NETWORK TECHNOLOGY WITH ADMINISTRATION AND MAINTENANCE

Name : ______________________________ ID number : ____________________

Ethernet Basics

In the beginning, there were no networks. Computers were isolated - solitary islands of information.

Ethernet

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 dominates today's networks and defines all of the issues involved in transferring data between computer systems. The original Ethernet used a single piece of coaxial cable in a bus topology 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 and served as the foundation for all later versions of Ethernet.

Ethernet 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. Xerox 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 802.3 (Ethernet) committee that continues to control the Ethernet standard to this day.

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 (discussed shortly) to determine which machine should access the wire at any given time.

Topology

Every version of Ethernet invented since the early 1990s uses a hybrid star-bus topology. At the center of the network is a hub. This hub is nothing more than an electronic repeater—it interprets the ones and zeros coming in from one port and repeats the same signal out to the other connected ports. Hubs do not send the same signal back down the port that originally sent it (Figure 3.1). Repeaters are not amplifiers- they read the incoming signal and send new copies of that signal out to every connected port on the hub.

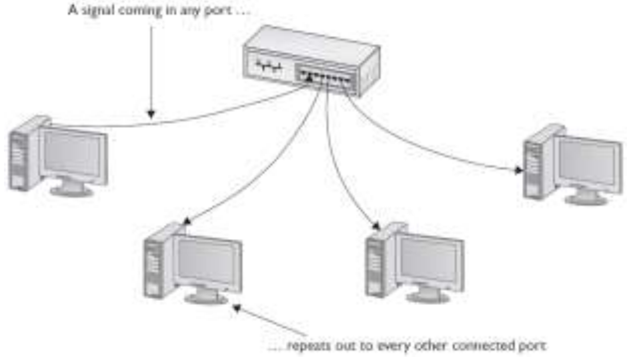

A signal coming in any port . . .
. . . repeats out to every other connected port

Figure 3.1- Ethernet hub

Ethernet Frames

All network technologies break data transmitted between computers into smaller pieces called frames. Using frames addresses two networking issues:

1. it prevents any single machine from monopolizing the shared bus cable.
2. frames make the process of retransmitting lost data more efficient.

A basic Ethernet frame contains seven 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, generically called a cyclic redundancy check (CRC). Figure 4-2 shows these components.

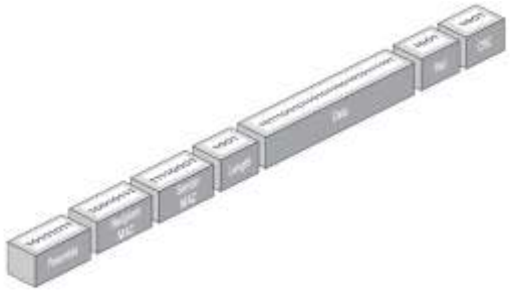


Figure 3.2 - Ethernet frame

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 (6-byte) binary addresses known as MAC addresses.

MAC addresses give each NIC a unique address. When a computer sends out a data frame, it goes into the hub that repeats an exact copy of that frame to every connected port, as shown Figure 4-3. All the other computers on the network listen to the wire and examine the frame to see if it contains their MAC address. If it does not, they ignore the frame. If a machine sees a frame with its MAC address, it opens the frame and begins processing the data.

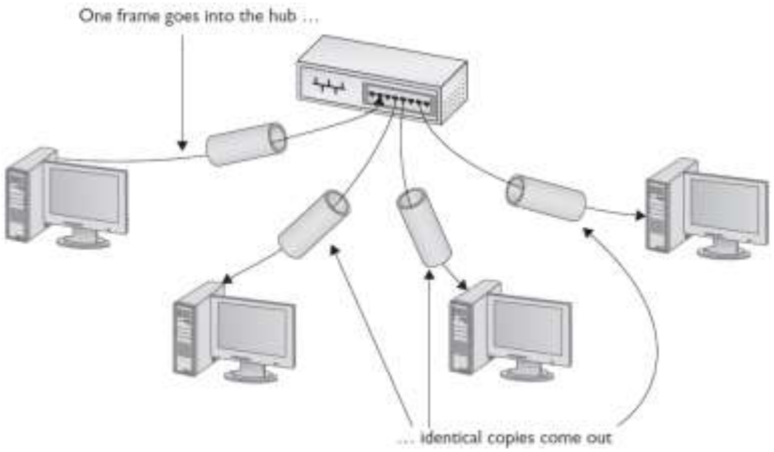

One frame goes into the hub ...
... identical copies come out

Figure 3.3 - Frames propagating on a network

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.

