

chapter 2

Building a Network with the OSI Model

"First we thought the PC was a calculator. Then we found out how to turn numbers into letters with ASCII—and we thought it was a typewriter. Then we discovered graphics, and we thought it was a television. With the World Wide Web, we've realized it's a brochure."

—DOUGLAS ADAMS



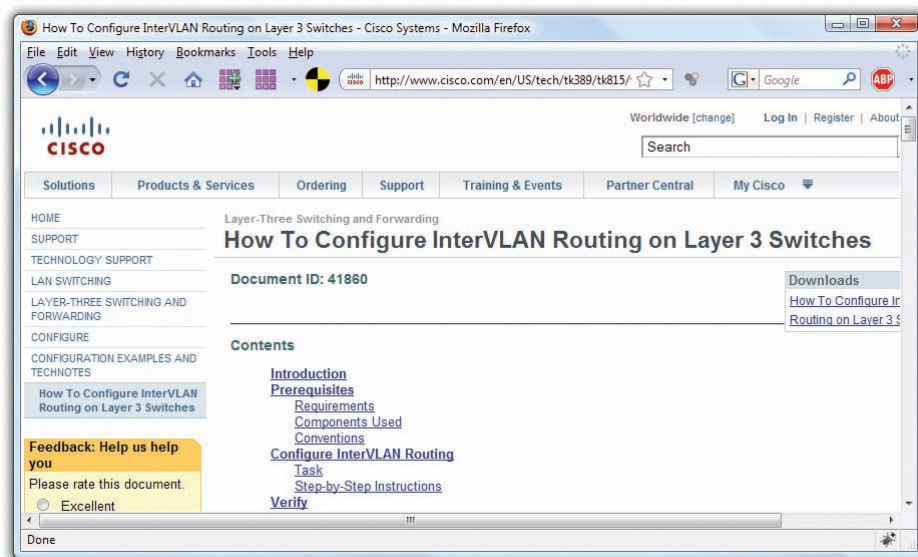
In this chapter, you will learn how to

- Describe models such as the OSI seven-layer model
- Explain the major functions of network hardware with OSI Layers 1–2
- Describe the functions of network software with OSI Layers 3–7

The CompTIA Network+ certification challenges you to understand virtually every aspect of networking—not a small task. Luckily for you, there's a long-used method to conceptualize the many parts of a network called the Open Systems Interconnection (OSI) seven-layer model.

The **OSI seven-layer model** is a guideline, a template that breaks down how a network functions into seven parts called layers. If you want to get into networking—and if you want to pass the CompTIA Network+ certification exam—you must understand the OSI seven-layer model in great detail.

The OSI seven-layer model provides a practical model for networks. The model provides two things. For network techs, the OSI seven-layer model provides a powerful tool for diagnosing problems. Understanding the model enables a tech to determine quickly at what layer a problem can occur and thus zero in on a solution without wasting a lot of time on false leads. The model also provides a common language to describe networks—a way for us to communicate with each other about the functions of a network. Figure 2.1 shows a sample Cisco Systems Web page about configuring routing—a topic this book covers in detail later on. A router operates at Layer 3 of the OSI seven-layer model, for example, so you'll hear techs (and Web sites) refer to it as a "Layer 3 switch." That's a use of the OSI seven-layer model as language.



• **Figure 2.1** Using the OSI terminology—Layer 3—in a typical setup screen

This chapter looks first at models, and specifically at the OSI seven-layer model to see how it helps make network architecture clear for techs. The second and third portions of the chapter apply that model to the practical pieces of networks, the hardware and software common to all networks.



Cross Check

Cisco and Certifications

You learned a little about Cisco, the major player in Internet hardware, in Chapter 1, “CompTIA Network+ in a Nutshell,” so check your knowledge. What does Cisco say about CompTIA Network+ certification? Where would you go to get more details?

Historical/Conceptual

■ Working with Models

The best way to learn the OSI seven-layer model is to see it in action. For this reason, I’ll introduce you to a small network that needs to copy a file from one computer to another. This example goes through each of the OSI layers needed to copy that file, taking time to explain each step and why it is necessary. By the end of the chapter you should have a definite handle on using the OSI seven-layer model as a way to conceptualize networks. You’ll continue to build on this knowledge throughout the book and turn it into a powerful troubleshooting tool.

Biography of a Model

What does the word “model” mean to you? Does the word make you think of a beautiful woman walking down a catwalk at a fashion show or some

hunky guy showing off the latest style of blue jeans on a huge billboard? Maybe it makes you think of a plastic model airplane? What about those computer models that try to predict weather? We use the term “model” in a number of ways, but each use shares certain common themes.



• **Figure 2.2** Types of models (images from left to right courtesy of NOAA, Mike Schinkel, and Albert Poawui)

All models are a simplified representation of the real thing. The human model ignores the many different types of body shapes, using only a single “optimal” figure. The model airplane lacks functional engines or the internal framework, and the computerized weather model might disregard subtle differences in wind temperatures or geology (Figure 2.2).

Additionally, a model must have at least all the major functions of the real item, but what constitutes a major

rather than a minor function is open to opinion. Figure 2.3 shows a different level of detail for a model. Does it contain all the major components of an airplane? There’s room for argument that perhaps it should have landing gear to go along with the propeller, wings, and tail.

In modeling networks, the OSI seven-layer model faces similar challenges. What functions define all networks? What details can be omitted and yet not render the model inaccurate? Does the model retain its usefulness when describing a network that does not employ all the layers?

In the early days of networking, different manufacturers made unique types of networks that functioned fairly well. But each network had its own cabling, hardware, drivers, naming conventions, and many other unique features. In fact, most commonly, a single manufacturer would provide everything for a customer: cabling, NICs, hubs, and drivers, even all the software, in one complete and expensive package!

Although these networks worked fine as stand-alone networks, the proprietary nature of the hardware and software made it difficult—to put it mildly—to connect networks of multiple manufacturers. To interconnect networks and improve networking as a whole, someone needed to create a guide, a model that described the functions of a network, so that people who made hardware and software could work together to make networks that worked together well.

The International Organization for Standardization, known as ISO, proposed the OSI seven-layer model. The OSI seven-layer model provides precise terminology for discussing networks—so let’s see it!



• **Figure 2.3** Simple model airplane



ISO may look like a misspelled acronym, but it’s actually a word, derived from the Greek word *isos*, which means equal.

The Seven Layers in Action

Each layer in the OSI seven-layer model defines a challenge in computer networking, and the protocols that operate at that layer offer solutions to those challenges. **Protocols** define rules, regulations, standards, and procedures so that hardware and software developers can make devices and applications that function properly. The OSI model encourages modular design in networking, meaning that each protocol is designed to deal with a specific layer and to have as little to do with the operation of other layers as possible. Each protocol needs to understand the protocols handling the layers directly above and below it, but it can, and should, be oblivious to the protocols handling the other layers.

The seven layers are

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

The best way to understand OSI is to see it in action—let’s see it work at the fictional company of MHTechEd, Inc.

Welcome to MHTechEd!

Mike’s High-Tech Educational Supply Store and Post Office, or MHTechEd for short, has a small network of PCs running Windows, a situation typical of many small businesses today. Windows runs just fine on a PC unconnected to a network, but it also comes with all the network software it needs to connect to a network. All the computers in the MHTechEd network are connected by special network cabling.

As in most offices, virtually everyone at MHTechEd has his or her own PC. Figure 2.4 shows two workers, Janelle and Tiffany, who handle all the administrative functions at MHTechEd. Because of the kinds of work they do, these two often need to exchange data between their two PCs. At the moment, Janelle has just completed a new employee handbook in Microsoft Word, and she wants Tiffany to check it for accuracy. Janelle could transfer a copy of the file to Tiffany’s computer by the tried-and-true sneakernet method, saving the file on a thumb drive and walking it over to her, but thanks to the wonders of computer networking, she doesn’t even have to turn around in her chair. Let’s watch in detail each piece of the process that gives Tiffany direct access to Janelle’s computer, so she can copy the Word document from Janelle’s system to her own.



Be sure to memorize both the name and the number of each OSI layer. Network techs use terms such as “Layer 4” and “Transport layer” synonymously. Students have long used mnemonics for memorizing such lists. One of my favorites for the OSI seven-layer model is Please Do Not Throw Sausage Pizza Away. Yum!



Keep in mind that these layers are not laws of physics—anybody who wants to design a network can do it any way he or she wants. While many protocols fit neatly into one of the seven layers, others do not.



This section is a conceptual overview of the hardware and software functions of a network. Your network may have different hardware or software, but it will share the same functions!



• **Figure 2.4** Janelle and Tiffany, hard at work

Long before Janelle ever saved the Word document on her system—when the systems were first installed—someone who knew what they were doing set up and configured all the systems at MHTechEd to be part of a common network. All this setup activity resulted in multiple layers of hardware and software that can work together behind the scenes to get that Word document from Janelle’s system to Tiffany’s. Let’s examine the different pieces of the network, and then return to the process of Tiffany grabbing that Word document.

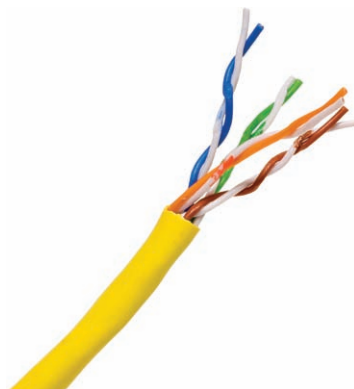
Test Specific

■ Let’s Get Physical—Network Hardware and Layers 1–2

Clearly the network needs a physical channel through which it can move bits of data between systems. Most networks use a cable like the one shown in Figure 2.5. This cable, known in the networking industry as **unshielded twisted pair (UTP)**, usually contains four pairs of wires that transmit data.

Another key piece of hardware the network uses is a special box-like device called a **hub** (Figure 2.6), often tucked away in a closet or an equipment room. Each system on the network has its own cable that runs to the hub. Think of the hub as being like one of those old-time telephone switchboards, where operators created connections between persons who called in wanting to reach other telephone users.

Layer 1 of the OSI model defines the method of moving data between computers. So the cabling and hubs are part of the **Physical layer** (Layer 1). Anything that moves data from one system to another, such as copper cabling, fiber optics, even radio waves, is part of the Physical layer. Layer 1 doesn’t care what data goes through; it just moves the data from one system to another system. Figure 2.7 shows the MHTechEd network in the OSI seven-layer model thus far. Note that each system has the full range of layers, so data from Janelle’s computer can flow to Tiffany’s computer.



• Figure 2.5 UTP cabling



• Figure 2.6 Typical hub

The real magic of a network starts with the **network interface card**, or **NIC** (pronounced “nick”), which serves as the interface between the PC and the network. While NICs come in a wide array of shapes and sizes, the ones at MHTechEd look like Figure 2.8.

On older systems, a NIC truly was a separate card that snapped into a handy expansion port, which is why they were called network interface *cards*. Even though they’re now built into the motherboard, we still call them NICs.

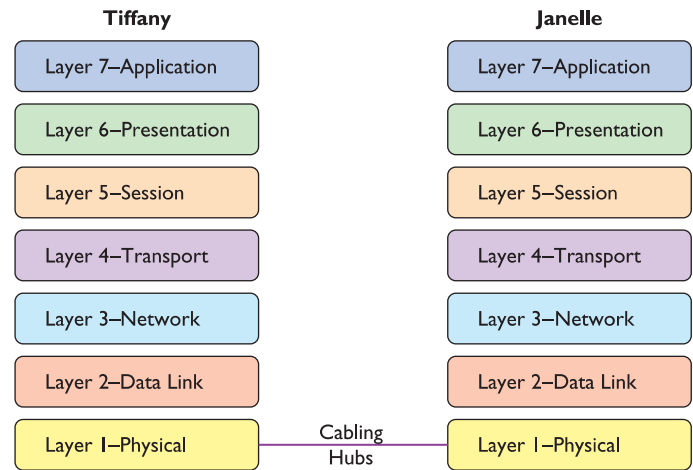
When installed in a PC, the NIC looks like Figure 2.9. Note the cable running from the back of the NIC into the wall; inside that wall is another cable running all the way back to the hub.

Cabling and hubs define the Physical layer of the network, and NICs provide the interface to the PC. Figure 2.10 shows a diagram of the network cabling system. I’ll build on this diagram as I delve deeper into the network process.

