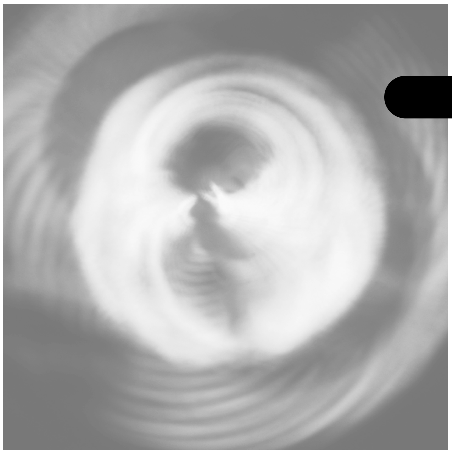

ONE

Introduction to Networking and the OSI Model

In This Chapter

- Local Area Networks
- Wide Area Networks
- Internetworking
- The Seven Layers of the OSI Model

In this chapter, we begin our journey toward the CCNA certification by examining some networking concepts key to working with Cisco routers. The most important concept is a discussion of the OSI model and how data flows across a network. Once the OSI model is understood, it will be easier to design, use, and, especially, troubleshoot Cisco networks.

Introduction to Networking

In the early days of computing, there were mainframe computers. These computers were large and centrally located, usually in a very cold and climate-controlled environment.

Although processing was performed on the mainframe, the average user did not walk up to it and start an application. Instead, he or she would sit at a terminal that was connected to the mainframe by some type of cabling. This terminal, located in a remote location, was the gateway to the processing power of the mainframe. The terminal performed very little work on its own. In fact, it merely displayed data on the monitor and processed keystrokes to send back to the mainframe. For this reason, these terminals were often called dumb terminals.

As time progressed, more and more users were connecting to the mainframe computer through the terminals. This increased the load on the mainframe, thereby slowing productivity. The mainframe computers were continually being enhanced and upgraded to keep up with the processing demand.

Technology started producing smarter terminals to decrease the load on the mainframe. When the personal computer (PC) became a reality in the late 1980s, the paradigm began to shift. PCs could connect to the mainframe in place of the dumb terminals, but more importantly, they could process data on their own. The PC revolution began, and the increasing importance of the home and office computer was realized.

As PCs began to work in conjunction with the mainframe, new technology was required to efficiently connect them. *Local Area Networks,* or *LANs,* became the term used to describe the way in which computers were connected together to share data. LANs were implemented in a business using technologies such as Ethernet and Token Ring to connect computers together using *Network Interface Cards,* or *NICs.* LAN connectivity became a new industry market, and new businesses worldwide started operations.

As more and more LANs became operational, it became necessary to link these networks together across floors, buildings, cities, and even countries; hence, the introduction of the *Wide Area Network,* or *WAN.* A WAN is a means of connecting LANs together across a distance boundary. Typical WAN connectivity was accomplished through phone lines.

Today, computers throughout the world are connected through WANs, LANs, and various combinations of the two. Perhaps the most well-known network is the Internet. The means of connecting all these networks together to achieve a desired goal is called *internetworking,* and this is where Cisco has positioned itself as the world leader.

The OSI Model

Networking evolved from the basic principle of moving data from one computer to another. The first method involved copying data to a storage media such as a floppy disk and then taking that storage media to another computer and copying the data. This was charmingly referred to as sneaker-net. As more efficient means were discovered—namely, electricity on a copper wire—networking became more popular. However, there were no standards in place. This meant that one network manufacturer implemented a different means of data transfer than another. If you had an IBM network, you purchased only IBM network devices.

In 1984, a group known as the International Organization for Standardization (ISO) created a model called the *Open Systems Interconnect (OSI).* This model defined guidelines for interoperability between network manufacturers. A company could now mix and match network devices and protocols from various manufacturers in its own network without being locked into using a single vendor. It also had a great side effect: Competition meant lower prices.

Although the OSI model defined a set of standards, it is important to note that it is merely a model. Many other models exist in the networking industry; however, understanding a single model gives us the capability of understanding other models in the future. The OSI model is the most widely taught as the foundation for this knowledge.

Why Use a Layered Model?

By using a layered model, we can categorize the procedures that are necessary to transmit data across a network. Let's explore this in more detail. Imagine that we are developers and we are about to create a new protocol for communication across a network.

