CHAPTER 5

Ethernet Basics

The Network+ Certification exam expects you to know how to

- 1.2 Specify the main features of 802.2 (Logical Link Control) [and] 802.3 (Ethernet) . . . networking technologies, including speed, access method, topology, [and] media
- 1.5 Recognize the following media types and describe their uses: coaxial cable
- 1.6 Identify the purposes, features, and functions of bridges
- 2.3 Identify the OSI (Open Systems Interconnect) layers at which bridges operate

To achieve these goals, you must be able to

- Describe the concept of Ethernet
- Define Ethernet cabling systems
- Explain the function of repeaters and bridges

In the beginning, there were no networks. Computers were isolated, solitary islands of information in a teeming sea of proto-geeks. If you wanted to move a file from one machine to another—and proto-geeks were as much into that as modern geeks—you had to use Sneakernet, which meant you saved the file on a disk, laced up your tennis shoes, and hiked over to the other system. All that walking no doubt produced lots of health benefits, but frankly, proto-geeks weren't all that into health benefits—they were into speed, power, and technological coolness in general. (Sound familiar?) It's no wonder, then, that geeks everywhere agreed on the need to replace Sneakernet with a faster and more efficient method of sharing data. The method they came up with is the subject of this chapter.

Historical/Conceptual

In 1973, Xerox answered the challenge of moving data without sneakers by developing *Ethernet*, a networking technology standard based on a bus topology. The Ethernet standard, which predominates in today's networks, defines many issues involved in transferring data between computer systems. The original Ethernet used a single piece of coaxial cable to connect several computers, enabling them to transfer data at a rate of up to 3 Mbps. Although slow by today's standards, this early version of Ethernet was a huge improvement over Sneakernet methods, and served as the foundation for all later

versions of Ethernet. It remained a largely in-house technology within Xerox until 1979, when Xerox decided to look for partners to help promote Ethernet as an industry standard. They worked with Digital Equipment Corporation (DEC) and Intel to publish what became known as the Digital-Intel-Xerox (DIX) standard. Running on coaxial cable, the DIX standard enabled multiple computers to communicate with each other at a screaming 10 Mbps. Although 10 Mbps represents the low end of standard network speeds today, at the time it was revolutionary. These companies then transferred control of the Ethernet standard to the IEEE, which in turn created the now famous *802.3 (Ethernet)* committee that continues to control the Ethernet standard to this day. The remainder of this book follows common parlance in using the terms Ethernet and IEEE 802.3 interchangeably.

TIP The source for all things Ethernet is but a short click away on the Internet. Check out www.ieee802.org for starters.

Ethernet today is not a single network technology, but rather a standard for a family of network technologies that share the same basic bus topology, frame type, and network access method. Ethernet manufacturers have created a number of network technologies since Ethernet first came onto the scene more than 30 years ago. Different types of Ethernet use completely different cabling and NICs. This chapter shows you how Ethernet works, and then shows you the first generation of Ethernet technologies: Thick Ethernet (a.k.a. 10Base5) and Thin Ethernet (a.k.a. 10Base2). Both of these versions of Ethernet used a physical bus topology—a single cable that connected to all the computers on the network.

Providing a clear and concise definition of Ethernet has long been one of the major challenges in teaching networking. This difficulty stems from the fact that Ethernet has changed over the years to incorporate new and improved technology. Most folks won't even try to define Ethernet, but here's my best attempt at a current definition.

Ethernet is a standard for a family of network technologies that share the same basic bus topology, frame type, and network access method. Because the technologies share these essential components, you can communicate between them just fine. The implementation of the network might be different, but the frames remain the same.

How Ethernet Works

Ethernet's designers faced the same challenges as the designers of any network: how to send data across the wire, how to identify the sending and receiving computers, and how to determine which computer should use the shared cable at what time. The engineers resolved these issues by using data frames that contain MAC addresses to identify computers on the network, and by using a process called *CSMA/CD* to determine which machine should access the wire at any given time. You saw some of this in action in Chapter 3, "Building a Network with OSI," but now I need to introduce you to a bunch of new terms, so let's look at each of these solutions.

Physical Bus

The first generations of Ethernet used a physical and logical bus topology. A physical bus means a physical cable, and all early Ethernet networks were distinguished by a single coaxial cable snaking around the network, usually in the ceiling. Each computer on the network connected into the cable. This single cable had many interchangeable names, among them the *segment*, the *cable*, and the *bus*. Be comfortable using any of these terms to describe that single piece of cable that connects all the computers on an Ethernet network (Figure 5-1).

Organizing the Data: Ethernet Frames

All network technologies break data transmitted between computers into smaller pieces called *frames*, as you'll recall from Chapter 3. Using frames addresses two networking issues. First, it prevents any single machine from monopolizing the shared bus cable. Second, frames make the process of retransmitting lost data more efficient.

TIP The terms *frame* and *packet* are often used interchangeably, especially on exams! This book uses the terms more strictly. You'll recall from Chapter 3, "Building a Network with OSI," that frames are based on MAC addresses; *packets* are generally associated with data assembled by the IP protocol at Layer 3 of the OSI seven-layer model.

The process you saw in the previous chapter of transferring a word processing document between two computers illustrates these two issues. First, if the sending computer sends the document as a single huge frame, it will monopolize the cable and prevent other machines from using the cable until the entire file gets to the receiving system. Using relatively small frames enables computers to share the cable easily—each computer listens on the segment, sending a few frames of data whenever it detects that no other computer is transmitting. Second, in the real world, bad things can happen to good data. When errors occur during transmission, the sending system must retransmit the frames that failed to get to the receiving system in good shape. If a word processing

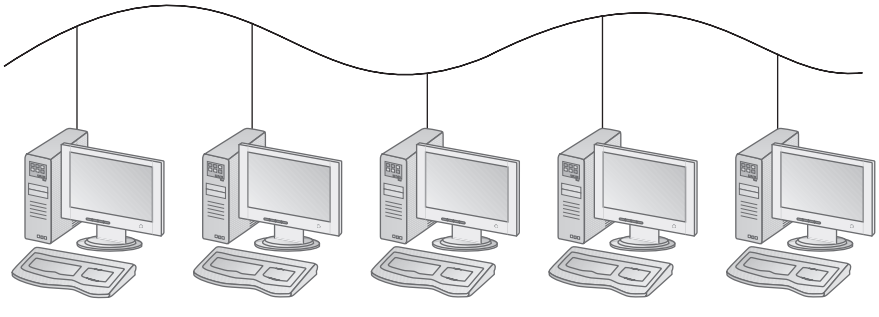

Figure 5-1 Ethernet segment

document were transmitted as a single massive frame, the sending system would have to retransmit the entire frame—in this case, the entire document. Breaking the file up into smaller frames enables the sending computer to retransmit only the damaged frames. Because of their benefits—shared access and reduced retransmission—all networking technologies use frames, and Ethernet is no exception to that rule.

In Chapter 3, you saw a generic frame. Let's take what you know of frames and expand on that knowledge by inspecting the details of an Ethernet frame. A basic Ethernet frame contains seven basic pieces of information: the preamble, the MAC address of the frame's recipient, the MAC address of the sending system, the length of the data, the data itself, a pad, and a frame check sequence. Figure 5-2 shows these components.

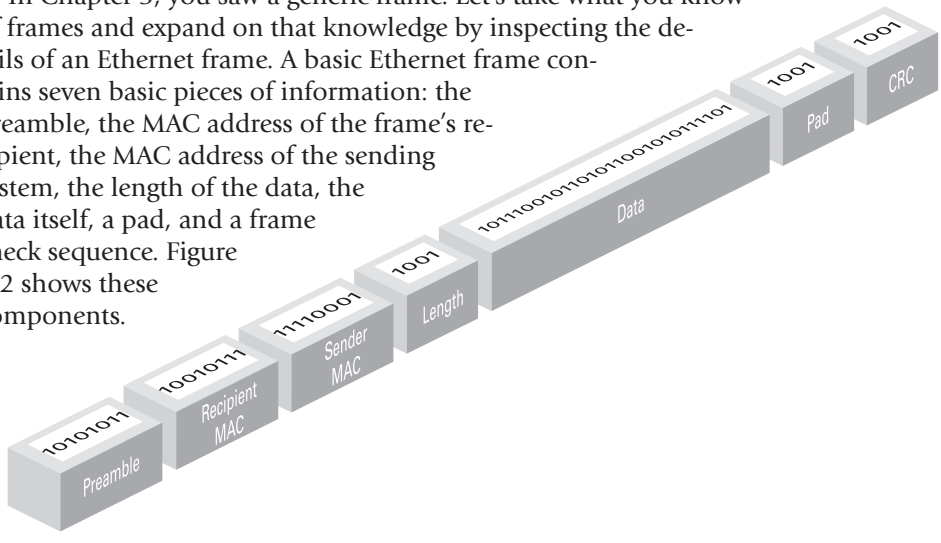

Figure 5-2 A simplified Ethernet data frame

Preamble

All Ethernet frames begin with a *preamble*, a 64-bit series of alternating ones and zeroes that ends with 11. The preamble gives a receiving NIC time to realize a frame is coming and to know exactly where the frame starts. The preamble is added by the sending NIC.