You might be tempted to categorize the NIC as part of the Physical layer at this point, and you’d have a valid argument. The NIC clearly is necessary for the physical connection to take place! The CompTIA Network+ exam and most authors put the NIC into Layer 2, the Data Link layer, though, so clearly something else is happening inside the NIC. Let’s take a closer look.

The NIC

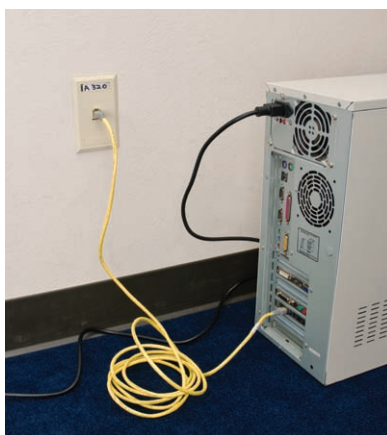
To understand networks, you must understand how NICs work. The network must provide a mechanism that gives each system a unique identifier—like a telephone number—so that data is delivered to the right system. That’s one of the most important jobs of a NIC. Inside every NIC, burned onto some type of ROM chip, is special firmware containing a



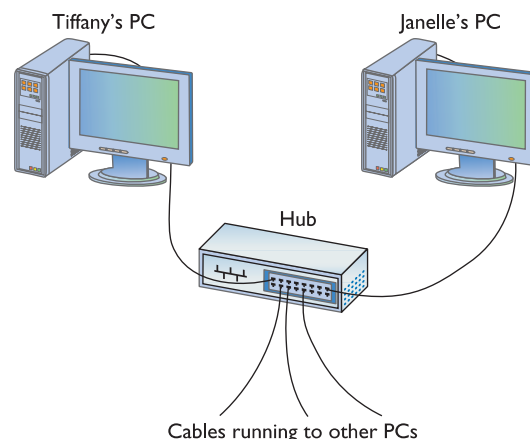
• **Figure 2.7** The network so far, with the Physical layer hardware installed



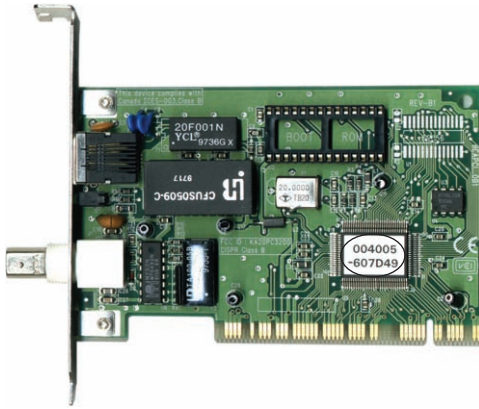
• **Figure 2.8** Typical NIC



• **Figure 2.9** NIC with cable connecting the PC to the wall jack



• **Figure 2.10** The MHTechEd network



• Figure 2.11 MAC address

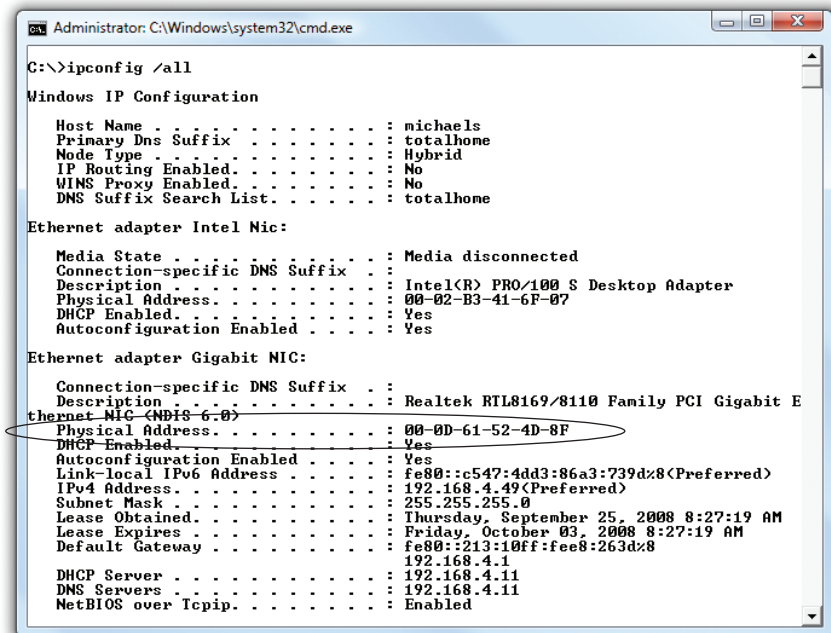
unique identifier with a 48-bit value called the *media access control address*, or **MAC address**.

No two NICs ever share the same MAC address—ever. Any company that makes NICs must contact the Institute of Electrical and Electronics Engineers (IEEE) and request a block of MAC addresses, which the company then burns into the ROMs on its NICs. Many NIC makers also print the MAC address on the surface of each NIC, as shown in Figure 2.11. Note that the NIC shown here displays the MAC address in hexadecimal notation. Count the number of hex characters—because each hex character represents 4 bits, it takes 12 hex characters to represent 48 bits.

The MAC address in Figure 2.11 is 004005-607D49, although in print, we represent the MAC as 00-40-05-60-7D-49. The first six digits, in this example 00-40-05, represent the number of the manufacturer of the NIC. Once the IEEE issues to a manufacturer those six hex digits—often referred to as the **organizationally unique identifier (OUI)**—no other manufacturer may use them. The last six digits, in this example 60-7D-49, are the manufacturer’s unique serial number for that NIC; this portion of the MAC is often referred to as the **device ID**.

Would you like to see the MAC address for your NIC? If you have a Windows system, type **IPCONFIG /ALL** from a command prompt to display the MAC address (Figure 2.12). Note that IPCONFIG calls the MAC address the **physical address**, which is an important distinction, as you’ll see a bit later in the chapter.

Okay, so every NIC in the world has a unique MAC address, but how is it used? Ah, that’s where the fun begins! Recall that computer data is binary, which means it’s made up of streams of ones and zeroes. NICs send and receive this binary data as pulses of electricity, light, or radio waves. The NICs that use electricity to send and receive data are the most common, so let’s



• Figure 2.12 Output from IPCONFIG /ALL

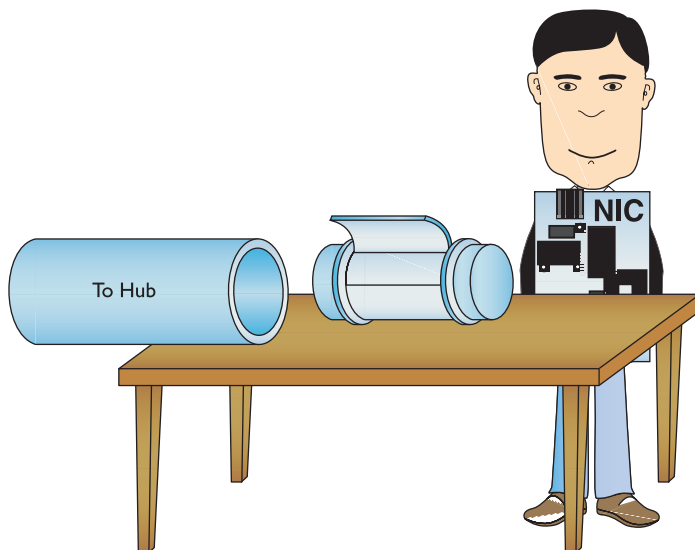
consider that type of NIC. The specific process by which a NIC uses electricity to send and receive data is exceedingly complicated, but luckily for you, not necessary to understand. Instead, just think of a *charge* on the wire as a *one*, and *no charge* as a *zero*. A chunk of data moving in pulses across a wire might look something like Figure 2.13.

If you put an oscilloscope on the wire to measure voltage, you'd see something like Figure 2.14. An oscilloscope is a powerful microscope that enables you to see electrical pulses.

Now, remembering that the pulses represent binary data, visualize instead a string of ones and zeroes moving across the wire (Figure 2.15).

Once you understand how data moves along the wire, the next question becomes this: how does the network get the right data to the right system? All networks transmit data by breaking whatever is moving across the physical layer (files, print jobs, Web pages, and so forth) into discrete chunks called frames. A **frame** is basically a container for a chunk of data moving across a network. The NIC creates and sends, as well as receives and reads, these frames.

I like to visualize an imaginary table inside every NIC that acts as a frame creation and reading station. I see frames as those pneumatic canisters you see when you go to a drive-in teller at a bank. A little guy inside the network card—named Nick, naturally!—builds these pneumatic canisters (the frames) on the table, and then shoots them out on the wire to the hub (Figure 2.16).



• Figure 2.16 Inside the NIC

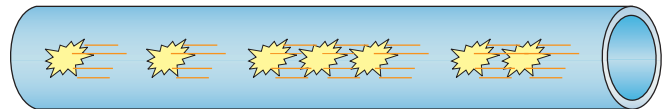


Try This!

What's Your MAC Address?

You can readily determine your MAC address on a Windows computer from the command line. This works in all modern versions of Windows.

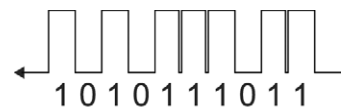
1. In Windows 2000/XP, click Start | Run. Enter the command **CMD** and press the ENTER key to get to a command prompt.
2. In Windows Vista, click Start, enter **CMD** in the Start Search text box, and press the ENTER key to get to a command prompt.
3. At the command prompt, type the command **IPCONFIG /ALL** and press the ENTER key.



• Figure 2.13 Data moving along a wire



• Figure 2.14 Oscilloscope of data




• Figure 2.15 Data as ones and zeroes



A number of different frame types are used in different networks. All NICs on the same network must use the same frame type or they will not be able to communicate with other NICs.

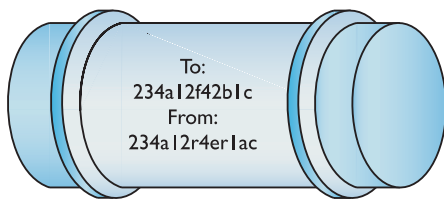
Recipient's MAC address	Sender's MAC address	Data	CRC
----------------------------	-------------------------	------	-----

• **Figure 2.17** Generic frame


Tech Tip

CRC in Depth

Most CRCs are only 4 bytes long, yet the average frame carries around 1500 bytes of data. How can 4 bytes tell you if all 1500 bytes in the data are correct? That's the magic of CRCs. Without going into the grinding details, think of the CRC as just the remainder of a division problem. (Remember learning remainders from division back in elementary school?) The NIC sending the frame does a little math to make the CRC. Using binary arithmetic, it works a division problem on the data using a divisor called a key. This key is the same on all the NICs in your network—it's built in at the factory. The result of this division is the CRC. When the frame gets to the receiving NIC, it divides the data by the same key. If the receiving NIC's answer is the same as the CRC, it knows the data is good.



• **Figure 2.18** Frame as a canister

Here's where the MAC address becomes important. Figure 2.17 shows a representation of a generic frame. Even though a frame is a string of ones and zeroes, we often draw frames as a series of rectangles, each rectangle representing a part of the string of ones and zeroes. You will see this type of frame representation used quite often, so you should become comfortable with it (even though I still prefer to see frames as pneumatic canisters!). Note that the frame begins with the MAC address of the NIC to which the data is to be sent, followed by the MAC address of the sending NIC. Then comes the data, followed by a special bit of checking information called the **cyclic redundancy check (CRC)** that the receiving NIC uses to verify that the data arrived intact.

So, what's inside the data part of the frame? We neither know nor care. The data may be a part of a file, a piece of a print job, or part of a Web page. NICs aren't concerned with content! The NIC simply takes whatever data is passed to it via its device driver and addresses it for the correct system. Special software will take care of *what* data gets sent and what happens to that data when it arrives. This is the beauty of imagining frames as little pneumatic canisters (Figure 2.18). A canister can carry anything from dirt to diamonds—the NIC doesn't care one bit (pardon the pun).