First, we need to define the term *protocol*: A protocol is a set of guidelines or rules of communication. Some think of a protocol as a dialect of a language; this is erroneous. The British and the Americans both speak the same language: English. However, certain words differ in meaning between the two countries. The timing of the exchange of words between the two cultures can also lead to difficulties in complete understanding. A protocol, then, is more than just the *words* of computers. It also includes the timing and the same dictionary so that at any time, both computers using the same protocol have an exact, complete understanding of each other.

Developing a new protocol without a model would be a tedious and time-consuming task. We would need to reinvent the wheel by recreating work that has already been done by other developers. We could save time by cutting and pasting the code, but we would still need to do extensive testing. Further, when we needed to update the protocol, we would have to redesign and retest the entire protocol.

However, if we were to use a layered design, we could separate the processes into specific layers. We could then design, enhance, and test each individual layer. As the process continued, we would have a complete protocol based on a layered model. When we needed to update code, we would only have to modify the one layer that needed the updating; the rest of the layers would not be affected. This allows us to enhance specific functions easier and quicker.

Further, by using a layered model, we could then license our protocol to other developers for use in their own networks. If the protocol did not work on their chosen hardware platform, they could replace one of the layers with their own version, thereby creating multivendor compatibility.

If we did decide to enhance one of the layers, we could take just that specific layer and redistribute it to all the developers, thereby making our protocol even better.

To summarize, layered modeling allows us to:

- Create a protocol that can be designed and tested in stages, which, in turn, reduces the complexity
- Enhance functionality of the protocol without adversely affecting the other layers
- Provide multivendor compatibility
- Allow for easier troubleshooting by locating the specific layer causing the problem

How Does a Model Work?

Before defining how a model works, we must clarify one thing. The OSI model defines what each layer should do—it does not tell you how to do it. This allows developers the freedom to choose the best method they can design.

The OSI model is divided into seven layers. Figure 1–1 lists the name and order of each layer. Notice that the bottom layer is identified as the first layer.

Application	Layer 7
Presentation	Layer 6
Session	Layer 5
Transport	Layer 4
Network	Layer 3
Data Link	Layer 2
Physical	Layer 1

FIGURE 1–1 The seven layers of the OSI model

It is important to remember the order of the layers in the OSI model. Doing so creates a better understanding of the network data flow. It is also needed to pass an exam. Many acrostics can be used to remember the order, but possibly the most common is

Please	(**P**hysical	Layer 1)
Do	(**D**ata Link	Layer 2)
Not	(**N**etwork	Layer 3)
Throw	(**T**ransport	Layer 4)
Sausage	(**S**ession	Layer 5)
Pizzas	(**P**resentation	Layer 6)
Away	(**A**pplication	Layer 7)

If we wish to use this acrostic to remember the order of the OSI model, there are two important items to note. First, this acrostic starts from the bottom (Layer 1) and moves toward the top. Second, there are two Ps, so we have to remember which is the Physical and which is the Presentation.

Each layer is separated, or *encapsulated*, from each other layer. This means that each layer can function on its own. Each layer thinks it is talking directly to the same layer on the remote computer (see Figure 1–2) through a *virtual* link. Furthermore, each layer can only communicate with the layers above and below it. In fact, the layer doesn't know that any other layers even exist. For example, notice in Figure 1–3 that the Transport layer can communicate only with the Network and the Session layers.

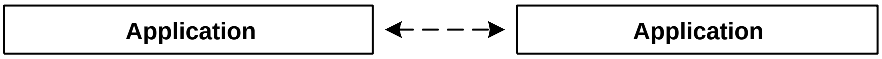

FIGURE 1–2 Virtual link between application layers

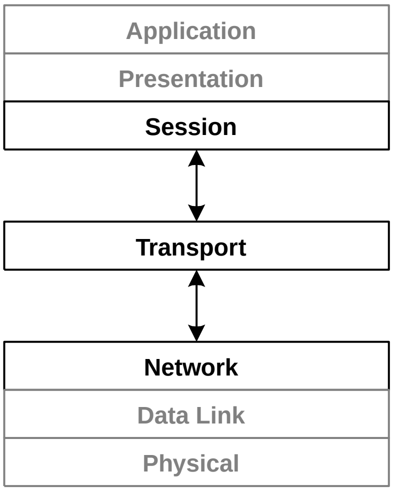

FIGURE 1–3 Each layer knows only about the layer above and below

Finally, the flow of data starts at the Application layer of the sending computer, flows down the layers, across the wire to the receiving computer, and then back up the layers to the Application layer (see Figure 1–4).