MAC Addresses

Each NIC, more commonly called a *node*, on an Ethernet network must have a unique identifying address. Ethernet identifies the NICs on a network using special 48-bit binary addresses known as *MAC addresses*.

TIP There are many situations where one computer might have two or more NICs, so one system might represent more than one node!

MAC addresses give each NIC a unique address. When a computer sends out a data frame, it transmits it to every other node across the wire in both directions, as shown Figure 5-3. All the other computers on the network listen to the wire and examine the frame

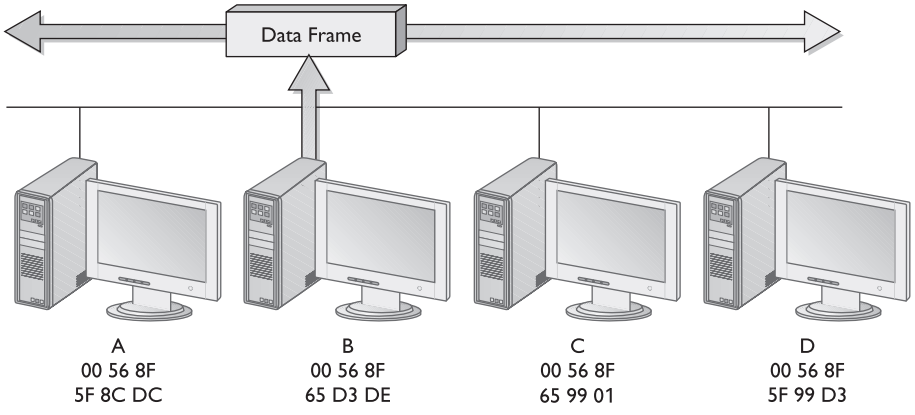

Figure 5-3 Sending out a frame

to see if it contains their MAC address. If not, they ignore the frame. If a machine sees a frame with its MAC address, it opens the frame and begins processing the data.

This system of allowing each machine to decide which frames it will process may be efficient, but because any device connected to the network cable can potentially capture any data frame transmitted across the wire, Ethernet networks carry a significant security vulnerability. Network diagnostic programs, commonly called *sniffers*, can order a NIC to run in *promiscuous mode*. When running in promiscuous mode, the NIC processes all the frames it sees on the cable, regardless of their MAC addresses. Sniffers are valuable troubleshooting tools in the right hands, but Ethernet provides no protections against their unscrupulous use.

NOTE You can find some software, such as AntiSniff, that can often detect when sniffers are in use on a network. None of the software hits close to 100 percent accuracy, but something is better than nothing. Good network administrators employ such countermeasures to stop malicious users from messing up the network.

Length

An Ethernet frame may carry up to 1500 bytes of data in a single frame, but this is only a maximum. Frames can definitely carry fewer bytes of data. The length field tells the receiving system how many bytes of data this frame is carrying.

Data

This part of the frame contains whatever data the frame carries. (If this is an IP network, it will include extra information, such as the IP addresses of both systems, sequencing numbers, and other information as well as data.)

Pad

The minimum Ethernet frame is 64 bytes in size, but not all of that has to be actual data. If an Ethernet frame has fewer than 64 bytes of data to haul, the sending NIC will automatically add extra data—a *pad*—to bring the data up to the minimum 64 bytes.

Frame Check Sequence

The *frame check sequence*—Ethernet's term for the cyclic redundancy check (CRC)—enables Ethernet nodes to recognize when bad things happen to good data. Machines on a network must be able to detect when data has been damaged in transit. To detect errors, the computers on an Ethernet network attach a special code to each frame. When creating an Ethernet frame, the sending machine runs the data through a special mathematical formula and attaches the result, the frame check sequence, to the frame. The receiving machine opens the frame, performs the same calculation, and compares its answer with the one included with the frame. If the answers do not match, the receiving machine will ask the sending machine to retransmit that frame.

At this point, those crafty network engineers have solved two of the problems facing them: they've created frames to organize the data to be sent, and put in place MAC addresses to identify machines on the network. But the challenge of determining which machine should send data at which time required another solution: CSMA/CD.

Test Specific

CSMA/CD

Ethernet networks use a system called *carrier sense, multiple access/collision detection (CSMA/CD)* to determine which computer should use a shared cable at a given moment. *Carrier sense* means that each node using the network examines the cable before sending a data frame (see Figure 5-4). If another machine is using the network, the node will detect traffic on the segment, wait a few milliseconds, and then recheck. If it detects no traffic—the more common term is to say the cable is "free"—the node will send out its frame.

Multiple access means that all machines have equal access to the wire. If the line is free, any Ethernet node may begin sending a frame. From the point of view of Ethernet, it doesn't matter what function the node is performing: it could be a desktop system running Windows XP, or a high-end file server running Windows 2003 Server or even Linux. As far as Ethernet is concerned, a node is a node is a node, and access to the cable is assigned strictly on a first-come, first-served basis.

So what happens if two machines, both listening to the cable, simultaneously decide that it is free and try to send a frame? When two computers try to use the cable simultaneously, a collision occurs, and both of the transmissions are lost (see Figure 5-5). A collision resembles the effect of two people talking at the same time: the listener hears a mixture of two voices, and can't understand either one.

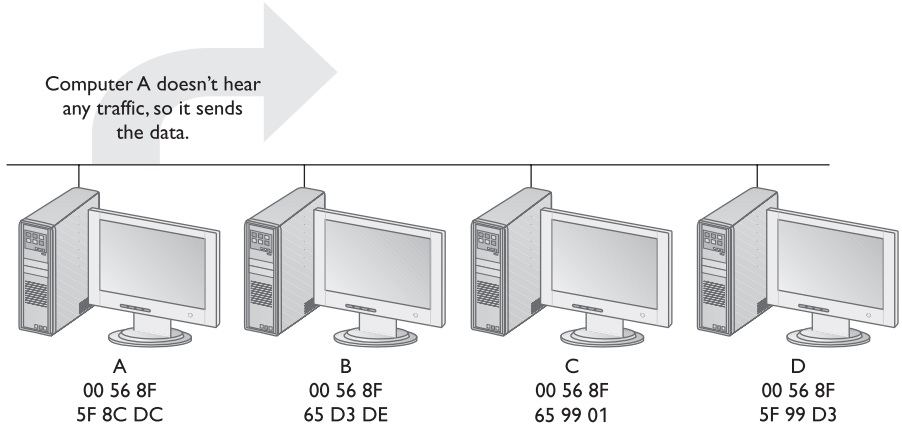

Figure 5-4 A node on an Ethernet network listens for traffic before it sends out a data frame.

Both machines will detect the fact that a collision has occurred by listening to their own transmissions. People talking on the telephone use a similar technique to know whether they are the only ones speaking at a particular moment. By comparing the words they speak with the sounds they hear, they know whether other people are talking. If a person hears words he or she didn't say, he or she knows someone else is also talking.

Ethernet nodes do the same thing. They compare their own transmission with the transmission they are receiving over the cable, and use the result to determine whether another node has transmitted at the same time (Figure 5-6). If they detect a collision, both nodes immediately stop transmitting. They then each generate a random number to determine how long to wait before trying again. If you imagine that each machine rolls its magic electronic dice and waits for that number of seconds, you wouldn't be too far from the truth, except that the amount of time an Ethernet node waits to retransmit is much shorter than one second (see Figure 5-7). Whichever node generates the lowest random number begins its retransmission first, winning the competition to use the wire. The losing node then sees traffic on the wire, and waits for the wire to be free again before attempting to retransmit its data.

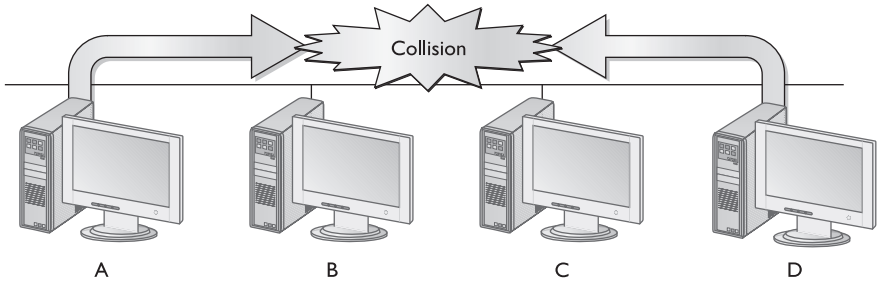

Figure 5-5 When two machines transmit simultaneously, their data frames collide.

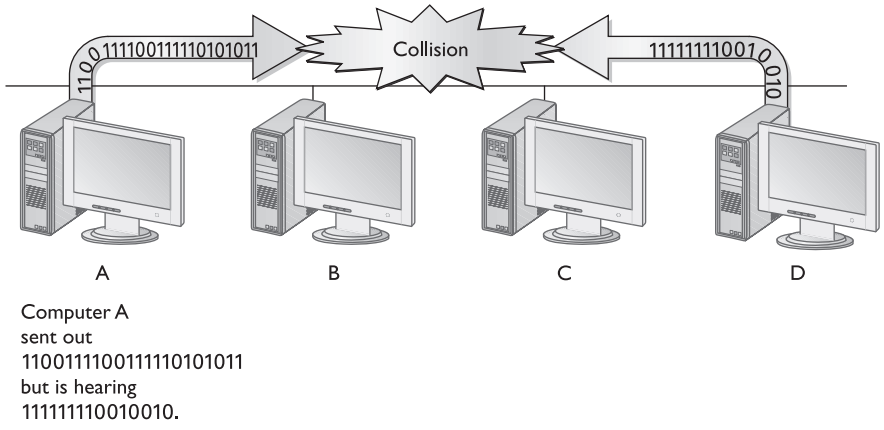

Figure 5-6 An Ethernet node detects a collision.