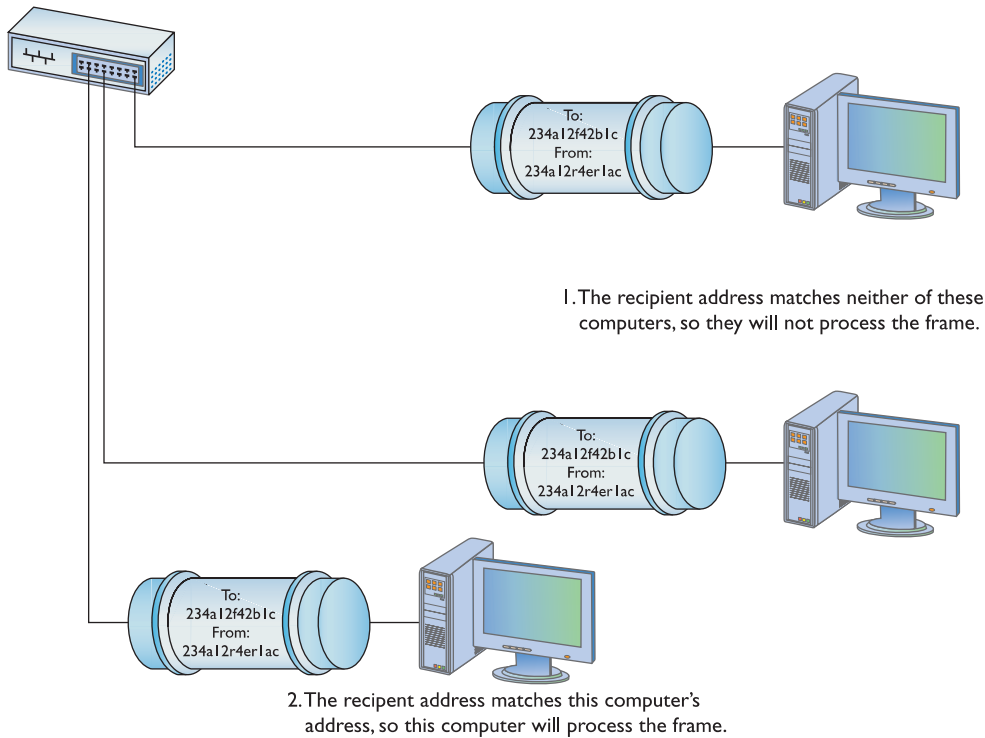
Like a canister, a frame can hold only a certain amount of data. Different networks use different sizes of frames, but generally, a single frame holds about 1500 bytes of data. This raises a new question: what happens when the data to be sent is larger than the frame size? Well, the sending system's software must chop the data up into nice, frame-sized chunks, which it then hands to the NIC for sending. As the receiving system begins to accept the incoming frames, it's up to the receiving system's software to recombine the data chunks as they come in from the network. I'll show how this disassembling and reassembling is done in a moment—first, let's see how the frames get to the right system!

When a system sends a frame out on the network, the frame goes into the hub. The hub, in turn, makes an exact copy of that frame, sending a copy of the original frame to every other system on the network. The interesting part of this process is when the copy of the frame comes into all the other systems. I like to visualize a frame sliding onto the receiving NIC's "frame assembly table," where the electronics of the NIC inspect it. Here's where the magic takes place: only the NIC to which the frame is addressed will process that frame—the other NICs simply erase it when they see that it is not addressed to their MAC address. This is important to appreciate: *every* frame sent on a network is received by *every* NIC, but only the NIC with the matching MAC address will process that particular frame (Figure 2.19).

When a system sends a frame out on the network, the frame goes into the hub. The hub, in turn, makes an exact copy of that frame, sending a copy of the original frame to every other system on the network. The interesting part of this process is when the copy of the frame comes into all the other systems. I like to visualize a frame sliding onto the receiving NIC's "frame assembly table," where the electronics of the NIC inspect it. Here's where the magic takes place: only the NIC to which the frame is addressed will process that frame—the other NICs simply erase it when they see that it is not addressed to their MAC address. This is important to appreciate: *every* frame sent on a network is received by *every* NIC, but only the NIC with the matching MAC address will process that particular frame (Figure 2.19).

Getting the Data on the Line

The process of getting data onto the wire and then picking that data off the wire is amazingly complicated. For instance, what happens to keep two NICs from speaking at the same time? Because all the data sent by one NIC is read by every other NIC on the network, only one system may speak at a time. Networks use frames to restrict the amount of data a NIC can send at



• **Figure 2.19** Incoming frame!

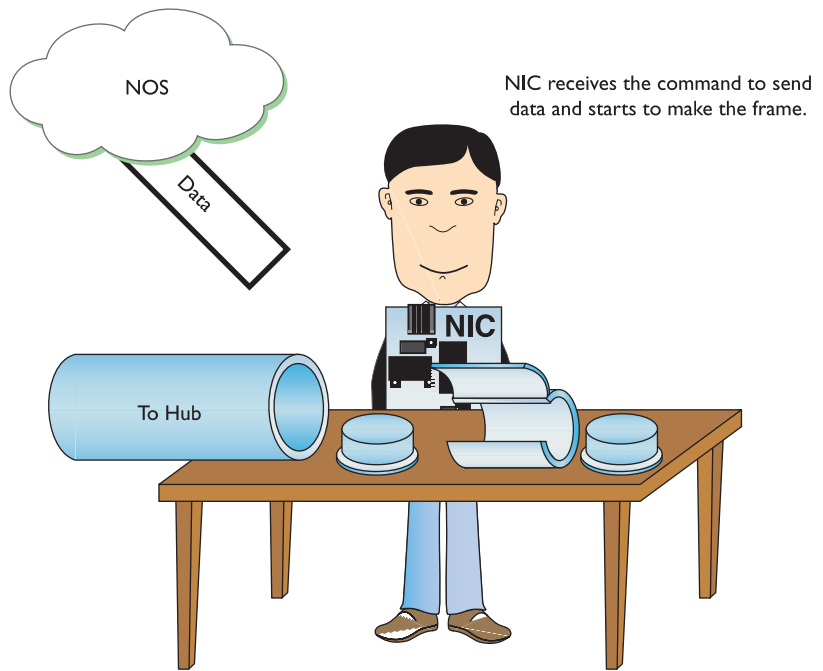
once, giving all NICs a chance to send data over the network in a reasonable span of time. Dealing with this and many other issues requires sophisticated electronics, but the NICs handle these issues completely on their own without our help. So, thankfully, while the folks who design NICs worry about all these details, we don't have to!

Getting to Know You

Using the MAC address is a great way to move data around, but this process raises an important question. How does a sending NIC know the MAC address of the NIC to which it's sending the data? In most cases, the sending system already knows the destination MAC address, because the NICs had probably communicated earlier, and each system stores that data. If it doesn't already know the MAC address, a NIC may send a *broadcast* onto the network to ask for it. The MAC address of FF-FF-FF-FF-FF-FF is the **broadcast address**—if a NIC sends a frame using the broadcast address, every single NIC on the network will process that frame. That broadcast frame's data will contain a request for a system's MAC address. The system with the MAC address your system is seeking will read the request in the broadcast packet and respond with its MAC address.

The Complete Frame Movement

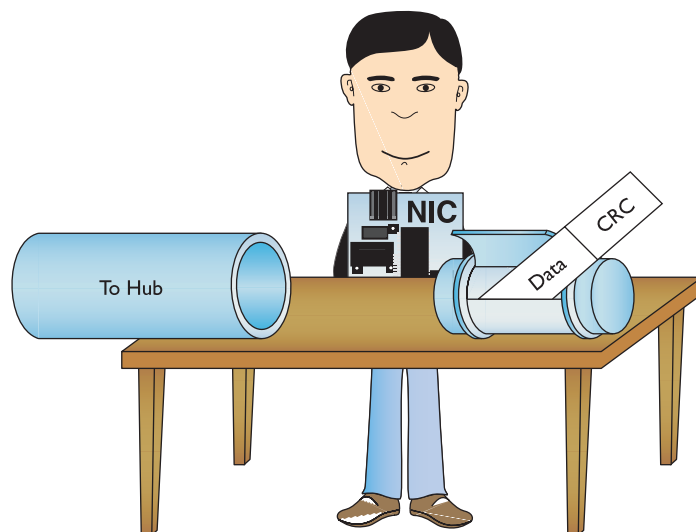
Now that you've seen all the pieces used to send and receive frames, let's put these pieces together and see how a frame gets from one system to another. The basic send/receive process is as follows.



• **Figure 2.20** Building the frame

First, the sending system network operating system (NOS) software—such as Windows Vista—hands some data to its NIC. The NIC begins building a frame to transport that data to the receiving NIC (Figure 2.20).

After the NIC creates the frame, it adds the CRC, and then dumps it and the data into the frame (Figure 2.21).



• **Figure 2.21** Adding the data and CRC to the frame

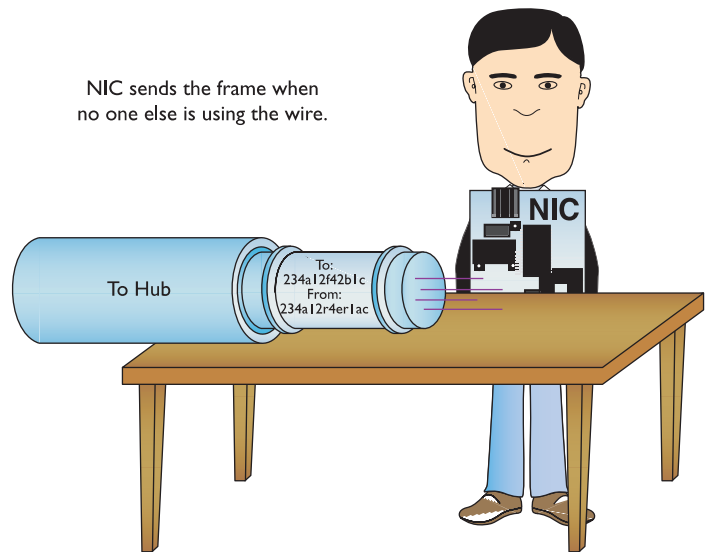
Next, the NIC puts both the destination MAC address and its own MAC address onto the frame. It waits until no other NIC is using the cable, and then sends the frame through the cable to the network (Figure 2.22).

The frame propagates down the wire into the hub, which creates copies of the frame and sends it to every other system on the network. Every NIC receives the frame and checks the MAC address. If a NIC finds that a frame is addressed to it, it processes the frame (Figure 2.23); if the frame is not addressed to it, the NIC erases it.

So, what happens to the data when it gets to the *correct* NIC? First, the receiving NIC uses the CRC to verify that the data is valid. If it is, the receiving NIC strips off all the framing information and sends the data to the software—the network operating system—for processing. The receiving NIC doesn't care what the software does with the data; its job stops the moment it passes on the data to the software.

Any device that deals with a MAC address is part of the OSI **Data Link layer**. Let's update the OSI model to include details about the Data Link layer (Figure 2.24).

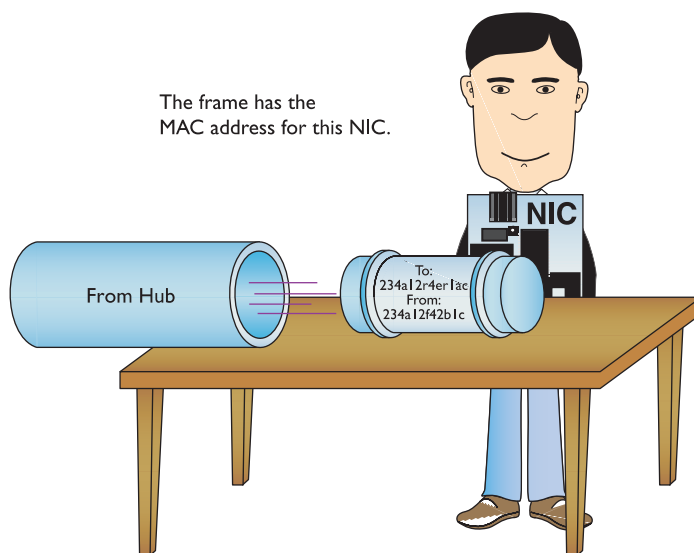
Note that the cabling and the hub are located in the Physical layer. The NIC is in the Data Link layer, but spans two sublayers.