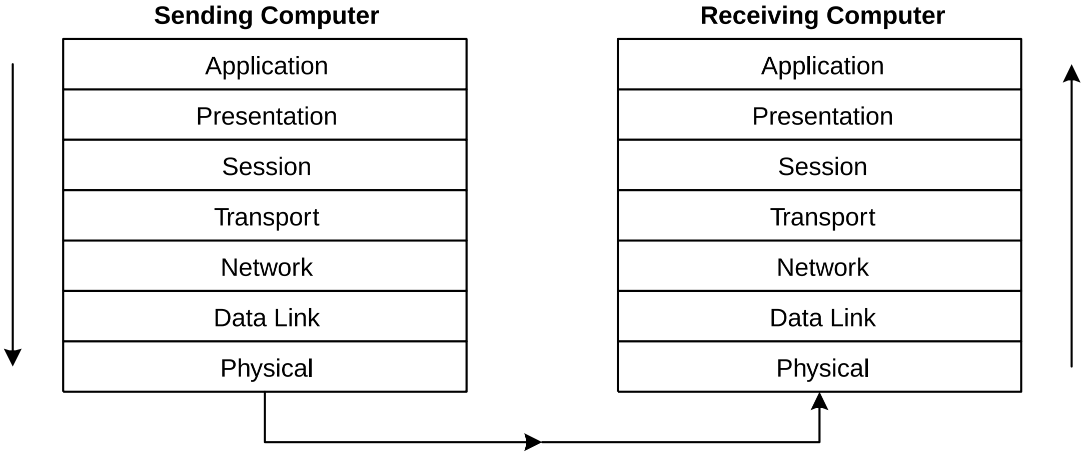

FIGURE 1–4 Flow of data from sending computer to receiving computer

Each layer has a specific function for which it is responsible. Although the layers start at the bottom, we will examine the layers starting at the top.

Application Layer (Layer 7)

The Application layer is a buffer between the user interface (what the user uses to perform work) and the network application. The network application may be a part of the user application or an *Application Programming Interface (API)* that is called by the user's application.

The Application layer is responsible for finding a communication partner on the network. Once a partner is found, it is then responsible for ensuring that there is sufficient network bandwidth to deliver the data.

This layer may also be responsible for synchronizing communication and providing error checking between the two partners. This ensures that the application is either sending or receiving, but not both, and that the data transmitted is the same data received.

Typical applications include a client/server application, an e-mail application, and an application to transfer files using FTP or HTTP.

Presentation Layer (Layer 6)

The Presentation layer is responsible for the presentation of data to the Application layer. This presentation may take the form of many structures.

For example, when communicating from a PC to a mainframe, data may need to be converted between ASCII and EBCDIC (a different character formatting method used on many mainframes).

Another structure of data includes multimedia formats used for enhancing our computer experiences. The World Wide Web (WWW) is a fantastic exchange of information that uses many types of multimedia. The Presentation layer must ensure that the application can view the appropriate data when it is reassembled. Graphic files such as PICT, JPEG, TIFF, and GIF, and video and sound files such as MPEG and Apple's QuickTime are examples of Presentation layer responsibilities.

One final data structure is data encryption. Sometimes, it is vital that we can send data across a network without someone being able to view our data, or *snoop* it. *Data encryption* is the method that allows us to accomplish this.

Session Layer (Layer 5)

The Session layer sets up communications between the two partners. This layer decides on the method of communication: half-duplex or *full-duplex*. Half-duplex is the method of sending data only when the other computer is finished. We can use the telephone to illustrate these methods. As a polite person (note the word polite), we wait until the other person is finished speaking before we respond. This is an example of half-duplex. If we were to both start speaking at the same time, it would take a trained ear to actually listen to the conversation. This is an example of full-duplex, where both sides communicate as fast as they are able, at the same time.