NOTE Because we're on the topic of collisions, a commonly used term in the Ethernet world is *collision domain*. A collision domain is a group of nodes that hear each other's traffic. A segment is certainly a collision domain, but there are ways to connect segments together to create larger collision domains. If the collision domain gets too large, you'll start running into traffic problems that manifest as general network sluggishness. That's one of the reasons to break up networks into smaller groupings. I'll discuss this in detail in Chapter 6, "Modern Ethernet."

CSMA/CD has the benefit of being simple to program into Ethernet devices, such as NIC cards. That simplicity comes at a price: an Ethernet node will waste some amount of its time dealing with collisions instead of sending data. To illustrate this waste, and the

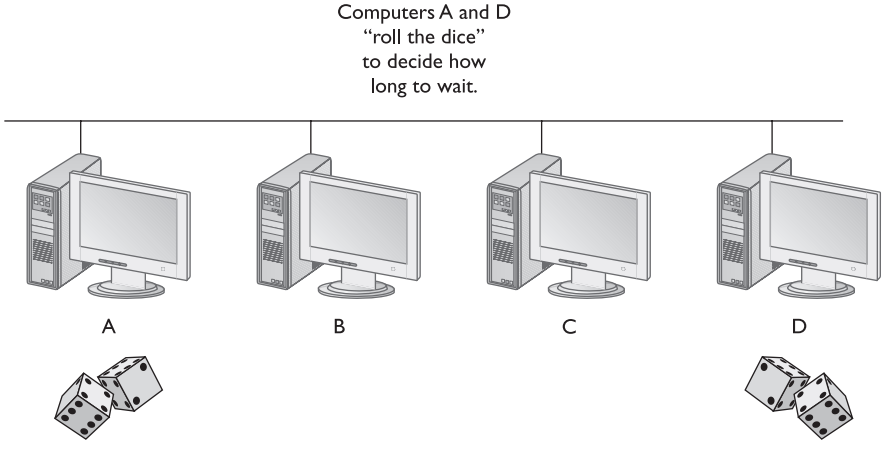

Figure 5-7 Following a collision, each node generates a random number and waits to try again.

chaos inherent to CSMA/CD, imagine a five-node network. Machines *A* and *C* both have outgoing data frames and begin the CSMA/CD process for sending traffic. They examine the cable and determine that no other node is currently sending out data (carrier sense). Because the cable is available, both *A* and *C* assume they are free to use it (multiple access). When they begin sending their respective data frames, they both detect that another station is also sending data (collision detection). Nodes *A* and *C* each generate a random number and begin counting down. Sticking with the dice analogy, assume node *A* rolls a 5 and node *C* rolls a 6. They begin counting down. 1, 2, 3, WAIT! Node *E* just started sending! Node *E* had no involvement in the original collision, and has no idea that nodes A and *C* are contending for the right to use the cable. All node *E* knows is that no device is using the cable at this moment. According to the CSMA/CD rules, *E* can begin sending. Nodes *A* and *C* have both lost out and now must wait again for the cable to be free.

The chaotic CSMA/CD method of determining access to the cable explains experiences common to users of Ethernet networks. At 9:00 on a Monday morning, 100 users sit down at approximately the same time, and type in their user names and passwords to log onto their Ethernet network. Virtually every station on the network contends for the use of the cable at the same time, causing massive collisions and attempted retransmissions. Only rarely will the end users receive any kind of error message caused by high levels of traffic. Instead, they will perceive that the network is running slowly. The Ethernet NICs will continue to retry transmission, and will eventually send the data frames successfully. Only if the collisions get so severe that a frame cannot be sent after 16 retries will the sending station give up, resulting in an error of some kind being reported to the user.

Collisions are a normal part of the operation of an Ethernet network. Every Ethernet network wastes some amount of its available bandwidth dealing with these collisions. A properly running average Ethernet network has a maximum of 10 percent collisions—for every ten frames sent, one will collide and require a resend. Collision rates greater than 10 percent often point to damaged NICs or out-of-control software.

Termination

The use of CSMA/CD in the real world has physical consequences for Ethernet networks. Most Ethernet networks use copper cabling to transmit their data frames as electrical signals. When an electrical signal travels down a copper wire, several things happen when the signal reaches the end of the wire. Some of the energy radiates out as radio waves, the cable functioning like the antennae on a radio transmitter. But some of the energy reflects off the end of the wire and travels back up the wire (see Figure 5-8).

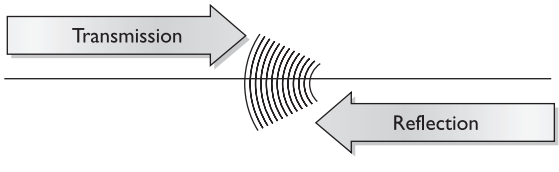

Figure 5-8
When electricity hits the end of the wire, some of the electricity comes back up the wire as a reflection.

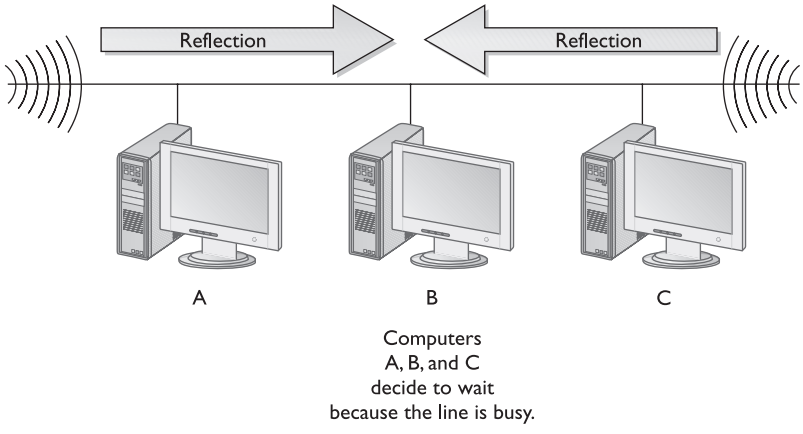

Figure 5-9 Reflections look like a busy signal to the computers attached to the network.

This reflection might make a radio run well, but it spells disaster for an Ethernet network unless we do something about the reflection. Imagine this scenario: an Ethernet card sends out a frame, which propagates along the segment, reflecting off both ends of the segment. When the other Ethernet nodes on the network attempt to send, they check the cable and misinterpret that reflection as another node sending out data frames. They wait for the reflection to dissipate before sending. The reflections quickly build up to a point that the network looks permanently busy to all of the nodes attached to it (see Figure 5-9).

To prevent these reflections, all Ethernet segments require a *terminating resistor* connected at each end (see Figure 5-10). This resistor, usually just called a *terminator*, absorbs the reflections, thereby enabling the segment to function properly. A CSMA/CD network using copper cabling won't function properly unless both ends of the network bus cable are terminated with terminating resistors.

NOTE Those of you who know something about networks might be wondering about star topologies and termination. I promise I'll cover all that in Chapter 6, "Modern Ethernet."

Figure 5-10 Two 50-Ohm terminating resistors of the type used with 10Base2 cable

Cable Breaks

The use of CSMA/CD in Ethernet networks causes some interesting behavior when the cable breaks. Figure 5-11 shows a five-node network connected to a single segment of cable. If the piece of cable between computer *A* and computer *B* breaks, computer *A* will not be able to communicate with the rest of the machines (see Figure 5-12). But that's not the end of the trouble, because a break anywhere in the bus cable causes a loss of termination in the cable. This results in reflections in both directions, prompting all the nodes on the network to go into perpetual waiting mode (see Figure 5-13), thereby shutting down the entire network.