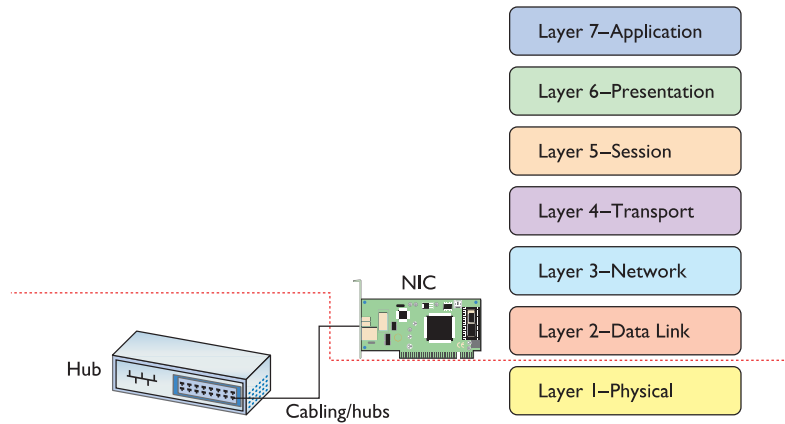
• Figure 2.22 Sending the frame

The Two Aspects of NICs

Consider how data moves in and out of a NIC. On one end, frames move into and out of the NIC's network cable connection. On the other end, data moves back and forth between the NIC and the network operating system software. The many steps a NIC performs to keep this data moving—sending and



• Figure 2.23 Reading an incoming frame



• **Figure 2.24** Layer 1 and Layer 2 are now properly applied to the network.

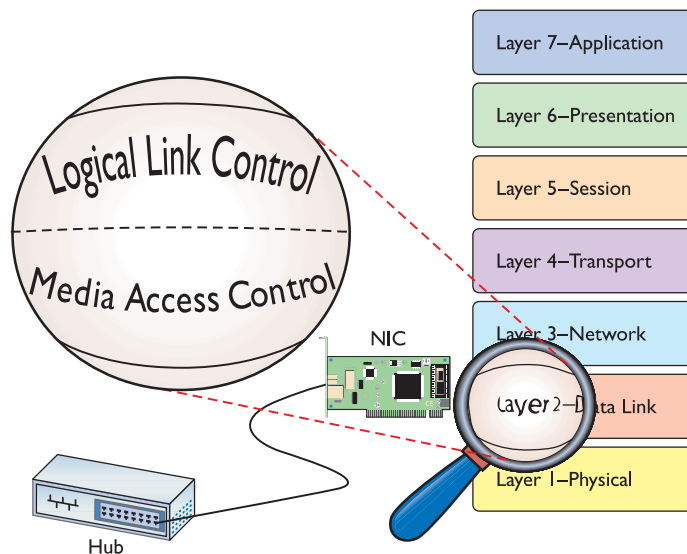
receiving frames over the wire, creating outgoing frames, reading incoming frames, and attaching MAC addresses—are classically broken down into two distinct jobs.

The first job is called the Logical Link Control (LLC). The LLC is the aspect of the NIC that talks to the operating system, places data coming from the software into frames, and creates the CRC on each frame. The LLC is also responsible for dealing with incoming frames: processing those that are addressed to this NIC and erasing frames addressed to other machines on the network.

The second job is called the Media Access Control (MAC), and I bet you can guess what it does! That’s right—it remembers the NIC’s own MAC address and handles the attachment of MAC addresses to frames. Remember that each frame the LLC creates must include both the sender’s and recipient’s MAC addresses. The MAC also ensures that the frames, now complete with their MAC addresses, are then sent along the network cabling. Figure 2.25 shows the Data Link layer in detail.



The CompTIA Network+ exam tests you on the details of the OSI seven-layer model, so know that the Data Link layer is the only layer that has any sublayers.



• **Figure 2.25** LLC and MAC, the two parts of the Data Link layer



Tech Tip

NIC and Layers

Most networking materials that describe the OSI seven-layer model put NICs squarely into the Data Link layer of the model. It's at the MAC sublayer, after all, that data gets encapsulated into a frame, destination and source MAC addresses get added to that frame, and error checking occurs. What bothers most students with placing NICs solely in the Data Link layer is the obvious other duty of the NIC—putting the ones and zeroes on the network cable. How much more physical can you get?

Many teachers will finesse this issue by defining the Physical layer in its logical sense—that it defines the rules for the ones and zeroes—and then ignore the fact that the data sent on the cable has to come from something. The first question when you hear a statement like that—at least to me—is, “What component does the sending?” It's the NIC of course, the only device capable of sending and receiving the physical signal.

Network cards, therefore, operate at both Layer 2 and Layer 1 of the OSI seven-layer model. If cornered to answer one or the other, however, go with the more common answer, Layer 2.

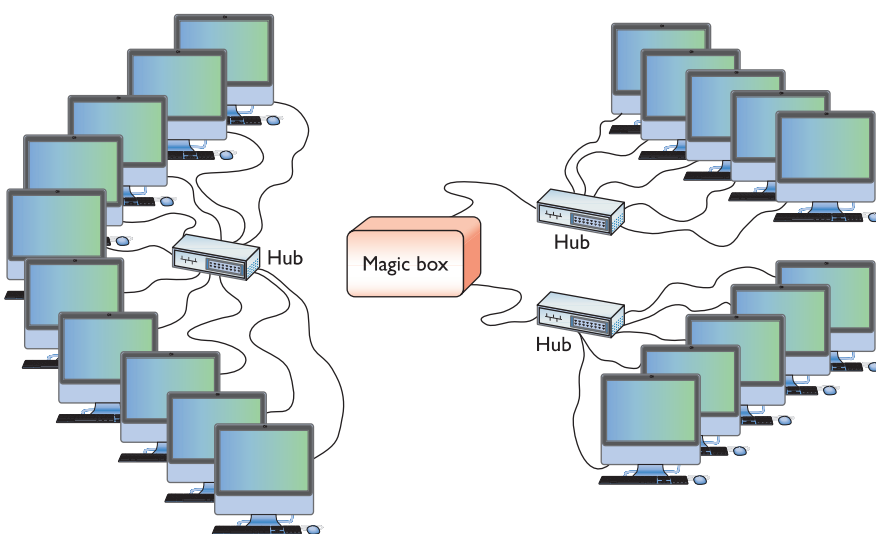
■ Beyond the Single Wire—Network Software and Layers 3–7

Getting data from one system to another in a simple network (defined as one in which all the computers connect to one hub) takes relatively little effort on the part of the NICs. But one problem with simple networks is that computers need to broadcast to get MAC addresses. It works for small networks, but what happens when the network gets big, like the size of the entire Internet? Can you imagine millions of computers all broadcasting? No data could get through. When networks get large, you can't use the MAC addresses anymore. Large networks need a logical addressing method that no longer cares about the hardware and enables us to break up the entire large network into smaller networks called **subnets**. Figure 2.26 shows two ways to set up a network. On the left, all the computers connect to a single hub. On the right, however, the LAN is separated into two five-computer subnets.

To move past the physical MAC addresses and start using logical addressing requires some special software, usually called a **network protocol**. Network protocols exist in every operating system. A network protocol not only has to create unique identifiers for each system, but must also create a set of communication rules for issues like how to handle data chopped up



MAC addresses are also known as physical addresses.



• **Figure 2.26** Large LAN complete (left) and broken up into two subnets (right)



TCP/IP is the most famous network protocol, but there are others.

into multiple packets, and how to make sure that those packets get from one subnet to another. Let's take a moment to learn a bit about the most famous network protocol—TCP/IP—and its unique universal addressing system.

To be accurate, TCP/IP is really several network protocols designed to work together—but two protocols, TCP and IP, do so much work the folks who invented all these protocols named the whole thing TCP/IP. TCP stands for **Transmission Control Protocol**, and IP stands for **Internet Protocol**. IP is the network protocol I need to discuss first; rest assured, however, I'll cover TCP in plenty of detail later.

IP—Playing on Layer 3, the Network Layer

The IP protocol is the primary protocol that TCP/IP uses at Layer 3 (Network) of the OSI model. The IP protocol makes sure that a piece of data gets to where it needs to go on the network. It does this by giving each device on the network a unique numeric identifier called an **IP address**. An IP address is known as a **logical address** to distinguish it from the physical address, the MAC address of the NIC.

Every network protocol uses some type of naming convention, but no two protocols use the same convention. IP uses a rather unique dotted decimal notation (sometimes referred to as a dotted-octet numbering system) based on four 8-bit numbers. Each 8-bit number ranges from 0 to 255, and the four numbers are separated by periods. (If you don't see how 8-bit numbers can range from 0 to 255, don't worry. By the end of this book, you'll understand these naming conventions in more detail than you ever believed possible!) A typical IP address might look like this:

192.168.4.232

No two systems on the same network share the same IP address; if two machines accidentally receive the same address, they won't be able to send or receive data. These IP addresses don't just magically appear—they must be configured by the end user (or the network administrator).

Take a look at Figure 2.26. What makes logical addressing powerful are the magic boxes—called **routers**—that separate each of the subnets. Routers work like a hub, but instead of forwarding packets by MAC address they use the IP address. Routers enable you to take one big network and chop it up into smaller networks. Routers also have a second, very important feature. They enable you to connect networks with different types of cabling or frames. Figure 2.27 shows a typical router. This router enables you to connect a network that uses MAC addresses—a small subnet—to a cable modem network. You can't do that with a hub—the cables, frames, and physical addressing are totally different!

What's important here is for you to appreciate that in a TCP/IP network, each system has two unique identifiers: the MAC address and the IP address. The MAC address (the physical address) is literally burned into the chips on the NIC, while the IP address (the logical address) is simply stored in the software of the system. MAC addresses come with the NIC, so we don't configure MAC addresses, whereas we must configure IP addresses through software. Figure 2.28 shows the MHTechEd network diagram again, this time with the MAC and IP addresses displayed for each system.



• Figure 2.27 Typical small router

This two-address system enables IP networks to do something really cool and powerful: using IP addresses, systems can send each other data without regard to the physical connection!

This capability requires more than the simple assignment of an IP address for each computer. The network protocol must also know where to send the frame, no matter what type of hardware the various computers are running. To do this, a network protocol also uses frames—actually, frames within frames!

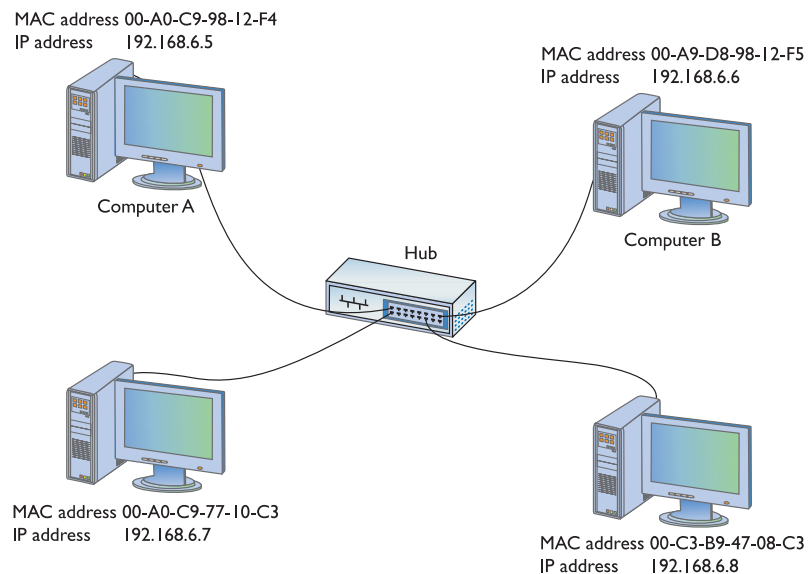
Anything that has to do with logical addressing works at the OSI **Network layer**. At this point there are only two items we know of that operate at the Network layer—routers and the part of the network protocol on every computer that understands the logical addressing (Figure 2.29).

There's Frames in Them Thar Frames!

Whoa! Frames within frames? What are you talking about, Mike? Never fear—I'll show you. Visualize the network protocol software as a layer between the system's software and the NIC. When the IP network protocol gets hold of data coming from your system's software, it places its own frame around that data. We call this inner frame an IP **packet**, so it won't be confused with the *frame* that the NIC will add later. Instead of adding MAC addresses to its packet, the network protocol adds sending and receiving IP addresses. Figure 2.30 shows a typical IP packet; notice the similarity to the frames you saw earlier.