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

The data part of the frame contains whatever data the frame carries. (If this is an IP net-work, it will also include extra information, such as the IP addresses of both systems, sequencing numbers, and other information.)

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—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 asks 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.

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 (Figure 3.4). If another machine is using the network, the node detects traffic on the segment, waits a few milliseconds, and then rechecks. If it detects no traffic—the more common term is to say the cable is "free"—the node sends out its frame.

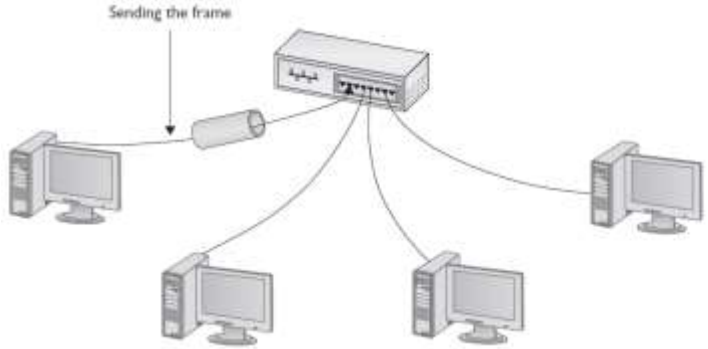


Figure 3.4 - No one else is talking—send the frame!

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 Server 2008 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? A collision occurs, and both of the transmissions are lost (Figure 3.5). 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.

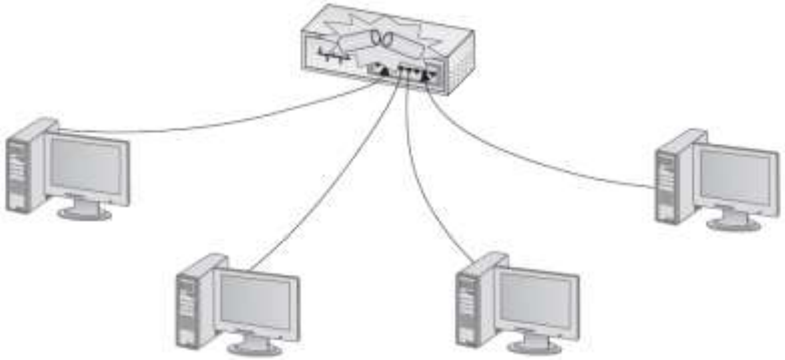


Figure 3.5 - Collision

It's easy for NICs to notice a collision. Collisions create nonstandard voltages that tell each NIC another node has transmitted at the same time. 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 (Figure 3.6). 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.

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 20 frames sent, approximately 2 frames will collide and require a resend. Collision rates greater than 10 percent often point to damaged NICs or out-of-control software.

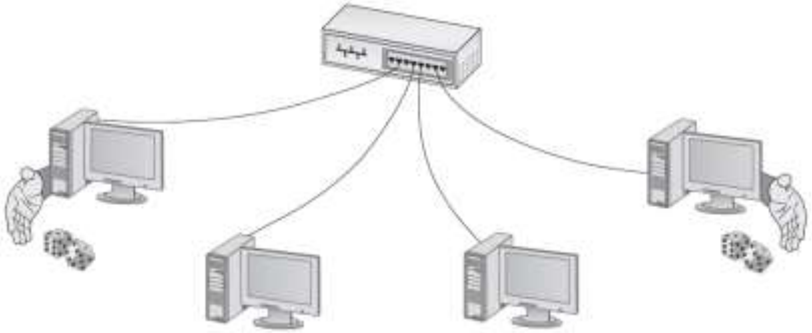


Figure 3.6 Rolling for timing

Defining Ethernet

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. This becomes important to remember as you learn about the implementation of Ethernet over time.

Early Ethernet Networks

MAC addresses identify each machine on the network. CSMA/ CD determines which machine should have access to the cable and when. Contemplating the physical network brings up numerous questions:

- What kind of cables should be used?
- What should they be made of?
- How long can they be?

For these answers, turn to the IEEE 802.3 standard and two early implementations of Ethernet: 10BaseT and 10BaseFL.