The Session layer starts a session by establishing the initial connection to the communication partner. This initial dialog allows the partners to decide on the communication method and the protocol to use. When this is finished, data transfer can occur between the partners. Finally, after all data has been transferred, the partners disconnect. Using our example of the telephone, this would be similar to me calling you. The initial "Hello" establishes the protocol (roughly equivalent to speaking English versus Spanish) and the method of communication, half-duplex. After we discuss the new products that Cisco is introducing and how they will make our lives better (the data transfer), we then conclude our conversation by saying "Good-bye."

In the early days of communication, reliability of data across the network was a major concern. With today's technology, this is less of a concern, but the Session layer has some error checking included. In order to ensure that the data has been transferred correctly, the Session layer sets up a checkpoint in the data stream. This checkpoint is a means for the receiving computer to acknowledge to the sender that the data has been received. If data is missed, the checkpoint acknowledgment back to the sender would indicate that data is missing, and the sender would then decide what data was missed, and resend it.

When the sender and receiver negotiate the use of checkpoints in the data stream, this is called connection-oriented service. We use this method when reliable delivery of data is required. However, there are times when reliable data is not needed, so to reduce network bandwidth, the checkpoints will not be used. The sender sends the data through the network and does not wait for any acknowledgments from the receiver. It is the responsibility of the Application layer to decide if data is missing. Although less reliable, it is a quicker method of sending data.

Examples of the Session layer protocol include:

- SQL (Structured Query Language). A database language originally developed by IBM.
- RPC (Remote Procedure Call). A method of running routines on a server called from the client.
- X Windows. A graphical-based system used to communicate with Unix servers.
- NFS (Network File System). A method of accessing resources on servers.

Transport Layer (Layer 4)

Although the Session layer is responsible for deciding on the communication method, the Transport layer implements it. This layer implements the functions necessary to send data to the communication partner. These mechanisms include multiplexing data from different applications, establishing data integrity, and management of virtual circuits.

Multiplexing is the method of combining data from the upper layers and sending them through the same data stream. This allows more than one application to communicate with the communication partner at the same time. When the data reaches the remote partner, the Transport layer then disassembles the segment and passes the correct data to each of the receiving applications.

Virtual circuits are the methods of setting up a communication path to the receiver. This path may physically change depending on the network, but the path remains open through a virtual link. The Transport layer is responsible for establishing, maintaining, and disconnecting the virtual circuits.

Data integrity is essential to passing data across a network. There are three methods that the Transport layer can use in order to ensure the integrity: buffering, source quench, and windowing. These three methods are implementations of flow control.

Buffering is maintained on the receiving computer. As data flows in faster than can be processed, some data is placed in a buffer and held until the computer has the time to process it. Unfortunately, if the speed of the data flow is too fast, the buffer will overflow and data will be lost.

Source quench is a technique where the receiving computer can send a control message back to the sending computer when too much data is being received. The sending computer then will delay sending any more data until the receiving computer can finish processing the current data. At this point, the receiving computer will send another control message back to the sender telling it to start sending data again.

The last method, *windowing*, works on the principle that the receiver tells the sender how much data it can send at one time. This amount sets the window size. After the receiver receives the data, it will send back an acknowledgment to the sender. This acknowledgment tells the sender which data segments have been received. The sender will remove the data from the window and fill it with new data. This process in sometimes referred to as a sliding window. Any data that has not been received will be resent. In fact, the sender has a timer on each segment that it sends, so it knows if the data has not been received. After the timer expires, it will resend the data and wait for the acknowledgment. This method is known as *Positive Acknowledgment with Retransmission*.

To illustrate window control, let's look at Figure 1–5.

1. The Sender sends a message asking to speak with the Receiver.
2. The Receiver decides on a window size and replies. The reply includes an acknowledgment to speak with the Sender and the size of the window to use; in this case, four bytes.
3. The Sender then creates a 4-byte window (shown in gray) and places the first data into it. The data *DATA* is sent across the network. The Receiver only receives the first two bytes, or *DA*, and sends back a positive acknowledgment. This packet tells the Sender that it has received bytes 1 and 2.
4. The Sender then slides the window to start with the third and fourth bytes and adds two more bytes to the free space. The data *TA12* is sent across the network. The Receiver again sends back a positive acknowledgment to the Sender that the third, fourth, fifth, and sixth bytes have been received. Both partners will now slide the window four more bytes, and the Sender will send the seventh through tenth bytes. This continues until all data has been sent and the two partners agree to stop communications.

Note that this is a simple example and more communication may be taking place between the Sender and the Receiver.