Data type	Packet Count	Recipient's IP address	Sender's IP address	Data
-----------	--------------	------------------------	---------------------	------

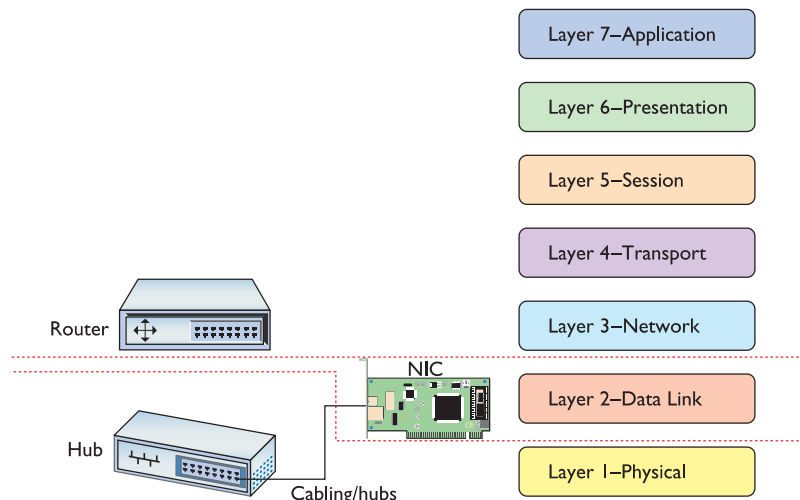
• **Figure 2.30** IP packet



• **Figure 2.28** MHTechEd addressing



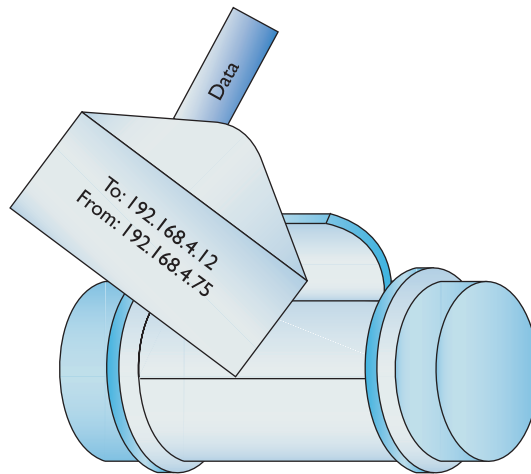
Head to Chapter 7, "TCP/IP Basics," and Chapter 8, "The Wonderful World of Routing," to get much deeper into routers.



• **Figure 2.29** Router now added to the OSI model for the network

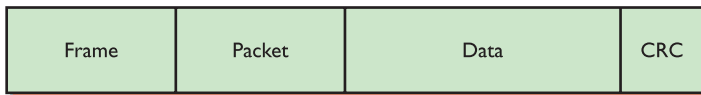


This is a highly simplified IP packet. I am not including lots of little parts of the IP packet in this diagram because they are not important to what you need to understand right now—but don't worry, you'll see them later in the book!



• **Figure 2.31** IP packet in a frame (as a canister)

But IP packets don't leave their PC home naked. Each IP packet is handed to the NIC, which then encloses the IP packet in a regular frame, creating, in essence, a *packet within a frame*. I like to visualize the packet as an envelope, with the envelope in the pneumatic canister frame (Figure 2.31). A more conventional drawing would look like Figure 2.32.

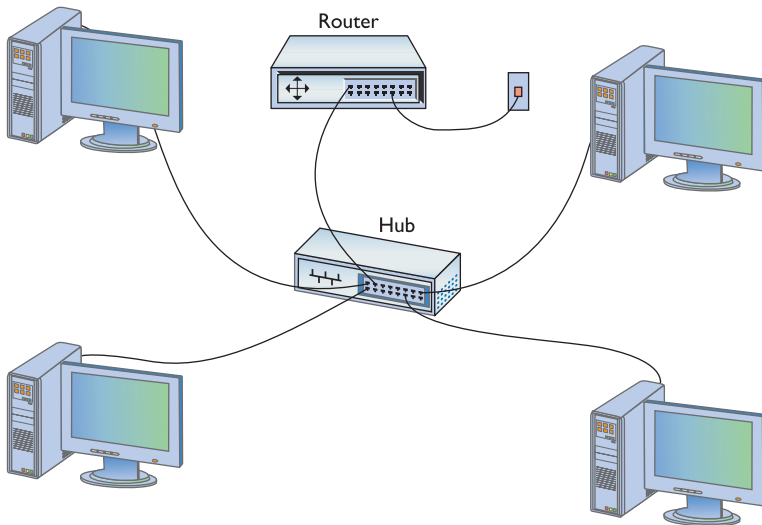


• **Figure 2.32** IP packet in a frame

All very nice, you say, but why hassle with this *packet in a frame* business when you could just use MAC addresses? For that matter, why even bother with this IP thing in the first place? Good question! Let's get back to talking about routers!

Let's say that Janelle wants to access the Internet from her PC using her cable line. A tech could add a cable modem directly to her computer, but the boss wants everyone on the network to get on the Internet using a single cable modem connection. To make this possible, the MHTechEd network will connect to the Internet through a router (Figure 2.33).

The router that MHTechEd uses has two connections. One is just a built-in NIC that runs from the router to the hub. The other connection links the router to a cable modem. Therein lies the answer: cable networks *don't use MAC addresses*. They use their own type of frame that has nothing to do with MAC addresses. If you tried to send a regular network frame on a cable modem network—well, I don't know exactly what would happen, but I assure you, it wouldn't work! For this reason, when a router receives an IP packet inside a frame added by a NIC, it peels off that frame



• **Figure 2.33** Adding a router to the network

and replaces it with the type of frame the cable network needs (Figure 2.34).

Once the network frame is gone, so are the MAC addresses! Thus, you need some *other* naming system the router can use to get the data to the right computer—and that’s why you use IP addresses on a network. After the router strips off the MAC addresses and puts on whatever type of addressing used by the cable modem network, the frame flies through the cable modem network, using the IP address to guide the frame to the router connected to the receiving system. At this point, the process reverses. The router rips off the cable modem frame, adds the MAC address for the receiving system, and sends it on the network, where the receiving system picks it up (Figure 2.35).

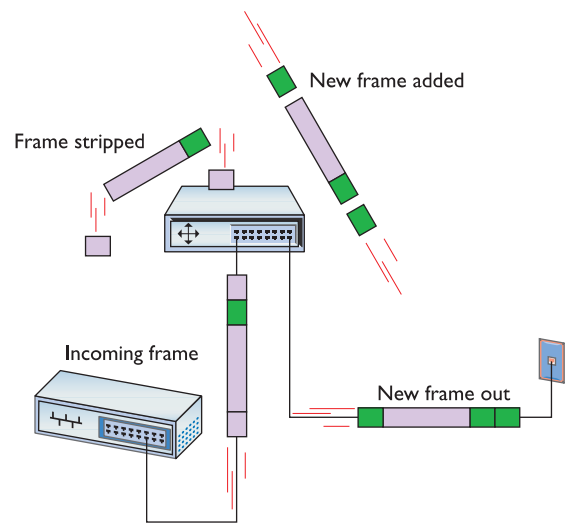
The receiving NIC strips away the MAC address header information and passes the remaining packet off to the software. The networking software built into your operating system handles all the rest of the work. The NIC’s driver software is the interconnection between the hardware and the software. The NIC driver knows how to communicate with the NIC to send and receive frames, but it can’t do anything with the packet. Instead, the NIC driver hands the packet off to other programs that know how to deal with all the separate packets and turn them into Web pages, e-mail messages, files, and so forth.

The Network layer is the last layer that deals directly with hardware. All the other layers of the OSI seven-layer model work strictly within software.

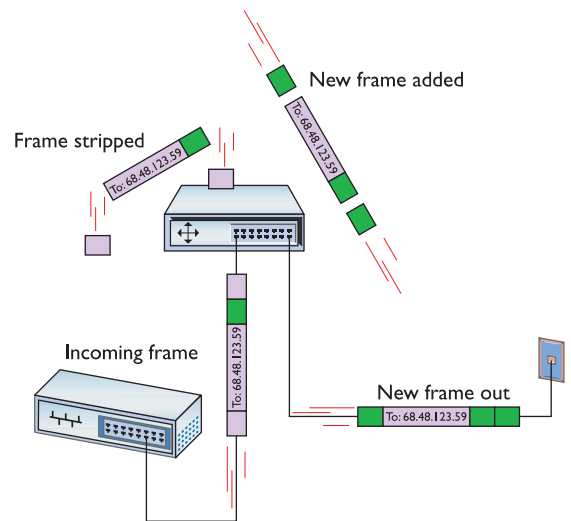
Assembly and Disassembly—Layer 4, the Transport Layer

Because most chunks of data are much larger than a single frame, they must be chopped up before they can be sent across a network. When a serving computer receives a request for some data, it must be able to chop the requested data into chunks that will fit into a packet (and eventually into the NIC’s frame), organize the packets for the benefit of the receiving system, and hand them to the NIC for sending. The receiving system must be able to recognize a series of incoming packets as one data transmission, reassemble the packets correctly based on information included in the packets by the sending system, and verify that all the packets for that piece of data arrived in good shape.

This part is relatively simple—the network protocol breaks up the data into packets and gives each packet some type of sequence number. I like to compare this process to the one that my favorite international shipping company uses. I receive boxes from UPS almost every day; in fact, some days I receive many, many boxes from UPS! To make sure I get all the boxes for one shipment, UPS puts a numbering system, like the one shown in Figure 2.36, on the label of each box. A computer sending data on a network does the same thing. Embedded into the



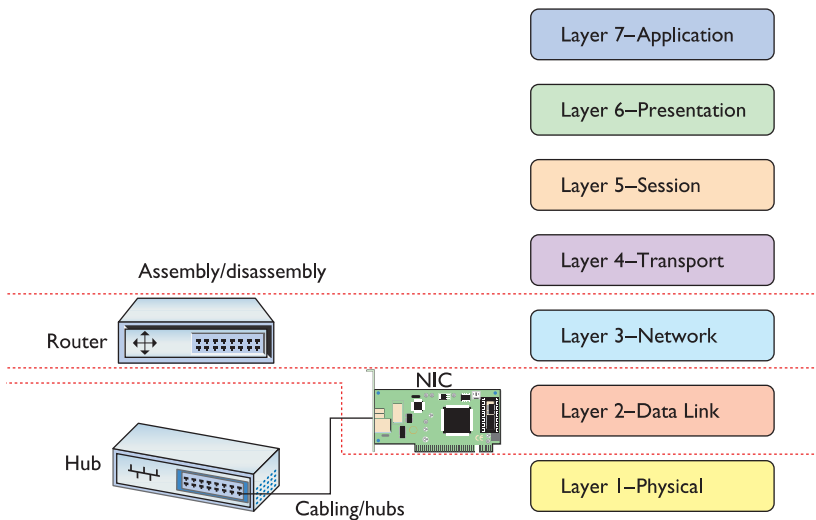
• **Figure 2.34** Router removing network frame and adding one for the cable line



• **Figure 2.35** Router in action

PARCEL ID: 32202273			
301-007		PARCEL #: 6 OF 50 PARCEL WT: 29 LB. 6 OZ.	
ACCOUNT #	CONTROL #	SPECIAL INSTRUCTIONS:	
CUSTOMER P.O. #			
ROW	DAY	LV	QTY
023	042	40	6
ISBN # / DESCRIPTION 0-07-222901-8 An ALL-IN-ONE SE			

• **Figure 2.36** Labeling the boxes



• Figure 2.37 OSI updated

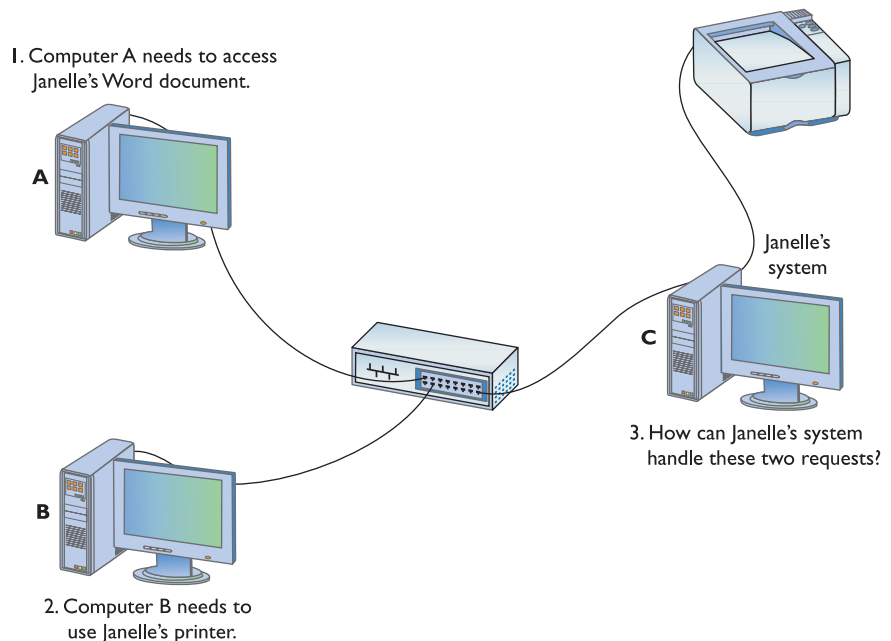
data of each packet is a sequencing number. By reading the sequencing numbers, the receiving system knows both the total number of packets and how to put them back together.