10BaseT

In 1990, the IEEE 802.3 committee created a new version of Ethernet called 10BaseT to modernize the first generations of Ethernet. Very quickly 10BaseT became the most popular network technology in the world, replacing competing and now long gone competitors with names like Token Ring and AppleTalk. Over 99 percent of all networks use 10BaseT or one of its faster, newer, but very similar versions. The classic 10BaseT network consists of two or more computers connected to a central hub. The NICs connect with wires as specified by the 802.3 committee.

10BaseT hubs come in a variety of shapes and sizes to support different sizes of networks. The biggest differentiator between hubs is the number of ports (connections) that a single hub provides. A small hub might have only 4 ports, while a hub for a large network might have 48 ports.

Regardless of size, all 10BaseT hubs need electrical power. Larger hubs will take power directly from a power outlet, while smaller hubs often come with an AC adapter. In either case, if the hub loses power, the entire segment will stop working.

The name 10BaseT follows roughly the naming convention used for earlier Ethernet cabling systems. The number 10 refers to the speed: 10 Mbps. The word Base refers to the signaling type: baseband. The letter T refers to the type of cable used: twisted-pair. 10BaseT uses UTP cabling.

UTP

Officially, 10BaseT requires the use of CAT 3 (or higher), two-pair, unshielded twisted pair (UTP) cable. One pair of wires sends data to the hub while the other pair receives data from the hub. Even though 10BaseT only requires two-pair cabling, everyone installs four-pair cabling to connect devices to the hub as insurance against the possible requirements of newer types of networking. Most UTP cables come with stranded Kevlar fibers to give the cable added strength, which in turn enables installers to pull on the cable without excessive risk of literally ripping it apart.

10BaseT also introduced the networking world to the RJ-45 connector. Each pin on the RJ-45 connects to a single wire inside the cable; this enables devices to put voltage on the individual wires within the cable. The pins on the RJ-45 are numbered from 1 to 8. The 10BaseT standard designates some of these numbered wires for specific purposes. As mentioned earlier, although the cable has four pairs, 10BaseT uses only two of the pairs. 10BaseT devices use pins 1 and 2 to send data, and pins 3 and 6 to receive data. Even though one pair of wires sends data and another receives data, a 10BaseT device cannot send and receive simultaneously.

The rules of CSMA/CD still apply: only one device can use the segment contained in the hub without causing a collision. Later versions of Ethernet will change this rule.

An RJ-45 connector is usually called a crimp, and the act of installing a crimp onto the end of a piece of UTP cable is called crimping. The tool used to secure a crimp onto the end of a cable is a crimper. Each wire inside a UTP cable must connect to the proper pin inside the crimp. Manufacturers color-code each wire within a piece of four-pair UTP to assist in properly matching the ends. Each pair of wires consists of a solid-colored wire and a striped wire: blue/blue-white, orange/orange-white, brown/brown-white, and green/green-white.

The Telecommunications Industry Association/Electronics Industries Alliance (TIA/EIA) defines the industry standard for correct crimping of four-pair UTP for 10BaseT networks. Two standards currently exist: TIA/EIA 568A and TIA/EIA 568B. Following an established color-code scheme, such as TIA/EIA 568A, ensures that the wires match up correctly at each end of the cable.

10BaseT Limits and Specifications

Like any other Ethernet cabling system, 10BaseT has limitations, both on cable distance and on the number of computers. The key distance limitation for 10BaseT is the distance between the hub and the computer. The twisted-pair cable connecting a computer to the hub may not exceed 100 meters in length. A 10BaseT hub can connect no more than 1024 computers, although that limitation rarely comes into play.

10BaseT Summary:

- Speed 10 Mbps
- Signal type Baseband
- Distance 100 meters between the hub and the node
- Node Limit No more than 1024 nodes per hub
- Topology Star-bus topology: physical star, logical bus
- Cable type Uses CAT 3 or better UTP cabling with RJ-45 connectors

10BaseFL

Just a few years after the introduction of 10BaseT, a fiber-optic version appeared, called 10BaseFL. As you know from the previous chapter, fiber-optic cabling transmits data packets using pulses of light instead of using electrical current. Using light instead of electricity addresses the three key weaknesses of copper cabling. First, optical signals can travel much farther. The maximum length for a 10BaseFL cable is up to two kilometers, depending how it is configured. Second, fiber-optic cable is immune to electrical interference, making it an ideal choice for high-interference environments. Third, the cable is much more difficult to tap into, making it a good choice for environments with security concerns. 10BaseFL uses multimode fiber-optic and employs either an SC or an ST connector.