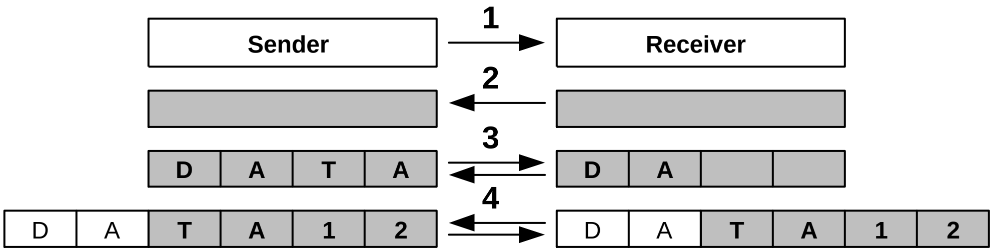

FIGURE 1–5 Sliding window example

Network Layer (Layer 3)

The bottom two layers allow the communication partners to communicate only if they are on the same segment. A segment is defined as all network devices, or nodes, that are directly connected together. Hubs and MAUs (Multistation Access Units) are part of the same segment. If we were to have all the computers in the world on a single segment, there would be no Internet. Either there would be far too many broadcasts, or the time for a single node to communicate with another via a token would be ridiculously slow. The function of the Network layer is to identify a remote network and deliver the data to it. This allows us to have segmentation. These concepts are explored in more detail in later chapters.

The Network layer enables us to send data to any computer in the world, as long as there is a physical network connection. However, since we can't have a single segment, we must divide these segments up and yet keep them communicating. The device that allows us to accomplish this spectacular feat is the router, sometimes referred to as a Layer 3 device. The majority of this book is centered on this specific layer.

A router may know more than one way to get data to its final destination. Again, this is the function of the Network layer. In order for the router to succeed in this endeavor, it must be able to identify the source segment and the final destination segment. This is done through network addresses, also called logical addresses.

A network address consists of two parts, the network portion and the host portion. As a simple example, suppose the number 1.2 was assigned to your specific machine. The 1 would identify the network segment and the 2 would identify you as a specific host on that network. Another computer on your segment would have an address of 1.x, where x would be a unique number. A computer on a different segment might have an address of 2.2, or 3.49. To connect these two segments together, we would need to place a router.

It is important to note that routers work only with the network address. They really don't want to know your specific host address. When a router receives data, it examines the Layer 3 data to determine the destination network address. It then looks up the address in a table that tells it which route to use to get the data to its final destination. It places the data on the proper connection, thereby routing the packet from one segment to another. The data may need to travel through many routers before reaching its destination host. Each router in the path would perform the same lookup in its table.

Examples of network protocols include:

- IP (Internet Protocol). A routed protocol used in the TCP/IP suite, made famous by the Internet.
- IPX (Internet Packet eXchange). A routed protocol used in the IPX/SPX protocol suite usually used in Novell environments.
- RIP (Routing Information Protocol). One of the many routing protocols implemented on Cisco routers.
- OSPF (Open Shortest Path First). Another routing protocol used by Cisco routers.

We discuss the difference between routed protocols and routing protocols later in this book.

Data Link Layer (Layer 2)

The Data Link layer is the layer connects the software protocols to the hardware protocols. It is responsible for taking the data from the upper layers and converting it to the bits needed to send across the physical wire, and vice versa.

The Data Link layer is split into two sublayers, the Logical Link Control (LLC) and the Media Access Control (MAC). As you can see in Figure 1–6, the MAC sublayer is closer to the Physical layer.

The MAC sublayer defines a physical address, called a MAC address or hardware address, which is unique to each individual network interface. This allows a way to uniquely identify each network interface on a network, even if the network interfaces are on the same computer. More importantly, though, the MAC address can be used in any network that supports the chosen network interface (Ethernet or Token Ring, for example). This allows us to take a computer from one network running TCP/IP and connect it to another network running IPX/SPX by changing just the Network layer protocol. Remember that the network address, or logical address, is specific to the Network layer protocol that is being used. This address may be a unique numbering scheme, as in TCP/IP, or it may be a copy of the MAC address, as in IPX/SPX.