The MHTechEd network just keeps getting more and more complex, doesn't it? And you still haven't seen the Word document get copied, have you? Don't worry; you're almost there—just a few more pieces to go!

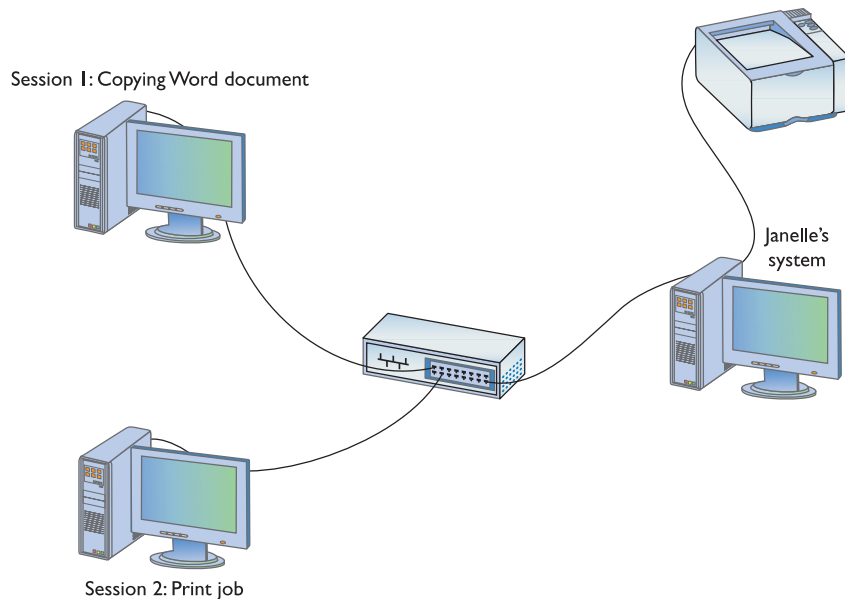
Layer 4, the **Transport layer** of the OSI seven-layer model, has only one big job: it's the assembler/disassembler software. As part of its job, the Transport layer also initializes requests for packets that weren't received in good order (Figure 2.37).

Talking on a Network—Layer 5, the Session Layer

Now that you understand that the system uses software to assemble and disassemble data packets, what's next? In a network, any one system may be talking to many other systems at any given moment. For example, Janelle's PC has a printer used by all the MHTechEd systems, so there's a better than average chance that as Tiffany tries to access the Word document, another system will be sending a print job to Janelle's PC (Figure 2.38). Janelle's system must direct these incoming files, print jobs, Web pages, and so on to the



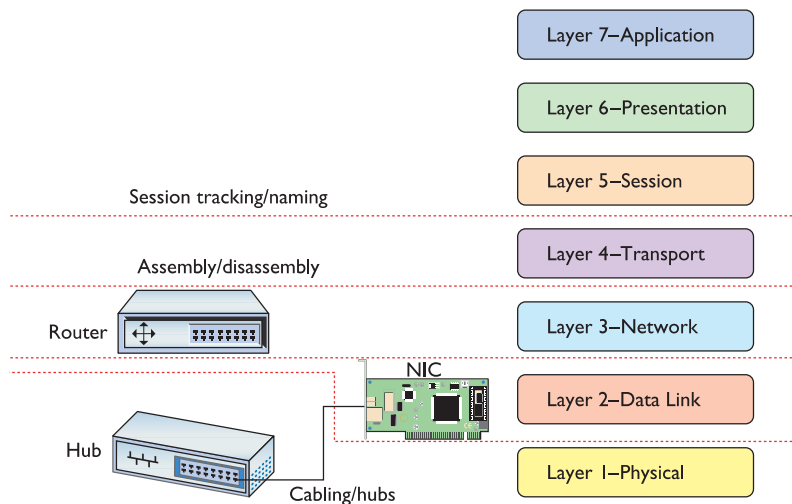
• Figure 2.38 Handling multiple inputs



• **Figure 2.39** Each request becomes a session.

right programs (Figure 2.39). Additionally, the operating system must enable one system to make a connection to another system to verify that the other system can handle whatever operation the initiating system wants to perform. If Bill's system wants to send a print job to Janelle's printer, it first contacts Janelle's system to ensure that it is ready to handle the print job. The **session software** handles this part of networking.

Layer 5, the **Session layer** of the OSI seven-layer model, handles all the sessions for a system. The Session layer initiates sessions, accepts incoming sessions, and opens and closes existing sessions. The Session layer also keeps track of computer naming conventions, such as calling your computer SYSTEM01 or some other type of name that makes more sense than an IP or MAC address (Figure 2.40).



• **Figure 2.40** OSI updated



Try This!

See Your Sessions

How many sessions does a typical system have running at one time? Well, if you have a TCP/IP network, you can run the NETSTAT program from a command prompt to see all of them. Open a command prompt and type the following:

```
netstat -a
```

Then press the ENTER key to see your sessions. Don't worry about trying to interpret what you see—Chapter 9, "TCP/IP Applications," covers NETSTAT in detail. For now, simply appreciate that each line in the output of NETSTAT is a session. Count them!

Standardized Formats, or Why Layer 6, Presentation, Has No Friends

One of the most powerful aspects of a network lies in the fact that it works with (almost) any operating system. Today's networks easily connect, for example, a Macintosh system to a Windows PC, despite the fact that these different operating systems use different formats for many types of data. Different data formats used to drive us crazy back in the days before word processors (like Microsoft Word) could import or export a thousand other word processor formats (Figure 2.41).

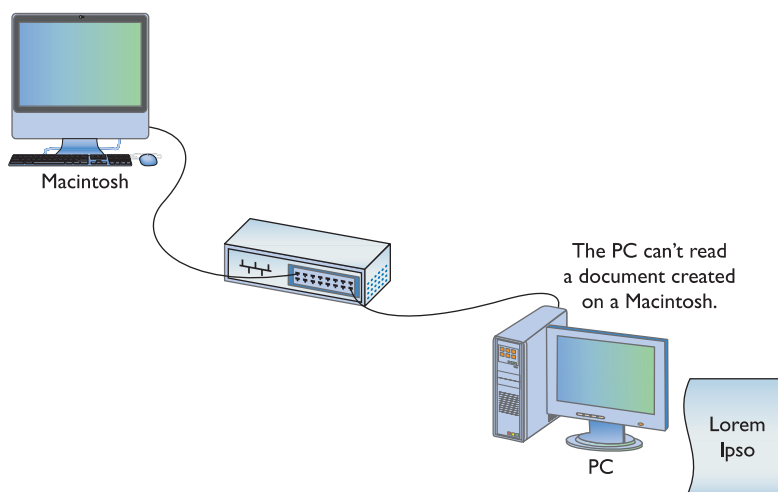
This created the motivation for standardized formats that anyone—at least with the right program—could read from any type of computer. Specialized file formats, such as Adobe's popular Portable Document Format (PDF) for documents and PostScript for printing, provide standard formats that any system, regardless of the operating system, can read, write, and edit (Figure 2.42).



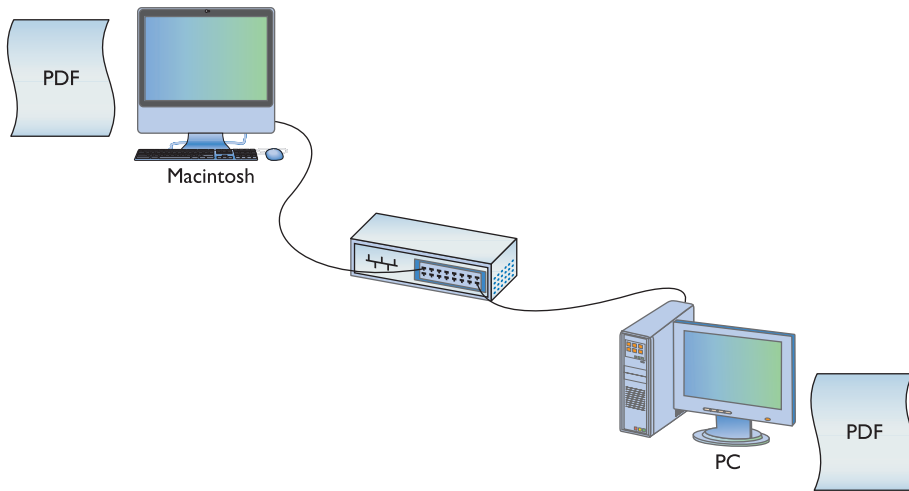
Tech Tip

Acrobat as Open Standard

Adobe released the PDF standard to ISO in 2007 and PDF became the ISO 32000 open standard. Adobe Acrobat remains the premier application for reading and editing PDF documents, so most folks call PDF documents Acrobat files.



• **Figure 2.41** Different data formats were often unreadable between systems.



• **Figure 2.42** Everyone recognizes PDF files!

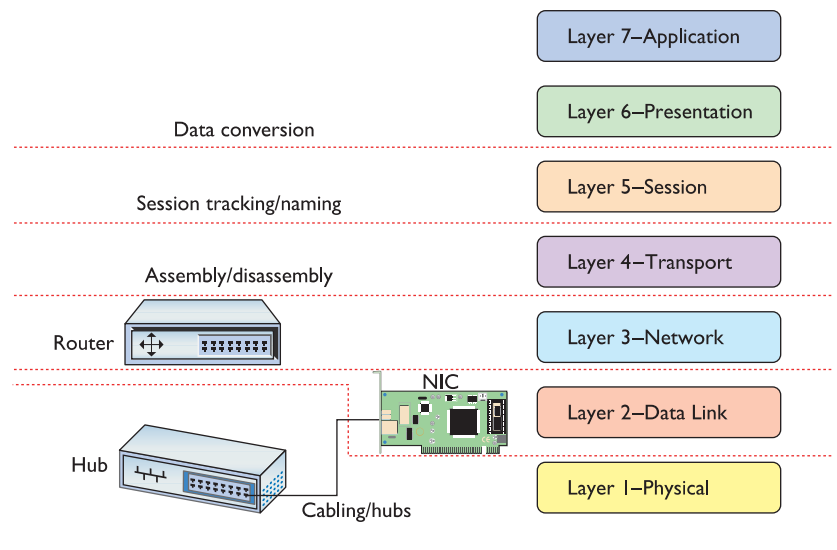
Layer 6, the **Presentation layer** of the OSI seven-layer model, handles converting data into formats that are readable by the system. Of all the OSI layers, the high level of standardization of file formats has made the Presentation layer the least important and least used (Figure 2.43).

Network Applications—Layer 7, the Application Layer

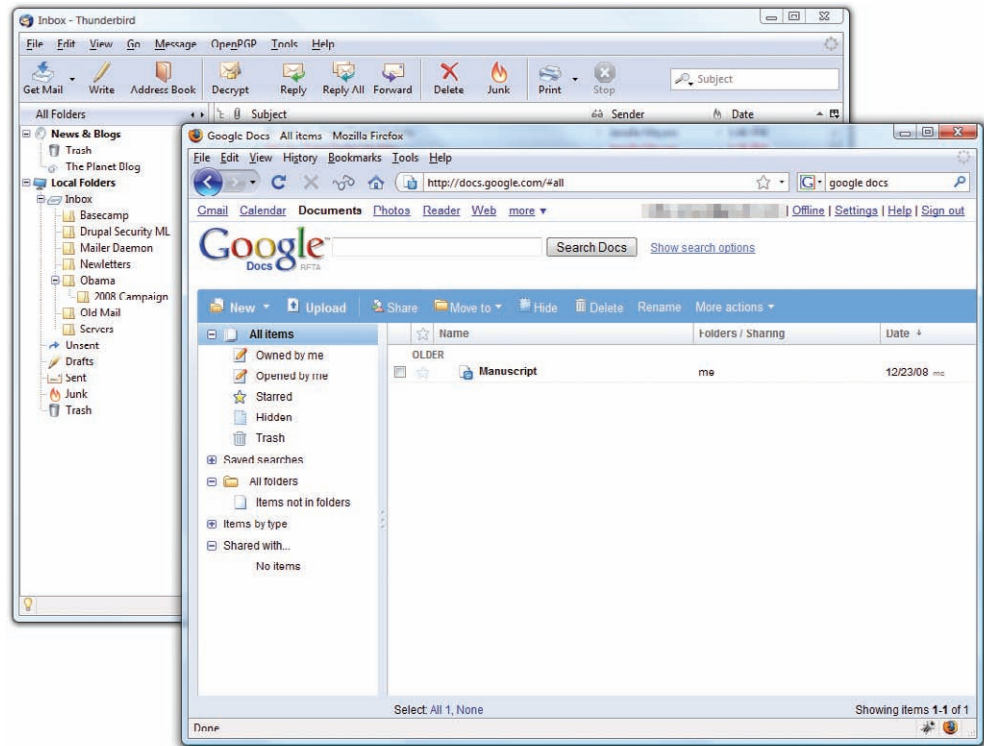
The last, and most visible, part of any network is the software applications that use it. If you want to copy a file residing on another system in your network, you need an application like Network in Windows Vista (or My Network Places in earlier versions of Windows) that enables you to access files on remote systems. If you want to view Web pages, you need a Web browser like Internet Explorer or Mozilla Firefox. The people who use a network experience it through an application. A user who knows nothing about all the other parts of a network may still know how to open an e-mail application to retrieve mail (Figure 2.44).