All fiber-optic networks use at least two fiber-optic cables. While 10BaseFL enjoyed some popularity for a number of years, most networks today are using the same fiber-optic cabling to run far faster network technologies.

10BaseFL Summary

- Speed 10 Mbps
- Signal type Baseband
- Distance 2000 meters between the hub and the node
- Node limit No more than 1024 nodes per hub
- Topology Star-bus topology: physical star, logical bus
- Cable type Uses multimode fiber-optic cabling with ST or SC connectors

Extending and Enhancing Ethernet Networks

Once you have an Ethernet network in place, you can extend or enhance that network in several ways. You can install additional hubs to connect multiple local area networks. A network bridge can connect two Ethernet segments, effectively doubling the size of a collision domain. You can also replace the hubs with better devices to reduce collisions.

Connecting Ethernet Segments

Sometimes, one hub is just not enough. Once an organization uses every port on its existing hub, adding additional nodes requires additional hubs or a device called a bridge. Even fault tolerance can motivate an organization to add more hubs. If every node on the network connects to the same hub, that hub becomes a single point of failure—if it fails, everybody drops off the network. There are two ways to connect hubs: an uplink port or a crossover cable. You can also connect Ethernet segments using a bridge.

Uplink Ports

Uplink ports enable you to connect two hubs together using a straight-through cable. They're always clearly marked on the hub, as shown in Figure 3.7. To connect two hubs, insert one end of a cable to the uplink and the other cable to any one of the regular ports. To connect more than two hubs, you must daisy-chain your hubs by using one uplink port and one regular port. Figure 3.8 shows properly daisy-chained hubs. As a rule, you cannot daisy-chain more than four hubs together.

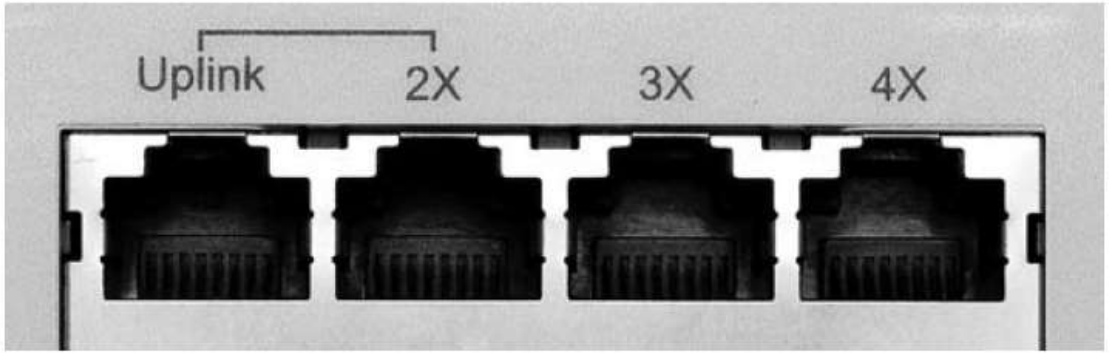


Figure 3.7 - Typical uplink port

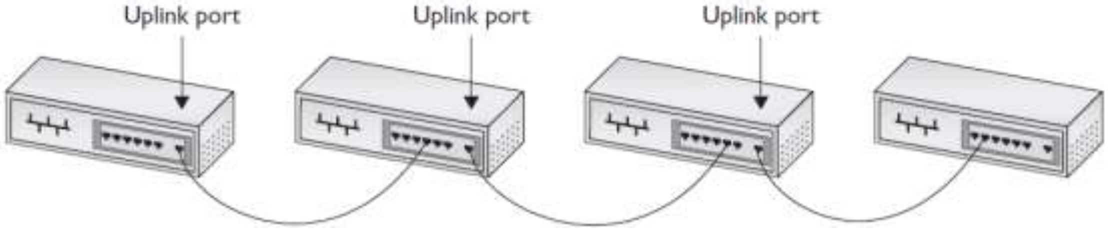


Figure 3.8 - Daisy-chained hubs

Working with uplink ports is sometimes tricky, so you need to take your time. It's easy to mess up and use a central hub. Hub makers give their uplink ports many different names, such as crossover, MDI-X, and OUT. There are also tricks to using uplink ports. See the line connecting the uplink port and the port labeled 2X? You may use only one of those two ports, not both at the same time. Additionally, some hubs place on one of the ports a switch that you can press to make it either a regular port or an uplink port (Figure 3.9). Pressing the button electronically reverses the wires inside the hub. Be sure to press the button so it works the way you want it to work.