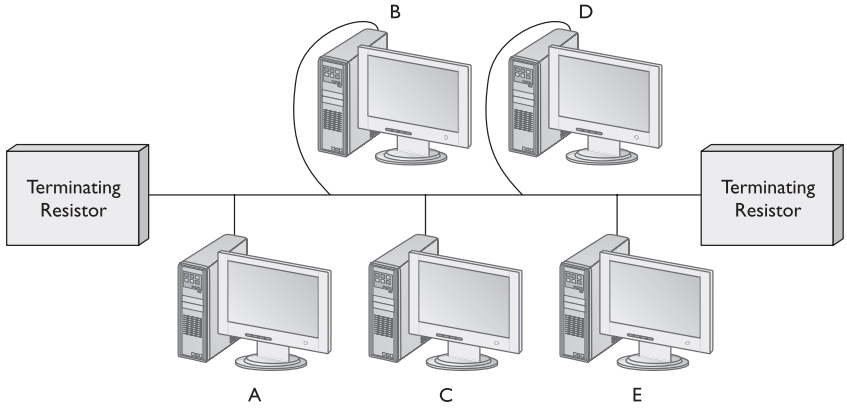

Figure 5-11 An Ethernet network with five computers

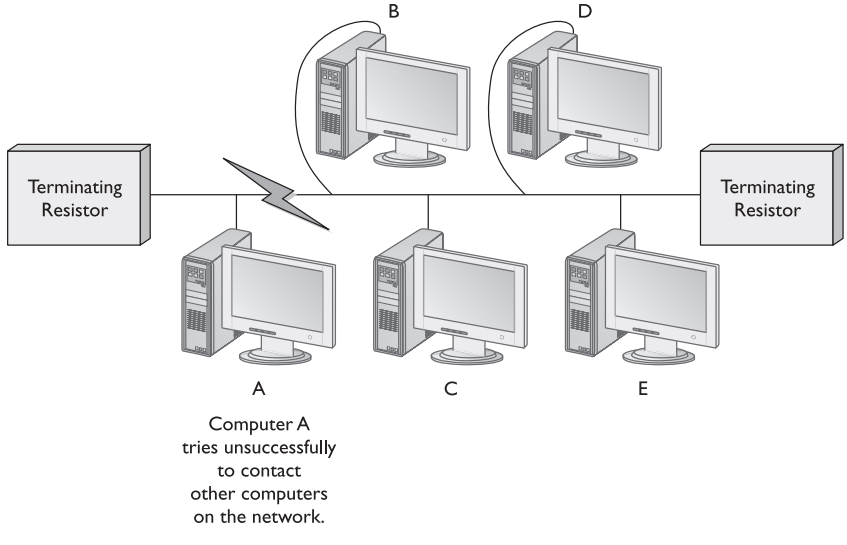

Figure 5-12 A cable break cuts computer A off from the rest of the network.

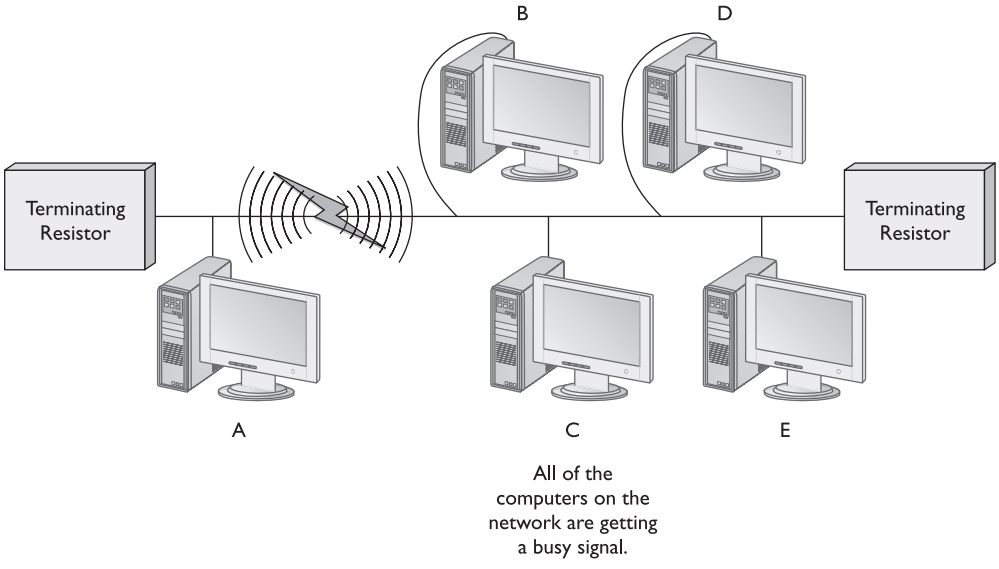

Figure 5-13 Reflections caused by the cable break bring the whole network down.

NOTE The bus cable to which computers on an Ethernet network connect is called a segment.

Now we have the answers to many of the questions that faced those early Ethernet designers. MAC addresses identify each machine on the network. CSMA/CD determines which machine should have access to the cable when. But all this remains in the realm of theory—we still need to build the thing! Numerous questions arise as we contemplate the physical network. What kind of cables should we use? What should they be made of? How long can they be? For these answers, we look to the IEEE 802.3 standard.

Ethernet Cabling Systems

The IEEE 802.3 committee recognizes that no single cabling solution can work in all situations, so it provides a variety of cabling standards, featuring cryptic names like 10Base5, 10Base2, 10BaseT, and 100BaseTX. This chapter concentrates on the Ethernet cabling systems based on coaxial cabling (10Base5 and 10Base2), while the next chapter discusses Ethernet cabling based on other cable types, such as twisted-pair (10BaseT and 100BaseTX) and fiber-optic (100BaseFX).

10Base5

In the beginning, the term "Ethernet" referred specifically to a CSMA/CD network running over a thick RG-8 coaxial cable, like the one shown in Figure 5-14. Although a specific color was not required by any standard, the cable was almost always yellow. Network techs refer to the original thick yellow cable used for Ethernet as Thick Ethernet, or Thicknet. Thicknet has the heaviest shielding of any cabling commonly used for 10-Mbps Ethernet, making it an excellent choice for high-interference environments. Because of its rigidity and typical color, the less formal among us occasionally refer to RG-8 cable as "yellow cable" or "frozen yellow garden hose."

Figure 5-14 Thick Ethernet cable (RG-8) is yellow, with a black band marking every 2.5 meters.

When the IEEE took charge of the Ethernet standard, it created a more structured way to refer to the various Ethernet cabling systems, and began referring to Thick Ethernet as *10Base5*, a term that specifies the speed of the cabling system, its signaling type, and its distance limitations. 10Base5 breaks down as follows (see Figure 5-15):

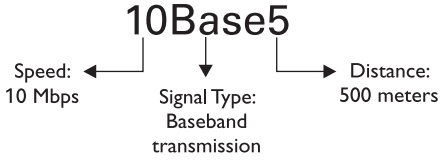

Figure 5-15 The term 10Base5 provides three key pieces of information.

- **Speed** The *10* in 10Base5 signifies an Ethernet network that runs at 10 Mbps.
- **Signal type** The *Base* in 10Base5 signifies the use of baseband signaling, meaning a single signal is on the cable.
- **Distance** The *5* in 10Base5 indicates that cables may not be longer than 500 meters.

Baseband vs. Broadband

Data signals can be sent over a network cable in two ways: *broadband* and *baseband*. Cable television is an example of broadband transmission. The single piece of coaxial cable that comes into your home carries multiple signals, and a small box enables you to select specific channels. Broadband creates these separate channels through a process called *frequency division multiplexing*. Each channel is a different frequency of signal. Your television or cable box filters out all but the frequency you want to see. Baseband is a much simpler process: it sends a single signal over the cable (see Figure 5-16). Ethernet networks use baseband signaling that employs simple *transceivers* (the devices that trans-

mit and receive signals on the cable) because they only need to distinguish among three states on the cable: one, zero, and idle. Broadband transceivers must be more complex because they have to be able to distinguish those three states on multiple channels within the same cable. Most computer networks use baseband signaling because of its relative simplicity.

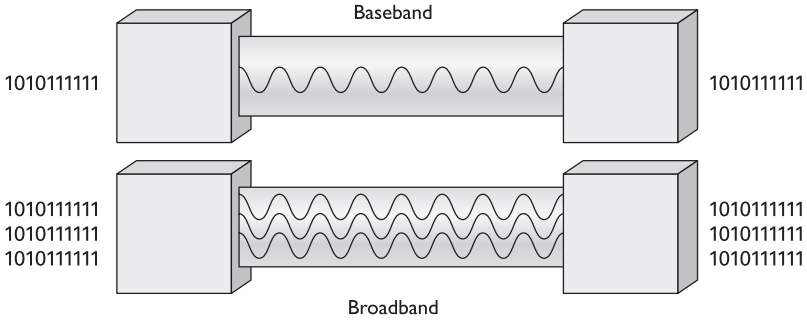

Figure 5-16 Baseband signaling sends a single signal at any given instant, whereas broadband signaling sends multiple signals on separate frequencies.

NOTE Cable modems are the only common networking devices that use broadband signaling.

Distance Limitations

10Base5 segments cannot be longer than 500 meters. A *segment* is the single length of cable to which the computers on an Ethernet network connect. The terminating resistors at each end of the segment define the ends of the segment (see Figure 5-17). The 500-meter segment limitation applies to the entire segment, not to the length of the cable between any two machines.

The distance limitations on Ethernet segments (of all sorts) provide a guideline, rather than a rigid rule for a properly functioning network. If you accidentally made a 10Base5 segment 501 meters long, the network would not suddenly cease to function or self-destruct! You would simply lower the possibility that data would get to the computers intact.

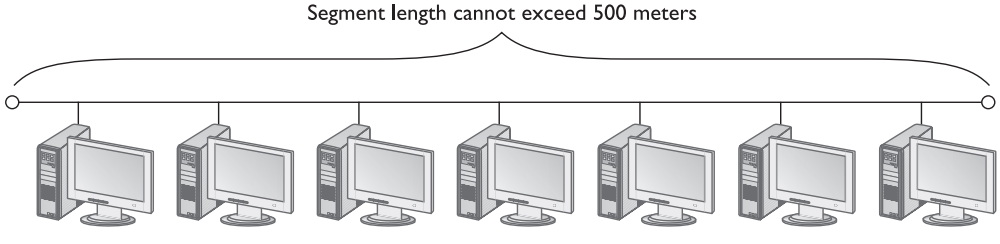

Figure 5-17 A 10Base5 Ethernet segment

The 10Base5 cabling standard strictly defines how nodes connect to the segment. Unlike nodes in many other cabling systems, 10Base5 nodes do not connect directly to the bus cable. Instead, 10Base5 NICs use a 15-pin female DB connector, called an *AUI connector*, to connect to an external transceiver (see Figure 5-18). This connector is physically identical to the MIDI and joystick connectors found on many sound cards. Confusing these connectors would not only drop the node off the network—it would also make your flight simulator game much more challenging!