Applications may include a number of additional functions, such as encryption, user authentication, and tools to control the look of the data. But these functions are specific to the given applications. In other words, if you want to put a password on your Word document, you must use the password functions of Word to do so.

Layer 7, the **Application layer** of the OSI seven-layer model, refers to the code built into all operating systems that enables network-aware applications. All operating systems have Application Programming Interfaces (APIs) that

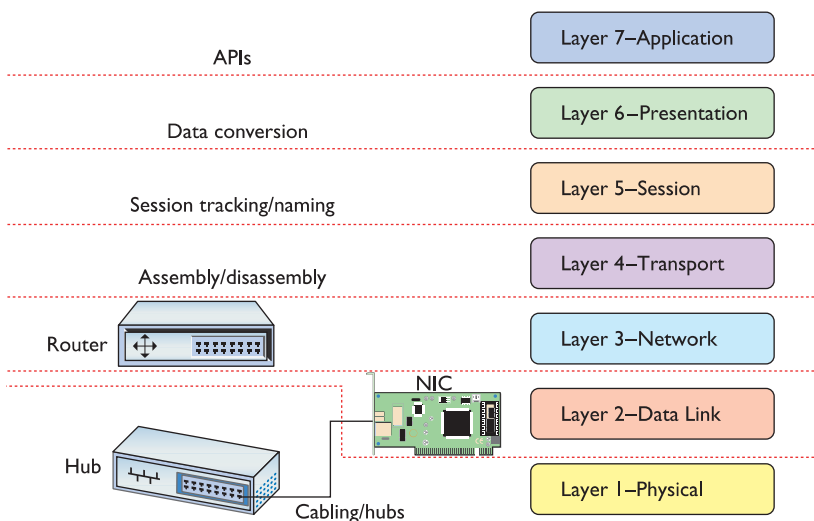


• **Figure 2.43** OSI updated



• **Figure 2.44** Network applications at work

programmers can use to make their programs network aware (Figure 2.45). An API in general provides a standard way for programmers to enhance or extend an application's capabilities.



• **Figure 2.45** OSI updated

How Tiffany Gets Her Document

Okay, you've now seen all the different parts of the network; keep in mind that not all networks contain all these pieces. Certain functions, such as encryption (which is used to make readable text unreadable), may or may not be present, depending on the needs of the particular network. With that understanding, let's watch the network do its magic as Tiffany gets Janelle's Word document.

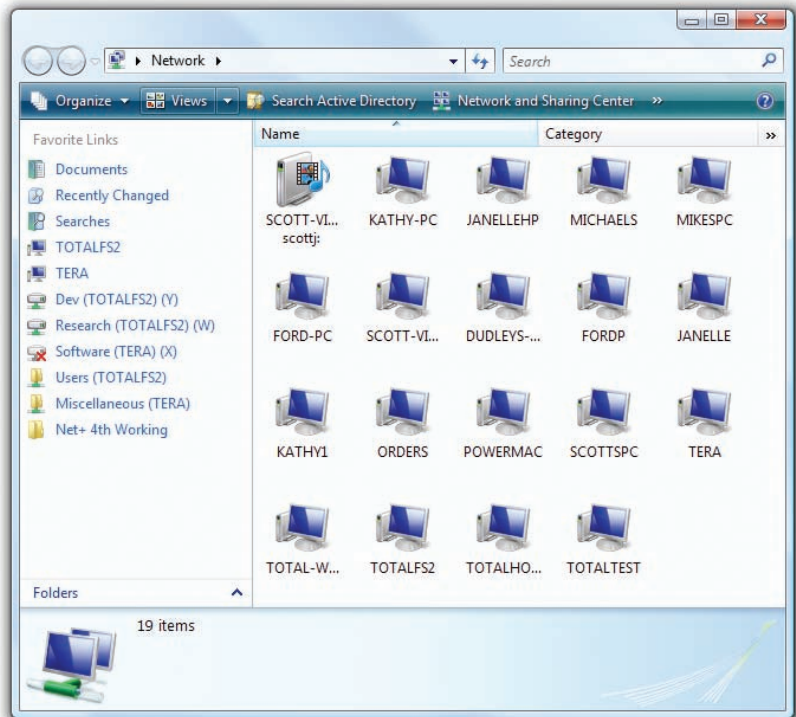
The Application layer gives Tiffany choices for accessing Janelle's Word document. She can access the document by opening Word on her system, selecting File | Open, and taking the file off Janelle's desktop; or she can use Network,

Computer, or Windows Explorer to copy the Word file from Janelle's desktop to her computer, and then open her own copy of the file in Word. Tiffany wants to make changes to the document, so she chooses to copy it over to her system. This will leave an original copy on Janelle's system, so Janelle can still use it if she doesn't like Tiffany's changes.

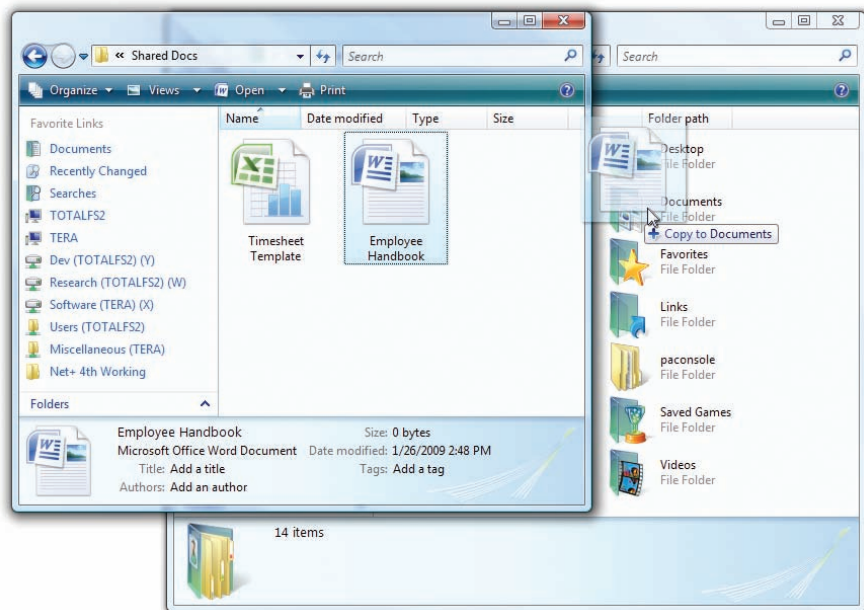
Tiffany's goal is to copy the file from Janelle's shared Desktop folder to her system. Let's watch it happen. The process begins when Tiffany opens her Network application. The Network application shows her all the computers on the MHTechEd network (Figure 2.46).

Both systems are PCs running Word, so Tiffany doesn't need to worry about incompatible data formats, which means the Presentation layer (Layer 6) doesn't come into play here. As soon as Tiffany clicks the icon for Janelle's system in Network, the two systems begin to use the OSI Session layer (Layer 5) and establish a session. Janelle's system checks a database of user names and privileges to see what Tiffany can and cannot do on Janelle's system. This checking process takes place a number of times during the process as Tiffany accesses various shared folders on Janelle's system. By this time, a session has been established between the two machines. Tiffany now opens the shared folder and locates the Word document. To copy the file, she drags and drops the Word document icon from her Network application onto her desktop (Figure 2.47).

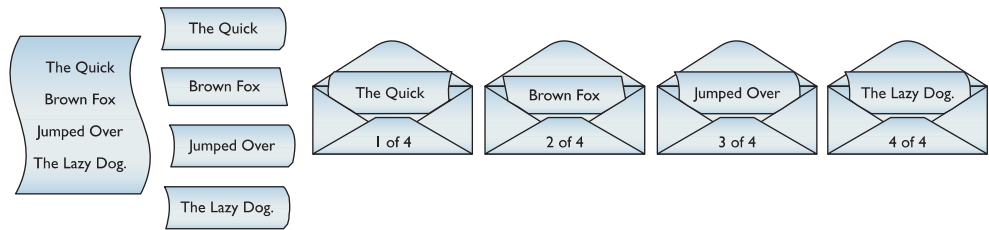
This simple act starts a series of actions. First, Janelle's OSI Transport layer (Layer 4) software begins to chop the Word document into packets



• **Figure 2.46** Network application showing computers on the MHTechEd network

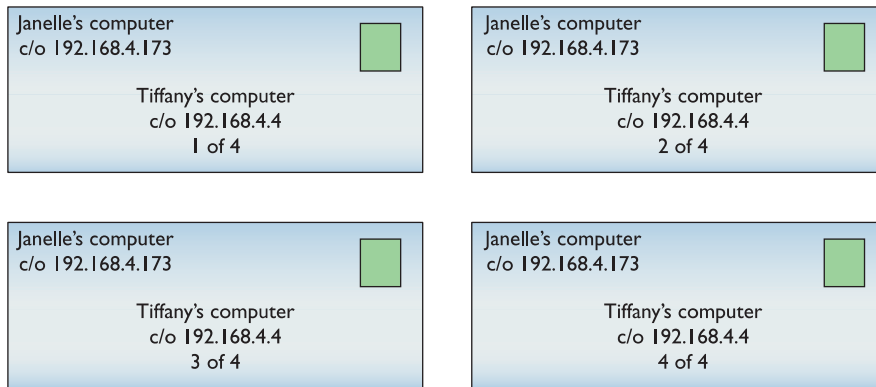


• **Figure 2.47** Copying the Word document



• **Figure 2.48** Chopping the Word document

and assign each a sequence number, so that Tiffany's system will know how to reassemble the packets when they arrive on her system (Figure 2.48).



• **Figure 2.49** Creating and addressing packets

After Janelle's system chops the data into numbered packets, the OSI Network layer (Layer 3) software adds to each packet the address of Tiffany's system, as well as Janelle's address (Figure 2.49).

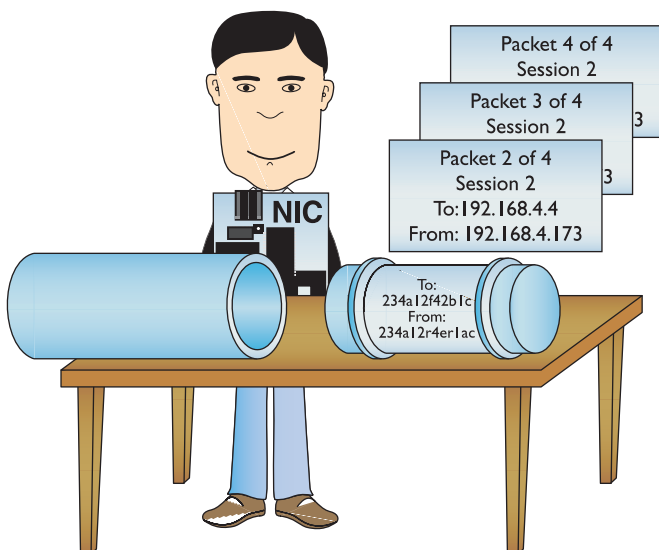
The packets now get sent to the NIC for transfer. The NIC's OSI Data Link layer (Layer 2) adds around each packet a frame that contains the MAC addresses for Tiffany's and Janelle's systems (Figure 2.50).