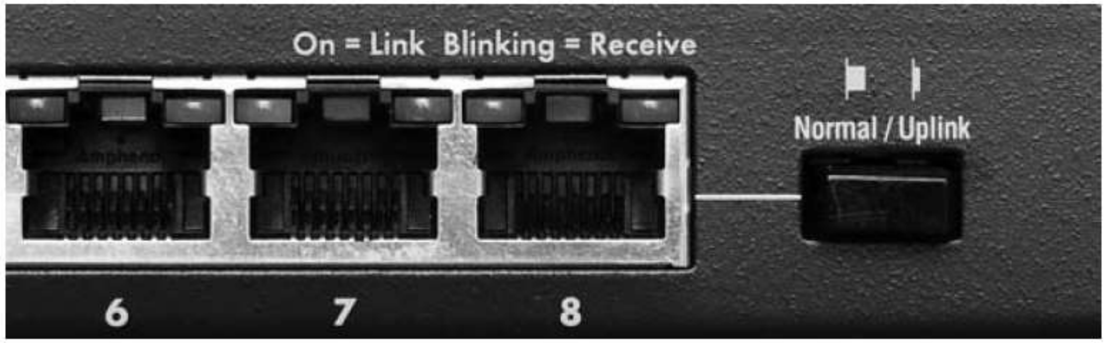


Figure 3.9 - Press-button port

When connecting hubs, remember the following:

- Only daisy-chain hubs.
- Take time to figure out the uplink ports.
- If you plug hubs in incorrectly, no damage will occur—they just won't work.

Crossover Cables

Hubs can also connect to each other via special twisted-pair cables called crossover cables. A standard cable cannot be used to connect two hubs without using an uplink port, because both hubs will attempt to send data on the second pair of wires (3 and 6) and will listen for data on the first pair (1 and 2). A crossover cable reverses the sending and receiving pairs on one end of the cable. One end of the cable is wired according to the TIA/EIA 568A standard, while the other end is wired according to the TIA/EIA 568B standard. With the sending and receiving pairs reversed, the hubs can hear each other; hence the need for two standards for connecting RJ-45 jacks to UTP cables.

A crossover cable connects to a regular port on each hub. Keep in mind that you can still daisy-chain even when you use crossover cables. Interestingly, many hubs, especially higher-end hubs, do not come with any uplink ports at all. In these cases your only option is to use a crossover cable.

We can use a crossover cable to connect two computers together using 10BaseT NICs with no hub between them at all. This is handy for the quickie connection needed for a nice little home network, or when you absolutely, positively must chase down a friend in a computer game.

Be careful about confusing crossover cables with uplink ports. First, never connect two hubs by their uplink ports. Take a regular cable; connect one end to the uplink port on one hub and the other end to any regular port on the other hub. Second, if you use a crossover cable, just plug each end into any handy regular port on each hub.

If you mess up your crossover connections, you won't cause any damage, but the connection will not work. If you take a straight cable (that is, not a crossover cable) and try to connect two PCs directly, it won't work. Both PCs will try to use the same send and receive wires. When you plug the two PCs into a hub, the hub electronically crosses the data wires, so one NIC sends and the other can receive. If you plug a second hub to the first hub using regular ports, you essentially cross the cross and create a straight connection again between the two PCs.

Bridges

The popularity and rapid implementation of Ethernet networks demanded solutions or workarounds for the limitations inherent in the technology. An Ethernet segment could only be so long and connect a certain number of computers. What if a network went beyond those limitations?

A bridge acts like a repeater or hub to connect two Ethernet segments, but it goes one step beyond—filtering and forwarding traffic between those segments based on the MAC addresses of the computers on those segments. This preserves precious bandwidth and makes a larger Ethernet network possible. 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.

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. A new bridge usually requires only a few seconds to gather enough information to start filtering and forwarding.

Although bridges offer a good solution for connecting two segments and reducing bandwidth usage, these days you'll mainly find bridges used in wireless, rather than wired networksMost networks instead have turned to a different magic box, a switch, to extend and enhance an Ethernet network.