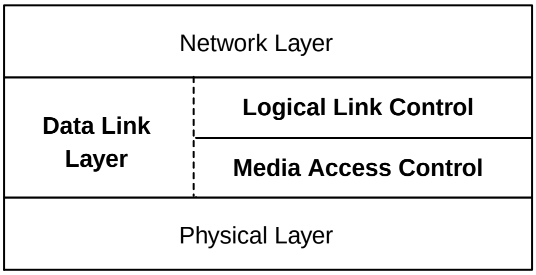

FIGURE 1–6 Split layers of the Data Link layer

A MAC address is a 6-byte value that is usually created by the network interface manufacturer. The first three bytes are assigned by the IEEE (Institute of Electrical and Electronics Engineers) and are specific to each vendor. The vendor generates the last three bytes. Examples are shown here where X can be any hex value from 0 to F.

• 00-00-0C-XX-XX-XX	Cisco
• 00-E0-98-XX-XX-XX	LinkSys
• 00-10-5A-XX-XX-XX	3Com

The MAC layer on the receiving computer will take the bits from the Physical layer and put them in order into a frame. It will also do a CRC (Cyclic Redundancy Check) to determine if there are any errors in the frame. It will check the destination hardware address to determine if the data is meant for it, or if it should be dropped or sent on to the next machine. If the data is meant for the current computer, it will pass it to the LLC layer.

The MAC layer can be referred to as the hardware layer. This implies that the software protocols above it are hidden from the physical media.

The LLC layer is the buffer between the software protocols and the hardware protocols. It is responsible for taking the data from the Network layer and sending it to the MAC layer. This allows the software protocols to run on any type of network architecture, including Ethernet and Token Ring.

When the LLC receives data from the MAC layer, it must determine which software protocol in the Network layer to send it to. In order to do this, the LLC includes service access points (SAP) in the header. The Source SAP (SSAP) identifies the sending protocol and the Destination SAP (DSAP) identifies the receiving protocol. When the LLC receives the frame from the MAC sublayer, it can then strip off the header and examine the DSAP. Using this information, the LLC can now forward the data to the correct Network layer protocol.

Examples of Data Link protocols include:

- HDLC (High-Level Data Link Control). A serial communication that is usually vendor specific.
- PPP (Point-to-Point Protocol). A low-speed serial protocol.
- 802.3 and 802.2
- ISDN (Integrated Services Digital Network). A digital communication method used over copper wire.
- Frame Relay

Physical Layer (Layer 1)

The Physical layer does only two things, yet these two things are vital to the network. It is responsible for sending data and receiving data across a physical medium. This data is sent in bits, either a 0 or a 1. The data may be transmitted as electrical signals (that is, positive and negative voltages), audio tones, or light.

This layer also defines the Data Terminal Equipment (DTE) and the Data Circuit-Terminating Equipment (DCE). The DTE is often accessed through a modem or a Channel Service Unit/Data Service Unit (CSU/DSU) connected to a PC or a router. The carrier of the WAN signal provides the DCE equipment. A typical device would be a packet switch, which is responsible for clocking and switching.

Typical interfaces of this layer include:

- HSSI (High-Speed Serial Interface). A point-to-point connection over copper wires.
- V.35. A synchronous communication method developed by the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T). All the V.xx standards have been created by the ITU-T.
- EIA/TIA-232. A serial port interface using the RS-232 port.
- X.21

Data Encapsulation Using the OSI Model

As we read through the description of the OSI layers, a question may arise: Since there may be more than one application using more than one communication partner using more than one protocol, how does the data get to its destination correctly?

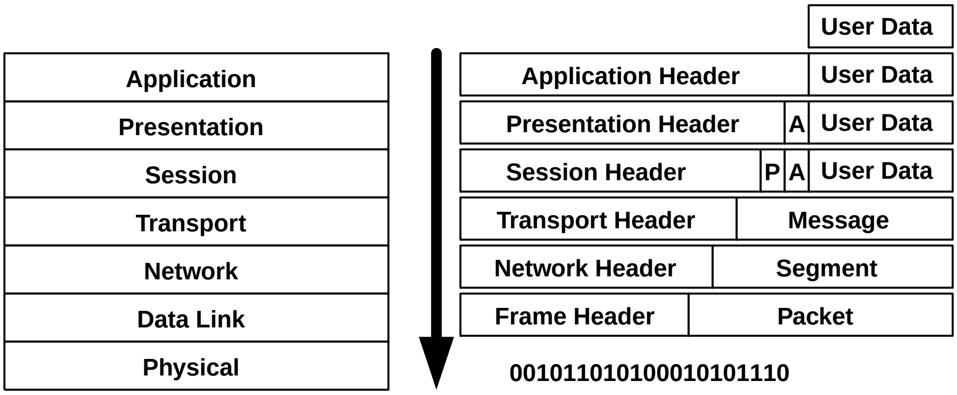

FIGURE 1–7 Data encapsulation flow

This is accomplished through a process called *data encapsulation* (see Figure 1–7). Basically, it works like this:

1. A user is working on an application and decides to save the data to a remote server. The application calls the Application layer to start the process.
2. The Application layer takes the data and places some information, called a header, at the beginning. The header tells the Application layer which user application sent the data.
3. The Application layer then sends the data to the Presentation layer, where the data conversion takes place. The Presentation layer places a header on all of the information received from the Application layer (including the Application layer header). This header identifies which protocol in the Application layer to pass it back.
4. The Presentation layer then sends the complete message to the Session layer. The Session layer sets up the synchronized communication information to speak with the communication partner and appends the information to another header.
5. The Session layer then sends the message to the Transport layer, where information is placed into the header identifying the source and the destination hosts and the method of connection (connectionless versus connection-oriented).

6. The Transport layer then passes the segment to the Network layer, where the network address for the destination and the source are included in the header.
7. The Network layer passes the packet (connection-oriented) or the datagram (connectionless) to the Data Link layer. The Data Link layer then includes the SSAP and the DSAP to identify which Transport protocol to return it to. It also includes the source and the destination MAC addresses.
8. The Data Link layer then passes the frame to the Physical layer for transmitting on the physical medium as individual bits.
9. Finally, the receiving computer receives the bits and reverses the process to get the original data to the source application; in this case, a file server service.

Note that since the top three layers have similar functionality, we can typically combine all of the data in those layers and simply refer to it as the Protocol Data Unit (PDU). In this Instance, we can substitute the term *PDU* for the term *message.*

In summary:

1. Data encapsulation takes the data from the user,
2. packages it as a message at the Session layer to send to the receiver,
3. encapsulates the segment inside a packet at the Network layer with the network addressing information,
4. encapsulates this packet into a frame at the Data Link layer with the MAC addresses,and
5. sends the frame across the wire as individual bits at the Physical layer.

Summary

The OSI model is a tool used to provide a standard set of rules for communication across multivendor hardware. Each layer of the OSI model has a function it must accomplish, but the method to perform this function is left to the developer. As data flows down each of the layers, encapsulation takes place allowing for reassembly of the data on the receiving machine. Each layer is responsible in part for the complete data transfer from the source host to the destination host. Knowing the OSI model will help in troubleshooting Cisco networks.

Scenario Lab 1.1

You have been asked to begin a large design and implementation project with Network Solutions, Inc. (a fictional company). However, before you are allowed to proceed, the project manager wants to be sure that you are the correct person for the job. This person asks you to list the layers of the OSI model and describe in brief the functions of each layer. Can you prove you are the correct person for the job?

Exam Objective Checklist

By working through this chapter, you should have sufficient knowledge to answer these exam objectives:

- Identify and describe the functions of each of the seven layers of the OSI reference model.
- Describe data link addresses and network addresses and identify the key differences between them.
- Define and describe the function of a MAC address.
- Identify at least three reasons why the industry uses a layered model.
- Define and explain the five conversion steps of data encapsulation.
- Define flow control and describe the three basic methods used in networking.
- List the key internetworking functions of the OSI Network layer and how they are performed in a router.

Practice Questions

1. The technology used to connect multiple computers together in a single office is called:

a. LAN
b. WAN
c. MAN
d. Internet

2. Connecting multiple networks together using an outside carrier's signal, such as the telephone service is known as:

a. LAN
b. WAN
c. Protocol
d. Internet

3. Why should we use layered models in a network architecture? (select all that apply)

a. It tells us exactly how to perform a specific function.
b. It allows us to take a complex method and break it into smaller, more manageable methods.
c. A change to one layer has no affect on any other layer.
d. A change to one layer affects all other layers.
e. It restricts us to using only one network vendor.
f. It makes troubleshooting networks easier by being able to locate the exact layer causing the problem.