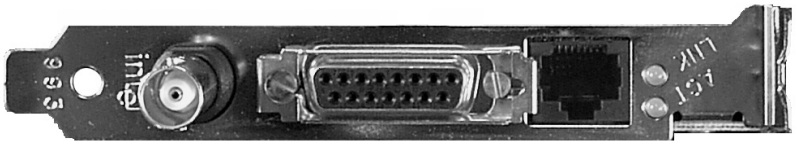

Figure 5-18 An AUI connector (center) on a 10Base5 NIC

Remember that black band on the cable in Figure 5-14? Those black bands, spaced every 2.5 meters, were created to help technicians space the connections properly when installing a network. The cable between a NIC and a transceiver can be up to 50 meters long, but the external transceivers must be placed exactly at any one of those 2.5-meter intervals along the Ethernet cable (see Figure 5-19). Figure 5-20 shows the connection between a 10Base5 transceiver and a NIC. Because 10Base5 uses an extremely stiff cable, the cables were often run through the ceiling, with drop cables used to connect the cable to the individual NICs (see Figure 5-21). A maximum of 100 nodes can be attached to each 10Base5 segment.

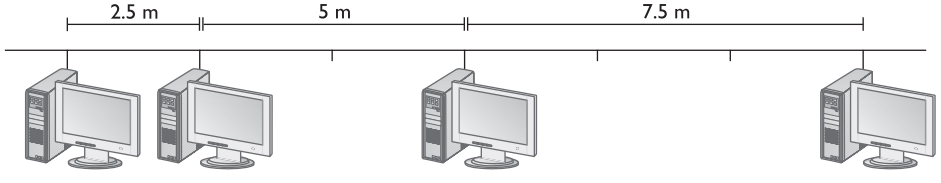

Figure 5-19 10Base5 requires that all nodes be attached at one of the 2.5-meter intervals.

Figure 5-20 A 10Base5 transceiver connected to a NIC drop cable

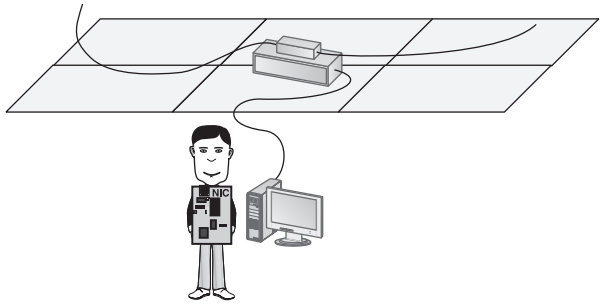

Figure 5-21 10Base5 uses drop cables to connect individual NICs to the segment, typically installed in the ceiling.

NOTE The fact that you must place every drop on one of the 2.5-meter intervals does *not* mean you must put a drop at every 2.5-meter interval. Don't get confused by this.

Goodbye 10Base5!

10Base5 is dead and gone. I'm sure that somewhere out there, there's still some 10Base5 working for a living, but with so many faster and cheaper options, 10Base5 has faded into the historical memory of the Ethernet world. After your Network+ exam, you have my permission to forget the following summary—but not until the end of the exam! Goodbye, good Thicknet. You served us well.

10Base5 Summary

- **Speed** 10 Mbps
- **Signal type** Baseband
- **Distance** 500 meters/segment
- No more than 100 nodes per segment
- Nodes must be spaced at 2.5 meter intervals
- Cables marked with a black band every 2.5 meters to ease installation
- Uses thick coaxial cable, which is almost always yellow (although nothing in the standard requires that color)
- Expensive cost per foot compared to other cabling systems
- Known as Thick Ethernet or Thicknet

10Base2

10Base2 can be used in many of the same instances as 10Base5, but it's much easier to install and much less expensive. 10Base2 uses RG-58 coaxial cable with BNC connectors, as shown in Figure 5-22. Although RG-58 cabling has less shielding than the more expensive RG-8 cabling used in 10Base5, its shielding is adequate for most installations.

Figure 5-22
A piece of RG-58 coaxial cabling with BNC connectors

The IEEE 802.3 committee tried to stay consistent with its name-signal type-distance scheme for naming Ethernet. The term 10Base2 breaks down as follows:

- **Speed** The *10* signifies an Ethernet network that runs at 10 Mbps.
- **Signal type** *Base* signifies the use of baseband signaling, meaning a single signal is on the cable.
- **Distance** The *2* indicates that cables may not be longer than 185 meters.

How does the 2 in 10Base2 translate into 185 meters? Don't ask—just live with it. Maybe at some point in the process, the distance limitation really was 200 meters and the IEEE later decided it had to be shortened. Maybe they thought 10Base1.85 looked funny and went for the closest round number. Who knows? Your job is to memorize the fact that the distance limitation for 10Base2 is 185 meters.

10Base2 has several advantages that make it the preferred choice for running Ethernet over coaxial cable. 10Base2 costs much less to install than 10Base5. RG-58 cabling costs significantly less per foot than 10Base5's RG-8 cabling. 10Base2's spacing requirements are also much less strict: computers must be spaced at least 0.5 meters apart, but they don't have to be spaced at specific intervals as required by 10Base5. RG-58's greater flexibility makes modifying and extending 10Base2 segments relatively painless. The only disadvantage is that 10Base2 allows only 30 computers per segment—far fewer than 10Base5.

Connectors

The connectors used with 10Base2 make it much easier to install and support than 10Base5. Unlike 10Base5's awkward requirement for external transceivers, 10Base2 NICs have a built-in transceiver and connect to the bus cable using a BNC connector (see Figure 5-23). The *BNC connector* provides an easy way to separate the center wire, which transmits data, from the outer shield, which protects the center wire from interference (see Figure 5-24).

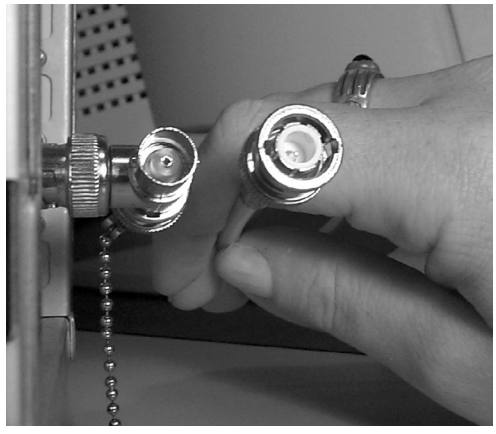

Figure 5-23 Male and female BNC connectors

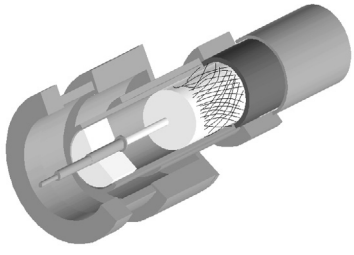

Figure 5-24 The BNC connector keeps the center wire and the shield from touching.

Traditional BNC connectors are crimped onto the wire using a crimping tool like the one shown in Figure 5-25. *Crimping* means bending the metal of the connector around the cable end to secure it to the cable. A properly crimped BNC connector keeps the center wire electrically insulated from the shield. An improperly crimped BNC connector allows the shield and the center wire to make electrical contact, creating a short in the cable (see Figure 5-26). A short, or *short circuit*, allows electricity to pass between the center wire and the shield. Because any current on the shield caused by interference will be conducted to the center strand, machines on the network will assume the network is busy and will not transmit data. The effect of a short circuit is the same as a break in the cable: the entire network goes down.

Figure 5-25 A typical crimping tool used for putting BNC connectors on a piece of RG-58 coaxial cable

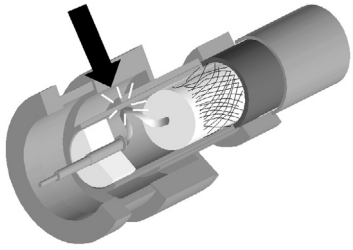

Figure 5-26
A poorly crimped cable allows electricity to pass between the shield and the center wire, creating a short.

NOTE The origins of the initials BNC have been lost. Some of the possible things it could stand for include Bayonet Nut Connector, Bayonet Navy Connector, British Naval Connector, Bayonet Neil Cofflin (purported inventor), and according to a long-time manufacturer of the devices, BNC stands for Bayonet Nut Coupling. My advice: If you can recognize a BNC connector, know what it's for, and know how to use one, then you don't need to worry about the initials!

If you find yourself in the (increasingly rare) position where you need a custom 10Base2 cable, but you have no crimping tool, all is not lost! For a quick connection on an RG-58 cable, you can dispense with the whole retro crimping scene and get a modern *twist-on* BNC connector. These convenient connectors install in seconds, and require no tools other than a pair of hands.