Switched Ethernet

10BaseT Ethernet performed well enough for first-generation networks, by the early 1990s networks used more demanding applications, such as Lotus Notes, SAP business management software, and Microsoft Exchange, which quickly saturated a 10BaseT network.

In a classic 10BaseT network, the hub, being nothing more than a multiport repeater, sends all packets out on all ports. While this works well, when you get a busy network with multiple conversations taking place at the same time, you lose speed. The problem with hubs is that the total speed of the network—the bandwidth—is 10 Mbps. To appreciate the problem with hubs, take a look at the two computers sending data in Figure 3.10.

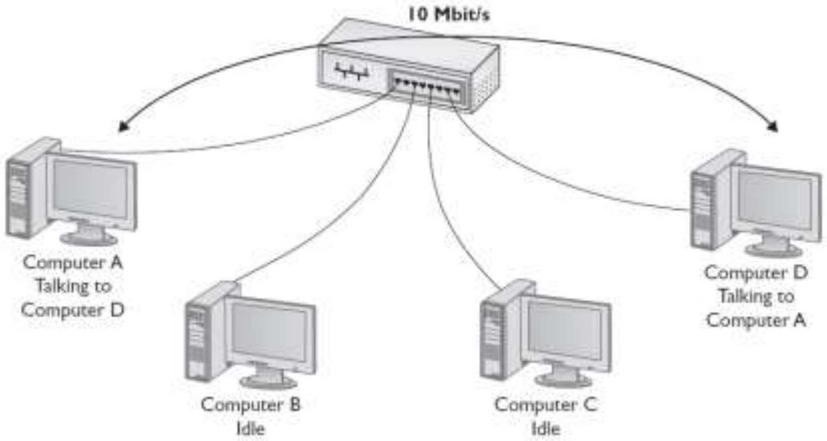


Figure 3.10 - One conversation gets all the bandwidth.

Since only one conversation is taking place, the connection speed between Computer A and Computer D runs at 10 Mbps. But what happens if Computer B and Computer C wish to talk at the same time? Well, CSMA/CD kicks in and each conversation runs at only ~5 Mbps (Figure 3.11).

Imagine a network with 100 computers, all talking at the same time! The speed of each conversation would deteriorate to a few hundred thousand bits per second, way too slow to get work done.

An Ethernet switch looks and acts like a hub, but comes with extra smarts that enable it to take advantage of MAC addresses, creating point-to-point connections between two conversing computers. This effectively gives every conversation between two computers the full bandwidth of the network.
To see a switch in action, check out Figure 3.12.. When you first turn on a switch, it acts exactly as though it were a hub, passing all incoming frames right back out to all the other ports. As it forwards all frames, however, the switch copies the source MAC addresses and quickly (usually in less than one second) creates an electronic table of the MAC addresses of each connected computer.