4. Which layer is responsible for finding a communication partner on the network?

- **a.** Transport
- **b.** Data Link
- **c.** Application
- **d.** Physical

5. What is the correct order for the shown layers? (bottom to top)

- **a.** Presentation
- **b.** Transport
- **c.** Application
- **d.** Network
- **e.** Data Link
- **f.** Physical
- **g.** Session

6. True or False: The Transport layer can communicate directly with the Network and Presentation layers.

7. Which of the following are performed at the Presentation layer? (Choose two)

- **a.** Presening data to the Application layer
- **b.** Setting checkpoints in the data stream for reliability
- **c.** Providing character conversion between dissimilar operating systems (such as PC to mainframe)
- **d.** Adding the network addresses to the header

8. The Presentation layer protocols include? (choose two)

- **a.** PICT
- **b.** SQL
- **c.** TCP
- **d.** IPX
- **e.** JPEG

9. The function of the Session layer is? (choose two)

- **a.** To determine if half-duplex or full-duplex is being used
- **b.** To present data to the Network layer
- **c.** To place checkpoints into the data stream for reliability
- **d.** To provide flow control

10. The Session layer protocols include? (choose three)

- **a.** PICT
- **b.** SQL
- **c.** TCP
- **d.** X Windows
- **e.** NFS

11. Which layer is responsible for multiplexing data from upper layers and placing the data into a segment?

a. Transport
b. Network
c. Data Link
d. Physical

12. Windowing is performed at the Transport layer. What is windowing?

a. A method of buffering
b. A method of session establishment
c. A method of flow control
d. A method of character conversion

13. The Network layer's primary function is to:

a. Add MAC addresses to the packet
b. Establish a communication path to the communication partner
c. Provide connection-oriented service
d. Route data between different network segments

14. What are the two parts to a network address? (Choose two)

a. Source Service Access Point
b. Host Identifier
c. MAC address
d. Network Identifier

15. The Data Link layer is split into two sublayers. Name them. (choose two)

a. Local Link Control
b. Logical Link Control
c. Machine Address Code
d. Media Access Control

16. List the functions of the MAC sublayer. (choose three)

a. Unique hardware addresses allow us to switch between different networks and still be uniquely identified
b. Provides SSAP and DSAP for passing frame to proper Transport protocol
c. Provides error checking through CRC
d. Provides an interface to the physical medium
e. Acts as a buffer between software and hardware protocols

17. List the functions of the LLC sublayer. (Choose two)

a. Unique hardware addresses allow us to switch between different networks and still be uniquely identified
b. Provides SSAP and DSAP for passing frames to the proper Network protocol

- **c.** Provides error checking through CRC
- **d.** Provides an interface to the physical medium
- **e.** Acts as a buffer between software and hardware protocols

18. Which layer is responsible for creating and disconnecting virtual circuits?

- **a.** Presentation
- **b.** Session
- **c.** Transport
- **d.** Network

19. Which of the following terms describes the address used at the Network layer?

- **a.** Physical
- **b.** Logical
- **c.** MAC
- **d.** Host

20. Place the following in the correct order of data encapsulation for the sending node.

- **a.** Encapsulating this packet or datagram into a frame with the MAC addresses
- **b.** Packaging it as a message to send to the receiver
- **c.** Sending the frame across the wire as individual bits
- **d.** Data encapsulation taking the data from the user
- **e.** Encapsulating the segment inside a packet or datagram with the network addressing information

21. If your network diagnostic tool identifies a problem with the logical addressing, what layer of the OSI model would you be troubleshooting?

- **a.** Transport
- **b.** Network
- **c.** Data Link
- **d.** Physical

22. What layer would you troubleshoot when no link connectivity is detected?

- **a.** Transport
- **b.** Session
- **c.** Network
- **d.** Physical

23. If two network cards were suspected of having the same MAC address, what layer would you troubleshoot to determine the conflict?

- **a.** Network
- **b.** Data Link
- **c.** Physical
- **d.** Transport

24. One of your nodes requests a window size of 1. This is having adverse effects on the network and you need to change it. What layer of the OSI model is responsible for this?

a. Transport
b. Network
c. Data Link
d. Physical

25. Identify the MAC address:

a. 00230405
b. 00-23-04-05
c. f4e3d2c1b0
d. f4-e3-d2-c1-b0-a9