10Base2 requires the use of a T-connector (see Figure 5-27) when connecting devices to the cable. The stem of the *T-connector* plugs into the female connector on the Ethernet NIC, and the two pieces of coaxial cable are plugged into either end of the top bar (see Figure 5-28).

Figure 5-27
A T-connector

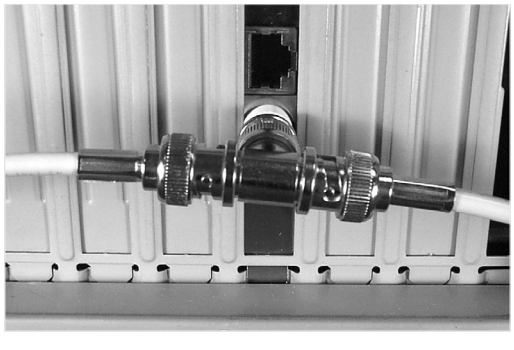

Figure 5-28
A T-connector with an RG-58 cable attached to either side

If an Ethernet node sits at the end of the cable, a terminating resistor takes the place of one of the cables (see Figure 5-29). All BNC connectors, including those on terminators and T-connectors, should be locked into place; you do this by turning their locking rings (see Figure 5-30). Although BNC connectors are basically easy to use, mistakes can happen. One frequent novice mistake is to connect a BNC connector directly to the female connection on a NIC (see Figure 5-31). While the connector locks in place just fine, the network will not function because there is no place to attach the terminating resistor.

Figure 5-29 A T-connector with a terminating resistor attached

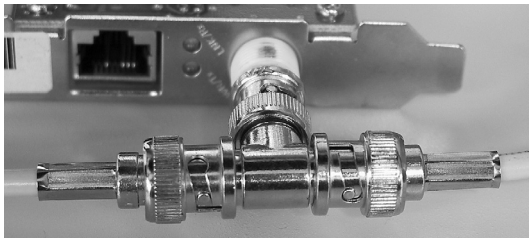

Figure 5-30 The BNC connector on the right is locked into place; the one on the left is not.

Figure 5-31 BNC connectors should never be attached directly to the NIC.

10Base2 Summary

- **Speed** 10 Mbps
- **Signal type** Baseband
- **Distance** 185 meters/segment
- No more than 30 nodes per segment
- Nodes must be spaced at least 0.5 meters apart
- RG-58 coaxial cable with BNC connectors connect to T-connectors on each node
- Nodes on the ends of the bus must have a terminator installed on one side of the T-connector
- Inexpensive cost per foot compared to 10Base5
- Known as Thin Ethernet, Thinnet, and sometimes Cheapernet

10Base2 offers a cheap and quick way to network a small number of computers using coaxial cable and Ethernet. Larger networks typically use twisted-pair wiring for Ethernet, but 10Base2 retains a strong installed base in smaller networks. 10Base2 retains the basic mechanisms of Ethernet: CSMA/CD, MAC addresses, and the Ethernet frame format. Rather than designing a new networking technology from scratch, 10Base2's designers built on proven, existing technology.

Although the network standards for Ethernet cabling lengths are commonly written in meters, I find that the distances make a lot more sense to most American students when they have the Standard English equivalents at hand. So here you go:

- 185 meters = approximately 607 feet
- 500 meters = approximately 1640 feet
- 1000 meters = approximately 6/10 of a mile

Extending the Network: Repeaters and Bridges

Some networks function perfectly well within the limitations of 10Base2 and 10Base5. For some organizations, however, the limitations of these cabling systems are unacceptable. Organizations that need longer distance limits, more computers, more fault tolerance, or the capability to combine different cabling systems can add special devices called repeaters and bridges to their networks. Let's take a look at both of these devices to see how they work.

Repeaters

A *repeater* is a device that takes all data frames it receives from one Ethernet segment and retransmits them on another segment. (Talk about truth in advertising: repeaters *repeat*!) Figure 5-32 shows a typical Ethernet repeater. A repeater takes the incoming electrical signals, translates them into binary code, and then retransmits the electrical signals. A repeater does not function as an amplifier. *Amplifiers* boost signals, flaws and all, like a

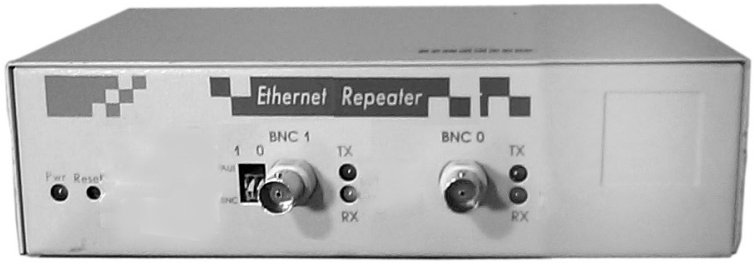

Figure 5-32 A typical Ethernet repeater

copy machine duplicating a bad original. A repeater, in contrast, re-creates the signals from scratch. Repeaters address the need for greater distances, improved fault tolerance, and integration of different Ethernet cabling systems.

NOTE Repeaters operate only at Layer 1 of the OSI model, the Physical layer (you *have* memorized these, right?).

Repeater Benefits

Repeaters have three key benefits. First, they extend the distance that a network can cover. Second, they provide a measure of fault tolerance, limiting the impact of cable breaks to the segment on which the break occurs. Third, they can link together segments using different types of Ethernet cabling.

A repeater increases the maximum possible distance between machines by linking together two segments. Each segment retains its own distance limitation. If a repeater connects two 10Base2 segments, for example, the maximum distance that can separate two machines

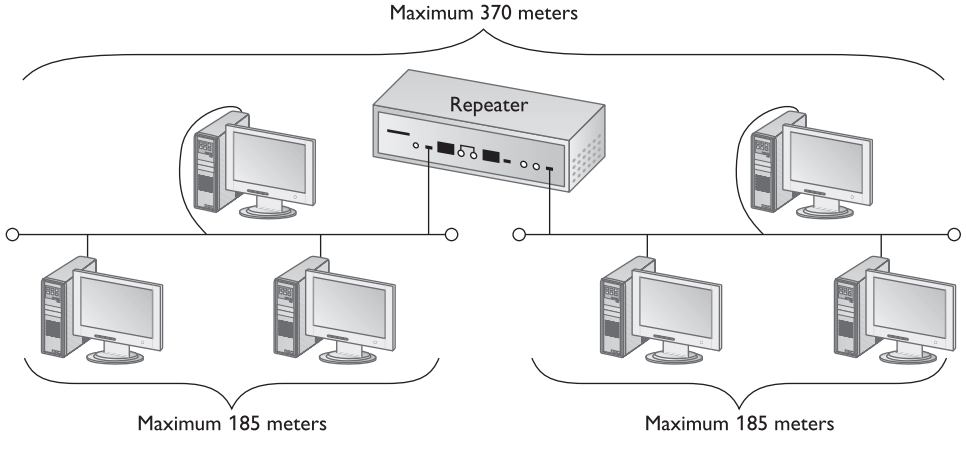

Figure 5-33 Two 10Base2 segments connected by a repeater can cover 370 meters.

on different segments is 2 × 185, or 370 meters (see Figure 5-33). Using this equation, two 10Base5 segments connected by a repeater can cover 1000 meters (2 × 500 meters).

Repeaters also add a degree of fault tolerance to a network. If one of the segments breaks, only that segment will fail. Computers on the adjacent segment will continue to function, unaffected when communicating within their own segment. The segment with the cable break fails because of reflections, but the segment on the far side of the repeater remains properly terminated and functions normally (see Figure 5-34).

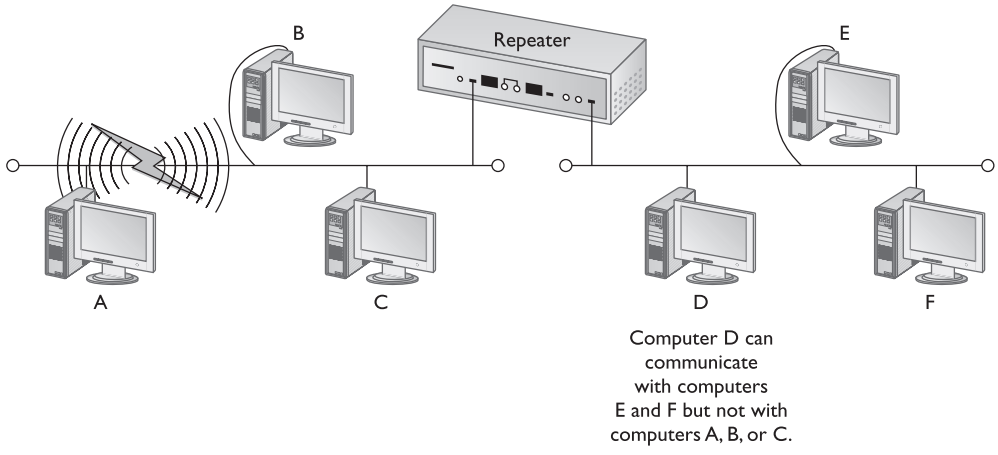

Figure 5-34 Cable breaks affect only the segment on which the break occurs.

NOTE Fault tolerance is the capability of a system to continue functioning even after some part of the system has failed.

