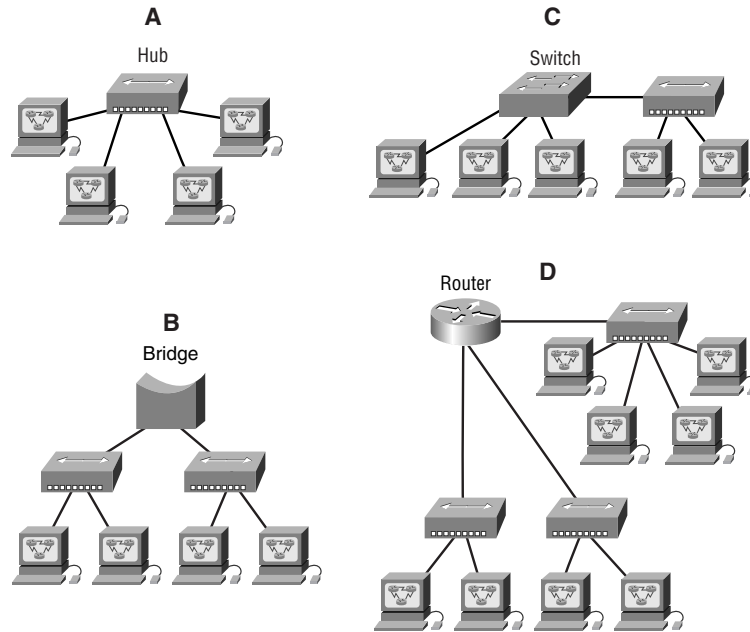
Description	Device or OSI Layer
This device sends and receives information about the Network layer.	
This layer creates a virtual circuit before transmitting between two end stations.	
This layer uses service access points.	
This device uses hardware addresses to filter a network.	
Ethernet is defined at these layers.	
This layer supports flow control and sequencing.	
This device can measure the distance to a remote network.	
Logical addressing is used at this layer.	
Hardware addresses are defined at this layer.	
This device creates one big collision domain and one large broadcast domain.	
This device creates many smaller collision domains, but the network is still one large broadcast domain.	
This device breaks up collision domains and broadcast domains.	



Written Lab 1.3: Identifying Collision and Broadcast Domains

In the following exhibit, identify the number of collision domains and broadcast domains in each specified device. Each device is represented by a letter:

1. Hub
2. Bridge
3. Switch
4. Router

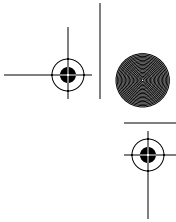


(The answers to Written Lab 1 can be found following the answers to the Review Questions for this chapter.)

Review Questions

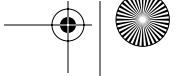
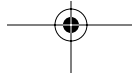
1. PDUs at the Network layer of the OSI are called what?
 - A. Transport
 - B. Frames
 - C. Packets
 - D. Segments
2. Which fields are contained within an IEEE Ethernet frame header? (Choose two.)
 - A. Source and destination MAC address
 - B. Source and destination network address
 - C. Source and destination MAC address and source and destination network address
 - D. FCS field
3. What is the maximum data rate for 802.11b wireless LANs?
 - A. 2Mbps
 - B. 10Mbps
 - C. 11Mbps
 - D. 54Mbps
4. Segmentation of a data stream happens at which layer of the OSI model?
 - A. Physical
 - B. Data Link
 - C. Network
 - D. Transport
5. Which of the following describe router functions? (Choose four.)
 - A. Packet switching
 - B. Collision prevention
 - C. Packet filtering
 - D. Broadcast domain enlargement
 - E. Internetwork communication
 - F. Broadcast forwarding
 - G. Path selection
6. Which layer of the OSI provides translation of data?
 - A. Application
 - B. Presentation
 - C. Session
 - D. Transport
 - E. Data Link