As the NIC assembles each frame, it checks the network cabling to see if the cable is busy. If not, it sends the frame down the wire, finally using the Physical

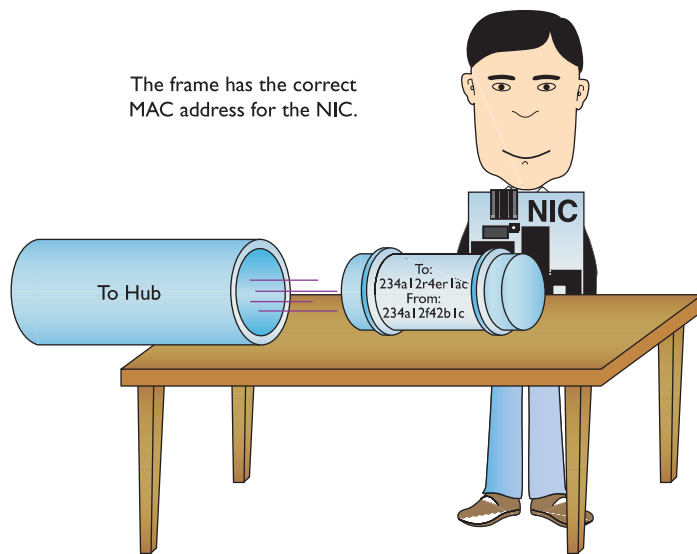
layer (Layer 1). Now it's time to reverse the process as the frames arrive at Tiffany's system. The frame goes through the hub and off to every other NIC in the network. Each NIC looks at the MAC address. All the other systems discard the frame, but Tiffany's system sees its MAC address and grabs it (Figure 2.51).

As Tiffany's NIC begins to take in frames, it checks each one using the CRC to ensure the validity of the data in the frame. After verifying the data, the NIC strips off both the frame and the CRC and passes the packet up to the next layer. Tiffany's system then begins to reassemble the individual packets back into the complete Word document. If Tiffany's system fails to receive one of the packets, it simply requests that Janelle's computer resend it.

Once Tiffany's system reassembles the completed Word document, it sends the document to the proper application—in this case, Windows Explorer. Once the system copies the file to the desktop, the network applications erase the session connection information



• **Figure 2.50** Creating frames



• **Figure 2.51** Tiffany's system grabbing a frame

from each system and prepare for what Tiffany and Janelle may want to do next.

The most amazing part of this process is that the users see virtually none of it. Tiffany simply opened her Network application, located Janelle's system, located the shared folder containing the Word document, and then dragged and dropped the Word document onto her desktop. This is the beauty and mystery of networks. The complexities of the different parts of software and hardware working together aren't noticed by users—nor should they be!

The Tech's Troubleshooting Tool

The OSI seven-layer model provides you with a way to conceptualize a network to determine what could cause a specific problem when the inevitable problems occur. Users don't need to know anything about this, but techs can use the OSI model for troubleshooting.

If Jane can't print to the networked printer, for example, the OSI model can help solve the problem. If her NIC shows activity, then you can set aside both the Physical layer (Layer 1) and Data Link layer (Layer 2) and go straight to the Network layer (Layer 3). If her computer has a proper IP address, then Layer 3 is done and you can move on up to check other layers to solve the problem.

By understanding how network traffic works throughout the model, you can troubleshoot with efficiency. You can use the OSI model during your career as a network tech as the basis for troubleshooting.

Chapter 2 Review

■ Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about networking.

Describe models such as the OSI seven-layer model

- The OSI seven-layer model defines the role played by each protocol. The OSI model also provides a common jargon that network techs can use to describe the function of any network protocol.
- Layer 1, the Physical layer, includes anything that moves data from one system to another, such as cabling or radio waves.
- Layer 2, the Data Link layer, defines the rules for accessing and using the Physical layer. The Data Link layer is divided into two sublayers: Media Access Control (MAC) and Logical Link Control (LLC). The MAC sublayer controls access to the Physical layer, or shared media. It encapsulates (creates the frames for) data sent from the system, adding source and destination MAC addresses and error-checking information; it also decapsulates (removes the MAC addresses and CRC from) data received by the system. The LLC sublayer provides an interface with the Network layer protocols. It is responsible for the ordered delivery of frames, including retransmission of missing or corrupt packets, and for flow control (moderating data flow so one system doesn't overwhelm the other). Any device that deals with a MAC address is part of the Data Link layer.
- Layer 3, the Network layer, is the last layer to work directly with hardware. It adds to the packets unique identifiers (such as IP addresses) that enable routers to make sure the packets get to the correct system without worrying about the type of hardware used for transmission. Anything having to do with logical addressing works at the Network layer.
- Layer 4, the Transport layer, breaks up data received from the upper layers into smaller pieces for transport, called packets.
- Layer 5, the Session layer, manages the connections between machines on the network.
- Layer 6, the Presentation layer, presents data from the sending system in a form that the applications on the receiving system can understand.

- Layer 7, the Application layer, defines a set of tools that programs can use to access the network. Application layer programs provide services to the programs that the users see.

Explain the major functions of network hardware with OSI Layers 1–2

- Network hardware consists most often of three components: cabling, a hub, and network interface cards (NICs).
- Every NIC has a hard-coded, unique identifying 48-bit number, called a media access control address (MAC address). The MAC address can be broken down into an organizationally unique identifier (OUI) and a device ID.
- All networks transmit data by breaking whatever is moving across the network (files, print jobs, Web pages, and so forth) into discrete chunks called *frames*. A frame is basically a container for a chunk of data moving across a network. The NIC creates and sends, as well as receives and reads, these frames.
- A frame begins with the MAC address of the NIC to which the data is to be sent, followed by the MAC address of the sending NIC. Then comes the data, followed by a special bit of checking information called the cyclic redundancy check (CRC) that the receiving NIC uses to verify that the data got to it correctly.
- Hubs make copies of frames they receive and forward them to every system on the network. Only the system for which the frame was destined disassembles the frame. All other systems ignore the frame.
- If the sending system does not know the MAC address of the receiving system, it broadcasts the frame using the broadcast address. In this case, every system inspects the frame and no system ignores it.

Describe the functions of network software with OSI Layers 3–7

- Network protocols create software-addressing schemes that enable network traffic to cross routers, which can connect networks of differing cabling or frame types. TCP/IP is a well-known

suite of network protocols, of which TCP and IP are the individual protocols that do most of the work in the suite.

- The Internet Protocol (IP) makes sure that a piece of data gets to where it needs to go on the network. It does this by giving each device on the network a unique numeric identifier, called, appropriately, an IP address. The IP address is a logical address stored in the software of the system, whereas a MAC address is a physical address burned into the NIC. Both the IP address and MAC address can be viewed by running the command-line IPCONFIG program.
- When packaging data, IP creates a packet containing the receiving and sending computers' IP addresses, and then places that packet into a network frame.
- The network protocol breaks up data into packets and gives each packet some type of sequence number so that the receiving PC can reassemble the data properly.
- Session software handles the process of differentiating between various types of connections on a PC. The NETSTAT program can be used to view existing sessions.
- Standardized data formats, such as PDF, enable computers running on different platforms to share data across a network, the result of which is that the Presentation layer is the least important and least used of the seven layers.
- Network applications, such as Network (or My Network Places), enable you to access files or features on remote systems.

■ Key Terms

Application layer (29)

broadcast address (17)

cyclic redundancy check (CRC) (16)

Data Link layer (19)

device ID (14)

frame (15)

hub (12)

Internet Protocol (IP) (22)

IP address (22)

logical address (22)

MAC address (14)

network interface card (NIC) (13)

Network layer (23)

network protocol (21)

OSI seven-layer model (8)

organizationally unique identifier (OUI) (14)

packet (23)

physical address (14)

Physical layer (12)

Presentation layer (29)

protocol (11)

router (22)

Session layer (27)

session software (27)

subnet (21)

Transmission Control Protocol (TCP) (22)

Transport layer (26)

unshielded twisted pair (UTP) (12)

■ Key Term Quiz

Use the Key Terms list to complete the sentences that follow. Not all terms will be used.

1. The _____ is an example of software that creates packets for moving data across networks.
2. Most often, the _____ provides the physical connection between the PC and the network.
3. Using the _____ enables a computer to send a packet that every other PC on the network will process.
4. You can connect two very different networks by using a(n) _____.
5. Every NIC has a hard-coded identifier called a(n) _____.
6. The _____ provides an excellent tool for conceptualizing how a network works.
7. On a sending machine, data gets broken up at the _____ of the OSI seven-layer model.
8. NICs encapsulate data into a(n) _____ for sending that data over a network.

9. A(n) _____ enables multiple machines to connect over a network.
10. The _____ provides the key interface between the Physical and Network layers.

■ Multiple-Choice Quiz

- Which of the following OSI layers converts the ones and zeroes to electrical signals and places these signals on the cable?
 - Physical layer
 - Transport layer
 - Network layer
 - Data Link layer
- The term “unshielded twisted pair” describes which of the following network components?
 - Cable
 - Hub
 - Router
 - NIC
- From the options that follow, select the one that best describes the contents of a typical network frame.
 - Sender’s MAC address, recipient’s MAC address, data, CRC
 - Recipient’s MAC address, sender’s MAC address, data, CRC
 - Recipient’s IP address, sender’s IP address, data, CRC
 - Recipient’s e-mail address, sender’s e-mail address, data, CRC
- Which of the following is most likely to be a MAC address assigned to a NIC?
 - 192.168.1.121
 - 24.17.232.7B
 - 23.4F.17.8A.4C.10
 - 713.555.1212
- Which layer of the OSI model involves routing?
 - Physical layer
 - Transport layer
 - Network layer
 - Data Link layer
- How much data can a typical frame contain?
 - 500 bytes
 - 1500 bytes
 - 1500 kilobytes
 - 1 megabyte
- Which of the following best describes an IP address?
 - A unique dotted decimal notation burned into every NIC
 - A unique 48-bit identifying number burned into every NIC
 - A dotted decimal notation assigned to a NIC by software
 - A 48-bit identifying number assigned to a NIC by software
- Which layer of the OSI model makes sure the data is in a readable format for the Application layer?
 - Application layer
 - Presentation layer
 - Session layer
 - Transport layer
- Which of the following enables you to connect a PC via modem to an IP network?
 - NIC
 - Hub
 - Router
 - TCP
- What handles the initial connection between two computers to verify that the receiving system can handle the network request of the sending system?
 - NIC
 - IP
 - TCP
 - Session software

11. What is Layer 3 of the OSI seven-layer model?
 - A. Presentation layer
 - B. Session layer
 - C. Transport layer
 - D. Network layer
12. What component of Layer 2 of the OSI seven-layer model is responsible for the ordered delivery of frames, including retransmission of missing or corrupt packets?
 - A. MAC sublayer
 - B. LLC sublayer
 - C. CRC sublayer
 - D. Data Link sublayer
13. Which components work at Layer 1 of the OSI seven-layer model? (Select two.)
 - A. Cables
 - B. Hub
 - C. Network protocol
 - D. Session software
14. Andalyn says complete 48-bit MAC addresses are allocated to NIC manufacturers from the IEEE. Buster says the IEEE only assigns the first 24 bits to manufacturers. Carlos says the IEEE assigns only the last 24 bits to manufacturers. Who is correct?
 - A. Only Andalyn is correct.
 - B. Only Buster is correct.
 - C. Only Carlos is correct.
 - D. No one is correct.
15. If a sending system does not know the MAC address of the intended recipient system, it sends a broadcast frame with what MAC address?
 - A. 192.168.0.0
 - B. FF-FF-FF-FF-FF-FF
 - C. 11-11-11-11-11-11
 - D. 00-00-00-00-00-00

■ Essay Quiz

1. Some new techs at your office are confused by the differences between a NIC's frame and an IP packet. Write a short essay describing the two encapsulations, including the components that do the encapsulating.
2. Your boss has received a set of files with the file extension .WP and is worried because he's never seen that extension before. He wants people to have access to the information in those files from anywhere in the network. Write a short memo describing how Microsoft Word can handle these files, including discussion of how that fits with the OSI seven-layer model.
3. After reading this chapter over your shoulder, your co-worker is not clear about how the Tiffany and Janelle story connects with the OSI seven-layer model. Write an essay that describes each layer and include an example from the Tiffany and Janelle story that fits at each layer.

Lab Projects

• Lab Project 2.1

Examine your classroom network. What components does it have? How would you classify

those components according to the OSI seven-layer model?

• Lab Project 2.2

Create a mnemonic phrase to help you remember the OSI seven-layer model. With two layers beginning with the letter *P*, how will you differentiate in your

mnemonic between Presentation and Physical? How will you incorporate the two sublayers of the Data Link layer?