As an added benefit, repeaters can give network designers the flexibility to combine different cabling types on the same network. Both 10Base5 and 10Base2 use exactly the same frame structure (that is, the actual ones and zeroes used are identical). Thus a repeater can connect a 10Base5 segment and a 10Base2 segment without difficulty (see Figure 5-35). Many repeaters come with both AUI and BNC connectors for that purpose (see Figure 5-36).

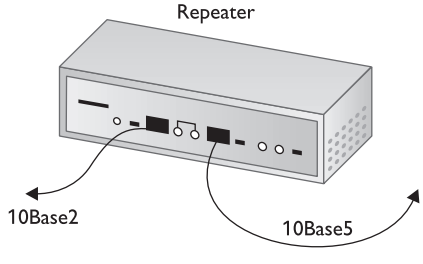

Figure 5-35 A repeater can connect Ethernet segments that use different types of cabling.

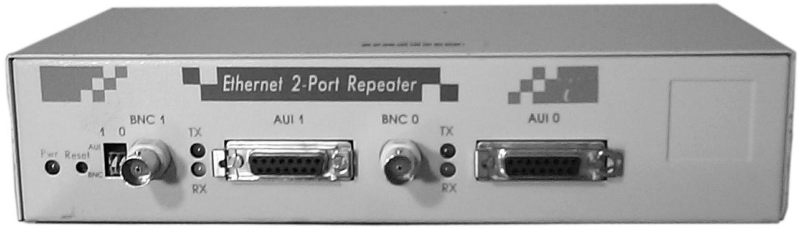

Figure 5-36 A typical Ethernet repeater with both AUI connectors for 10Base5 and BNC connectors for 10Base2

Repeaters Repeat Traffic—They Don't Manage It

Repeaters are not smart devices! They repeat every data frame they hear, regardless of its origin. Because the repeater repeats all frames that hit the wire, without regard to the source or destination, the rules of CSMA/CD apply to the entire network as a whole. If two computers on two different segments connected by a repeater both transmit a frame at the same time, a collision will result. Thus, using repeaters to build larger networks can lead to traffic jams, meaning more traffic and slower overall performance. Because all of the computers on this network hear each other and can possibly cause a collision, we call the entire network, both segments, a *single collision domain*.

In Figure 5-37, computers *A*, *B*, and *C* connect to segment 1; computers *D*, *E*, and *F* connect to segment 2. Computer *A* transmits a frame to computer *C*, which sits on the same side of the repeater. Computers *D*, *E*, and *F*, sitting on the far side of the repeater, do not need to hear the frames sent between computers *A* and *C*, but the repeater sends the frames to their network segment anyway. Machines on segment 1 cannot transmit while machines on segment 2 are using the network, and vice versa. Because all of the machines, regardless of the network segment to which they attach, can potentially have collisions with all of the other machines, segments 1 and 2 are both considered part of the same collision domain (see Figure 5-38). Even when using repeaters, an Ethernet network functions like a single CB radio channel: only one user can talk and be understood at any given time.

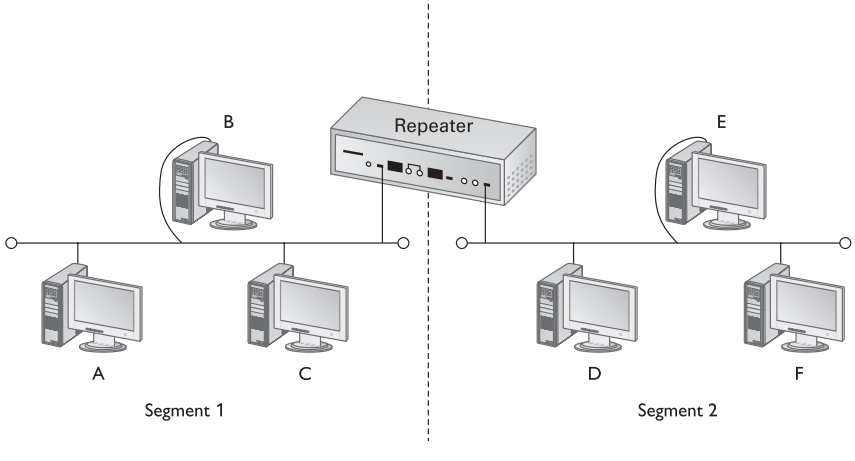

Figure 5-37 Two Ethernet segments connected by a repeater

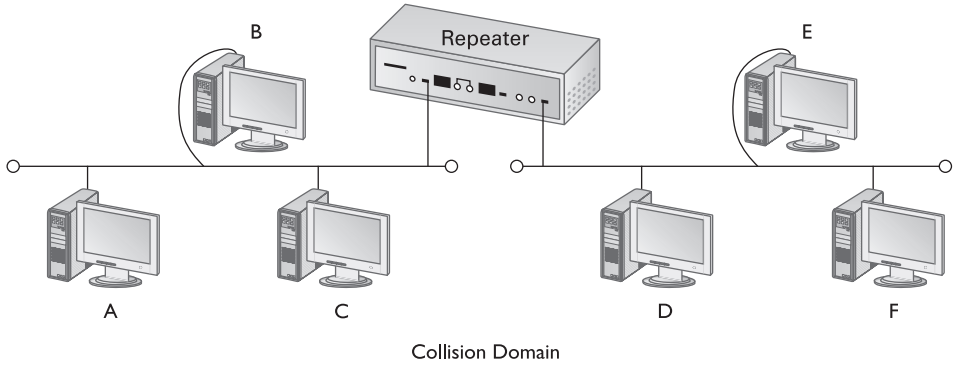

Figure 5-38 A single collision domain

NOTE A set of Ethernet segments that receive all traffic generated by any node within those segments is a collision domain. Chapter 6, "Modern Ethernet," discusses devices that can break a network into multiple collision domains.

Repeater Summary

- Repeaters increase total network cable distance.
- Repeaters provide a measure of fault tolerance.
- Repeaters can provide interoperability between different Ethernet cabling systems.
- Repeaters operate only at the Physical layer (Layer 1) of the OSI model.
- Repeaters do not help reduce or manage network traffic, but their other attributes make them important tools for network technicians and architects.

Bridges

As the demands on network bandwidth grow, the number of machines that can peacefully coexist within an Ethernet collision domain shrinks. Fortunately, a special device called a *bridge* can link together Ethernet segments to form larger networks. At first you might say, "Isn't that what repeaters do?"—but bridges do not merely connect segments. They also filter traffic between the segments, preserving precious bandwidth. Let's look at bridges to see how they accomplish this amazing feat.

Bridges filter and forward traffic between two or more networks based on the MAC addresses contained in the data frames. To *filter traffic* means to stop it from crossing from one network to the next; to *forward traffic* means to pass traffic originating on one side of the bridge to the other. Figure 5-39 shows two Ethernet segments connected by a

bridge. The bridge is represented here as a simple box, because the physical appearance of a bridge can vary a great deal. The bridge can be a stand-alone device that looks similar to an Ethernet repeater or hub, or it might be a PC with two NICs running special bridging software. The bridge might even be built into a multifunction device that provides other functions in addition to acting as a bridge. No matter how they look, all bridges do the same job: filtering and forwarding network traffic by inspecting the MAC address of every frame as it comes into the bridge.

How Bridges Work

A newly installed Ethernet bridge initially behaves exactly like a repeater, passing frames from one segment to another. Unlike a repeater, however, a bridge monitors and records the network traffic, eventually reaching a point where it can begin to filter and forward. This makes the bridge more "intelligent" than a repeater. The time for a new bridge to gather enough information to start filtering and forwarding is usually only a few seconds.

Let's watch a bridge in action. In the network shown in Figure 5-39, machine *A* sends a frame to machine *D*. When the frame destined for machine *D* hits the bridge, the bridge does not know the location of machine *D*, so it forwards the frame to segment 2. At this point, the bridge begins building a list of MAC addresses and the segment from which they came. As it forwards the packet to machine *D*, the bridge records that it received a frame from machine *A*'s MAC address from segment 1. Now that the bridge knows the location of at least one machine, it can begin filtering. Eventually, each machine will have sent out at least some frames and the bridge will have a full list of each machine's MAC address and location. For the example used here, the table would look something like Table 5-1. (A real bridge's list would not have the machine letters—those are provided only for description.)