7. When data is encapsulated, which is the correct order?
- A. Data, frame, packet, segment, bit
 - B. Segment, data, packet, frame, bit
 - C. Data, segment, packet, frame, bit
 - D. Data, segment, frame, packet, bit
8. Why does the data communication industry use the layered OSI reference model? (Choose two.)
- A. It divides the network communication process into smaller and simpler components, thus aiding component development, design, and troubleshooting.
 - B. It enables equipment from different vendors to use the same electronic components, thus saving research and development funds.
 - C. It supports the evolution of multiple competing standards, and thus provides business opportunities for equipment manufacturers.
 - D. It encourages industry standardization by defining what functions occur at each layer of the model.
 - E. It provides a framework by which changes in functionality in one layer require changes in other layers.
9. What are two purposes for segmentation with a bridge?
- A. Add more broadcast domains.
 - B. Create more collision domains.
 - C. Add more bandwidth for users.
 - D. Allow more broadcasts for users.
 - E. Reduce collisions within a broadcast domain.
10. Which of the following are unique characteristics of half-duplex Ethernet when compared to full-duplex Ethernet?
- A. Half-duplex Ethernet operates in a shared collision domain.
 - B. Half-duplex Ethernet operates in a private collision domain.
 - C. Half-duplex Ethernet has higher effective throughput.
 - D. Half-duplex Ethernet has lower effective throughput.
 - E. Half-duplex Ethernet operates in a private broadcast domain.
11. You want to implement a network medium that is not susceptible to EMI. Which type of cabling should you use?
- A. Thicknet coax
 - B. Thinnet coax
 - C. Category 5 UTP cable
 - D. Fiber optic cable



52 Chapter 1 • Internetworking

12. Acknowledgements, sequencing, and flow control are characteristic of which OSI layer?
- A. Layer 2
 - B. Layer 3
 - C. Layer 4
 - D. Layer 7
13. Which of the following are types of flow control?
- A. Buffering
 - B. Cut-through
 - C. Windowing
 - D. Congestion avoidance
 - E. VLANs
14. Which of the following types of connections can use full duplex? (Choose three.)
- A. Hub to hub
 - B. Switch to switch
 - C. Host to host
 - D. Switch to hub
 - E. Switch to host
15. Which of the following describes a MAC address? (Choose two.)
- A. It is a globally unique IP address.
 - B. It is a unique address in a broadcast domain.
 - C. It is provided by the manufacturer of the NIC.
 - D. It is used as part of the IPX/SPX configuration.
 - E. It is a logical address.
16. Which of the following are considered some reasons for LAN congestion? (Choose six.)
- A. Bill Gates
 - B. Low bandwidth
 - C. Too many users in a broadcast domain
 - D. Broadcast storms
 - E. Routers
 - F. Multicasting
 - G. The addition of hubs to the network
 - H. Large amount of ARP or IPX traffic



17. Which of the following are reasons for breaking up a network into two segments with a router? (Choose two.)
- A. To create fewer broadcast domains
 - B. To create more broadcast domains
 - C. To create one large broadcast domain
 - D. To stop one segment's broadcasts from being sent to the second segment
18. How does a host on an Ethernet LAN know when to transmit after a collision has occurred?
- A. The destination host sends a request to the source for retransmission.
 - B. The jam signal indicates that the collision has been cleared.
 - C. The hosts will attempt to resume transmission after a time delay has expired.
 - D. An electric pulse indicates that the collision has cleared.
 - E. The router on the segment will signal that the collision has cleared.
19. You want to use full-duplex Ethernet instead of half duplex. Which two of the following will be benefits on your network?
- A. You will have more collision domains.
 - B. You'll have no collisions on each segment.
 - C. It should be faster.
 - D. It will be less expensive.
20. You have the following MAC address: C9-3F-32-B4-DC-19. What is the OUI portion in binary?
- A. 11000110-11000000-00011111
 - B. 1100110000111111-00011000
 - C. 11001001-00111111-00110010
 - D. 11001100-01111000-00011000

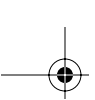
Answers to Review Questions

1. C. Protocol Data Units are used to define data at each layer of the OSI model. PDUs at the Network layer are called packets.
2. A, D. An Ethernet frame has source and destination MAC addresses, an Ether-Type field to identify the Network layer protocol, the data, and the FCS field that holds the result of the CRC.
3. C. 802.11b, also called “Wi-Fi,” has a maximum data rate of 11Mbps.
4. D. The Transport layer receives large data streams from the upper layers and breaks these up into smaller pieces called segments.
5. A, C, E, G. Routers provide packet switching, packet filtering, internetwork communication, and path selection.
6. B. The only layer of the OSI model that can change data is the Presentation layer.
7. C. The encapsulation method is: data, segment, packet, frame, bit.
8. A, D. The main advantage of a layered model is that it can allow application developers to change aspects of a program in just one layer of the layer model’s specifications. Advantages of using the OSI layered model include, but are not limited to, the following: It divides the network communication process into smaller and simpler components, thus aiding component development, design, and troubleshooting; allows multiple-vendor development through standardization of network components; encourages industry standardization by defining what functions occur at each layer of the model; allows various types of network hardware and software to communicate; and prevents changes in one layer from affecting other layers, so it does not hamper development.
9. B, C. Bridges break up collision domains, which allows more bandwidth for users.
10. A, D. Unlike full duplex, half-duplex Ethernet operates in a shared collision domain, and it has a lower effective throughput than full duplex.
11. D. Fiber optic cable provides a more secure, long-distance cable that is not susceptible to EMI interference at high speeds.
12. C. A reliable Transport layer connection uses acknowledgments to make sure all data is transmitted and received reliably. A reliable connection is defined by a virtual circuit that uses acknowledgments, sequencing, and flow control, which are characteristics of the Transport layer (layer 4).
13. A, C, D. The common types of flow control are buffering, windowing, and congestion avoidance.
14. B, C, E. Hubs cannot run full-duplex Ethernet. Full duplex must be used on a point-to-point connection between two devices capable of running full duplex. Switches and hosts can run full duplex between each other.