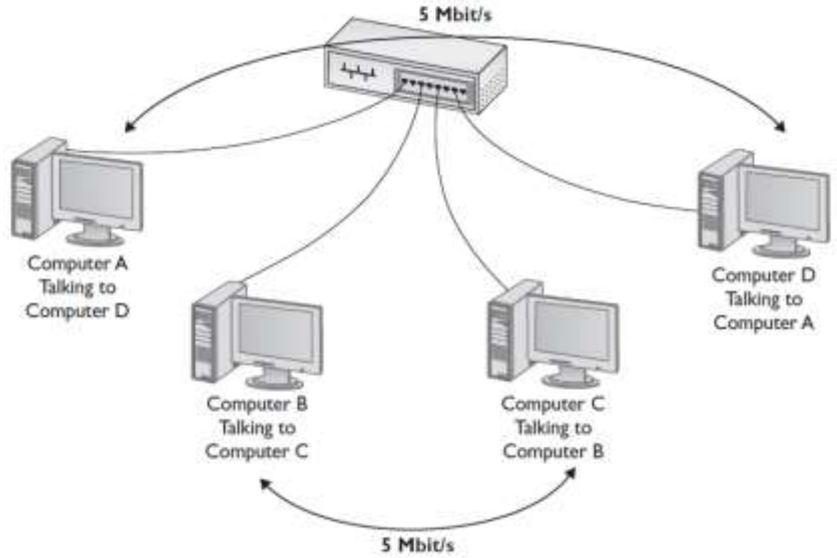
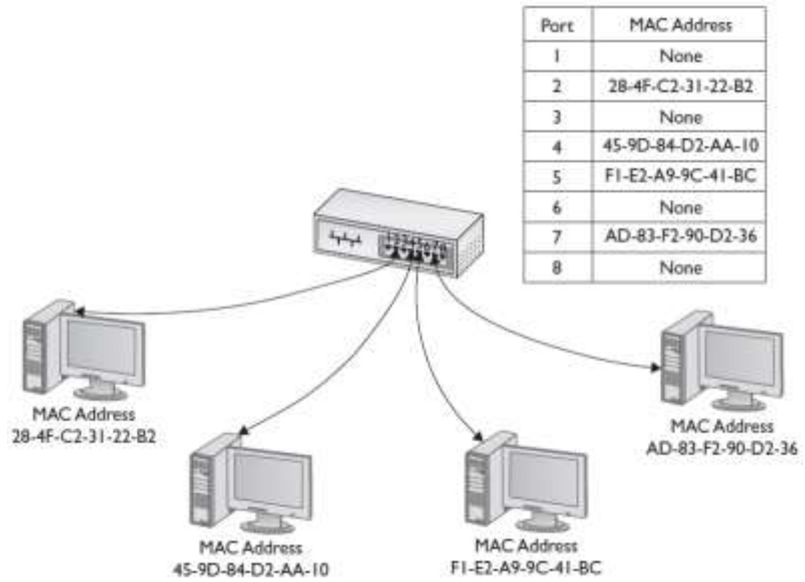


Figure 3.11- Two conversations must share the bandwidth.

As soon as this table is created, the switch begins to do something amazing. The moment it detects two computers talking to each other, the switch starts to act like a telephone operator, creating an on-the-fly, hard-wired connection between the two devices. While these two devices communicate, it's as though they are the only two computers on the network. Figure 3.13 shows this in action. Since each conversation is on its own connection, each runs at 10 Mbps.



Port	MAC Address
1	None
2	28-4F-C2-31-22-B2
3	None
4	45-9D-84-D2-AA-10
5	F1-E2-A9-9C-41-BC
6	None
7	AD-83-F2-90-D2-36
8	None

Figure 3.12- A switch tracking MAC addresses

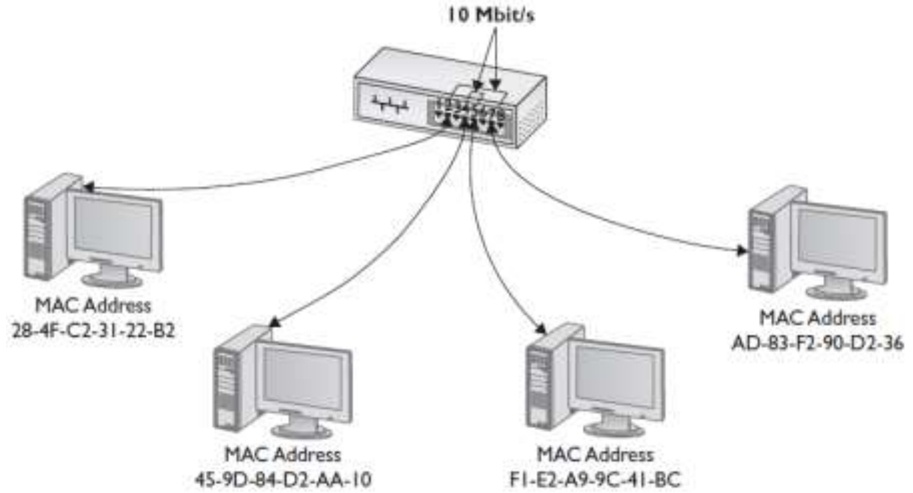


Figure 3.13 A switch making two separate connections

Speed isn't the only benefit switches bring to a 10BaseT network. When you use switches instead of hubs, the entire CSMA/CD game goes out the window. Forget about daisy-chain only. Feel free to connect your switches pretty much any way you wish (Figure 4-24).