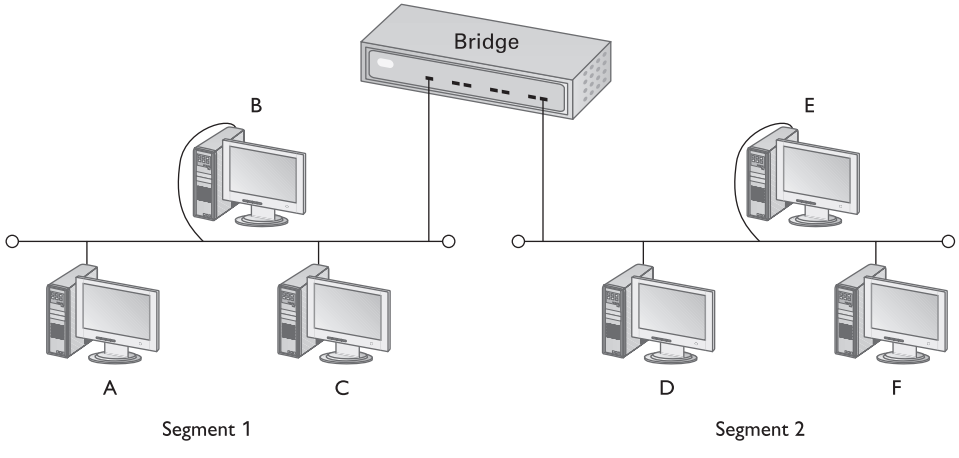

Figure 5-39 Two Ethernet segments connected by a bridge

Table 5-1 Bridge's MAC Listing

Segment 1	
Machine	**MAC address**
A	00 45 5D 32 5E 72
B	9F 16 C6 55 4D EE
C	9F 16 C6 99 DF F1
Segment 2	
Machine	**MAC address**
D	9F 16 C6 85 E5 55
E	9F 16 C6 DD 41 11
F	00 45 5D 00 25 19

Once the bridge has a complete table listing each machine's MAC address and the side of the bridge on which it sits, it looks at every incoming frame and decides whether or not to forward it to the other side. Let's see how a bridge uses this list. Let's say machine *A* decides to send another frame to machine *D*. When machine *D* responds to machine *A*, the bridge forwards the frame to segment 1 because it knows that machine *A* resides on segment 1 (Figure 5-40).

If machine *C* sends a frame to machine *A*, machine *B* will receive that frame as well because they all reside on the same segment. However, the bridge recognizes that no machine on segment 2 needs to see the frame being sent from machine *C* to machine *A* on segment 1. It filters this frame accordingly (Figure 5-41), so that the frame never makes it to segment 2.

Machines on either side of the bridge can remain blissfully unaware of the bridge's presence. When a bridge forwards a frame, it copies the frame exactly, even using the

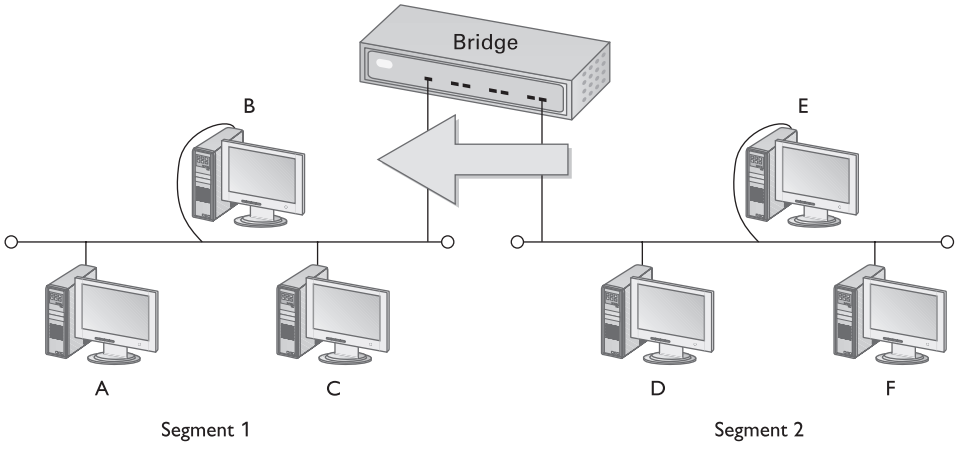

Figure 5-40 Bridge forwarding a frame

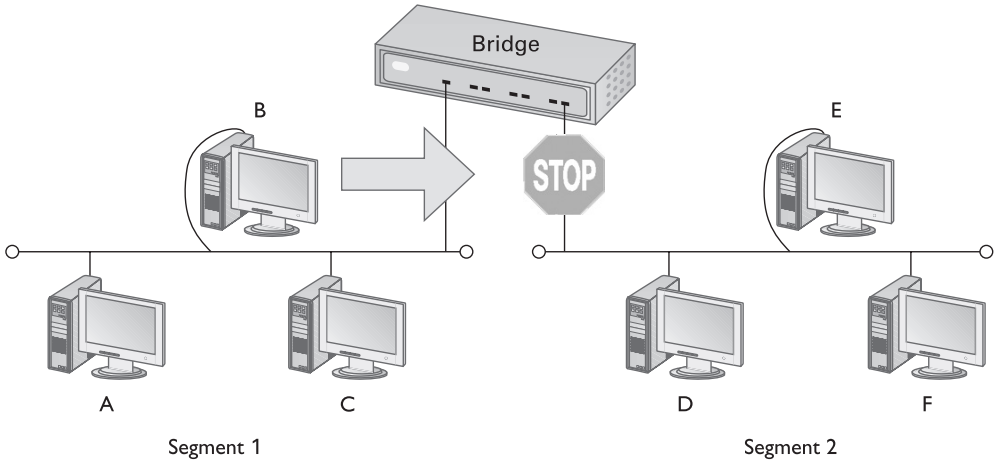

Figure 5-41 Bridge filtering a frame

originating machine's MAC address as the source MAC address in the new copy of the frame. Adding a bridge to a network does not require you to reconfigure any of the other nodes on the network. You simply rewire the cabling, and the bridge takes care of the rest.

Because bridges forward data frames without changing the frames themselves, the frame format used on each side of the bridge must be the same. The previous examples discuss bridges that connect two Ethernet networks. Bridges also exist for other technologies, such as Token Ring (see Chapter 7). Bridges cannot, however, connect an Ethernet network to a Token Ring network, because the two network technologies use totally different types of frames.

NOTE Terminology alert! To be absolutely precise, the type of bridging described here is *transparent bridging*. Some documentation, especially documentation that deals with networking theory, will refer to *translational bridges*, which can translate between different frame formats. Translational bridges rarely, if ever, appear in Ethernet or Token Ring networks. You can assume the term "bridge" refers to the transparent type, unless you are specifically told otherwise.

Bridges filter some unnecessary traffic, preserving precious network bandwidth. Bridges do have limitations, however. They cannot connect dissimilar networks and cannot take advantage of multiple routes between nodes. Overcoming these challenges requires another type of device: a router. I'll save the big router discussion for Chapter 11, "TCP/IP."

Bridge Summary

- Bridges filter or forward traffic based on the MAC addresses contained in each data frame.
- Bridges operate at the Data Link layer of the OSI model.
- Bridges can connect two networks only if they use the same type of data frames (for example, Ethernet to Ethernet, or Token Ring to Token Ring).
- Bridges learn the MAC addresses of machines on each network by listening to the cable.
- Bridges cannot be used to provide multiple routes between machines.

Chapter Review

1. Which Ethernet cabling standard is limited to 10 megabits per second (Mbps) and can support cable segments up to a maximum distance of 185 meters?
   - **A.** 10Base5
   - **B.** 10Base2
   - **C.** 10BaseT
   - **D.** 10BaseF
2. Which Ethernet cabling standard is limited to 10 Mbps and can support cable segments up to a maximum distance of 500 meters?
   - **A.** 10Base5
   - **B.** 10Base2
   - **C.** 10BaseT
   - **D.** 10BaseF
3. At which layer of the OSI model do bridges operate?
   - **A.** Physical
   - **B.** Data Link
   - **C.** Network
   - **D.** Transport
4. Which of the following requires an external transceiver?
   - **A.** 10Base5
   - **B.** 10Base2
   - **C.** 10BaseT
   - **D.** 10BaseF

5. What kind of topology does 10Base2 use?
   - **A.** Mesh
   - **B.** Bus
   - **C.** Star
   - **D.** Ring
6. What kind of cable does 10Base2 use?
   - **A.** RJ-45
   - **B.** RJ-58
   - **C.** RG-45
   - **D.** RG-58
7. When a NIC is configured to accept all incoming packets, it is said to be running in _____:
   - **A.** Master mode
   - **B.** CSMA mode
   - **C.** Promiscuous mode
   - **D.** Multimode
8. An Ethernet segment is
   - **A.** Any network connected by a bridge
   - **B.** The same thing as an Ethernet network excluding any bridges
   - **C.** The single length of cable that connects the computers on the network
   - **D.** A grouping of network nodes running at a specific OSI layer
9. The type of connector used in 10Base2 is called a(n)
   - **A.** AUI
   - **B.** RJ-45
   - **C.** BNC
   - **D.** RG-8
10. The type of connector used in 10Base5 is called a(n)
    - **A.** AUI
    - **B.** RJ-45
    - **C.** BNC
    - **D.** RG-8

Answers

1. **B.** 10Base2 is the Ethernet cabling standard that is limited to 10 Mbps and can support cable segments up to a maximum distance of 185 meters.
2. **A.** 10Base5 is the Ethernet cabling standard that is limited to 10 Mbps and can support cable segments up to a maximum distance of 500 meters.
3. **B.** Bridges operate at the Data Link layer of the OSI model.
4. **A.** 10Base5 requires an external transceiver. 10Base2 transceivers are built into the NICs.
5. **B.** 10Base2 Ethernet uses a bus topology.
6. **D.** 10Base2 uses RG-58 coaxial cable. RJ-45 is a type of connector used with unshielded twisted-pair wiring. RJ-58 and RG-45 are not common network terms.
7. **C.** A NIC that is configured to accept all incoming packets is said to be running in promiscuous mode.
8. **C.** A segment is the single length of cable that connects the computers on the network.
9. **C.** 10Base2 networks use BNC connectors.
10. **A.** 10Base5 networks use AUI connectors.