15. C, D. A MAC address is 6 bytes long (48 bits) and is burned into the NIC card by the manufacturer. IPX, when it was still used, used the MAC address as part of the configuration.
16. B, C, D, F, G, H. Although Bill Gates is a good answer for me, the answers are: not enough bandwidth, too many users, broadcast storms, multicasting, adding hubs for connectivity to the network, and a large amount of ARP or IPX traffic.
17. B, D. Routers, by default, break up broadcast domains, which means that broadcasts sent on one network would not be forwarded to another network by the router.
18. C. Once transmitting stations on an Ethernet segment hear a collision, they send an extended jam signal to ensure that all stations recognize the collision. After the jamming is complete, each sender waits a predetermined amount of time, plus a random time. After both timers expire, they are free to transmit.
19. B, C. No collision on a point-to-point full-duplex Ethernet segment should occur, and full-duplex Ethernet should be faster than half-duplex Ethernet.
20. C. The first three bytes of a hardware (MAC) address are considered the OUI portion and identify the manufacturer of the card. This code, which is assigned by the IEEE, is called the organizationally unique identifier (OUI). The first three bytes are C9-3F-32, which is 11001001-00111111-00110010 in binary.

Answers to Written Lab 1.1

1. The Application layer is responsible for finding the network resources broadcast from a server and adding flow control and error control (if the application developer chooses).
2. The Physical layer takes frames from the Data Link layer and encodes the 1s and 0s into a digital signal for transmission on the network medium.
3. The Network layer provides routing through an internetwork and logical addressing.
4. The Presentation layer makes sure that data is in a readable format for the Application layer.
5. The Session layer sets up, maintains, and terminates sessions between applications.
6. PDUs at the Data Link layer are called frames. As soon as you see “frame” in a question, you know the answer.
7. The Transport layer uses virtual circuits to create a reliable connection between two hosts.
8. The Network layer provides logical addressing, typically IP addressing and routing.
9. The Physical layer is responsible for the electrical and mechanical connections between devices.
10. The Data Link layer is responsible for the framing of data packets.
11. The Session layer creates sessions between different hosts’ applications.
12. The Data Link layer frames packets received from the network layer.
13. The Transport layer segments user data.
14. The Network layer creates packets out of segments handed down from the Transport layer.
15. The Physical layer is responsible for transporting 1s and 0s in a digital signal.
16. Segments, packets, frames, bits
17. Transport
18. Data Link
19. Network
20. 48 bits (6 bytes) expressed as a hexadecimal number



Answer to Written Lab 1.2

Description	Device or OSI Layer
This device sends and receives information about the Network layer.	Router
This layer creates a virtual circuit before transmitting between two end stations.	Transport
This layer uses service access points.	Data Link (LLC sublayer)
This device uses hardware addresses to filter a network.	Bridge or switch
Ethernet is defined at these layers.	Data Link and Physical
This layer supports flow control and sequencing.	Transport
This device can measure the distance to a remote network.	Router
Logical addressing is used at this layer.	Network
Hardware addresses are defined at this layer.	Data Link (MAC sublayer)
This device creates one big collision domain and one large broadcast domain.	Hub
This device creates many smaller collision domains, but the network is still one large broadcast domain.	Switch or bridge
This device breaks up collision domains and broadcast domains.	Router

Answers to Written Lab 1.3

1. Hub: One collision domain, one broadcast domain
2. Bridge: Two collision domains, one broadcast domain
3. Switch: Four collision domains, one broadcast domain
4. Router: Three collision domains, three broadcast domains