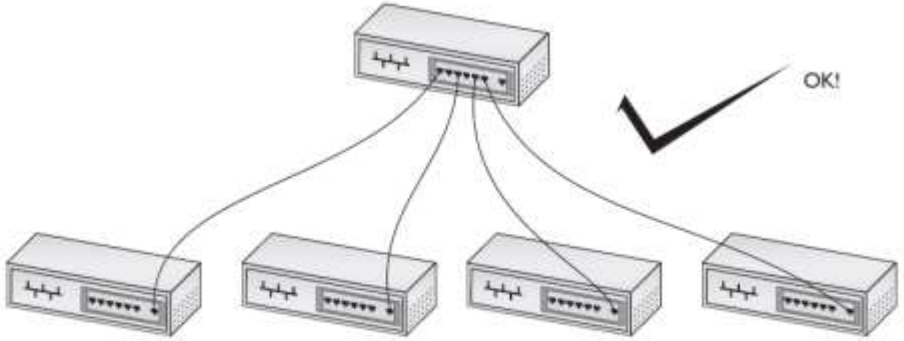


Figure 3.14 - Switches are very commonly connected in a tree organization.

Physically, an Ethernet switch looks much like an Ethernet hub (Figure 3.15). Logically, because the switch creates a point-to-point connection between any two computers, eliminating CSMA/CD, the entire concept of collision domain disappears because there are no longer any collisions. Instead, the common term used today is broadcast domain, because all devices connected to a switch will hear a broadcast sent from any one system.

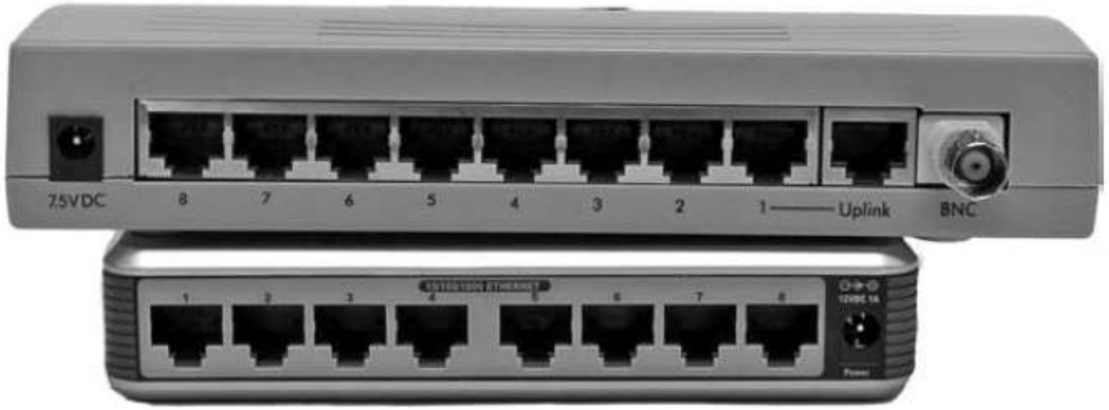


Figure 3.15 - Hub (top) and switch (bottom) comparison

Spanning Tree Protocol

The ease of interconnecting switches makes them prone to a nefarious little problem called bridge loops. As its name implies, a bridge loop is nothing more than an interconnection of switches in such a fashion that they create a loop. In the network shown in Figure 3.15, for example, packets going between switches A, B, and C have multiple paths. This creates a problem.

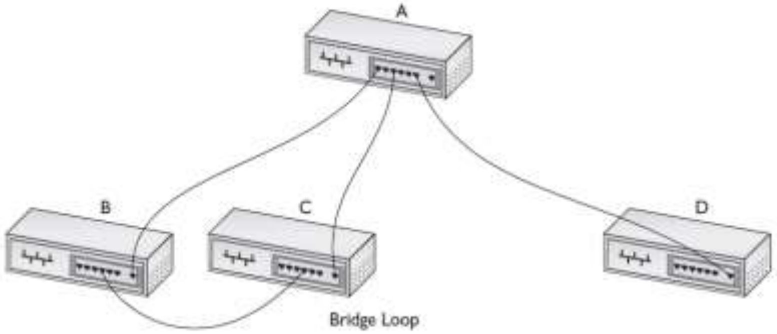


Figure 3.15 - Bridge loops are bad!

A bridge loop using the first generations of Ethernet switches was a very bad thing, creating a path sending packets in an endless loop and preventing the network from working. To prevent this, the Ethernet standards body adopted the Spanning Tree Protocol (STP). STP adds a little more intelligence to switches that enables them to detect bridge loops. If detected, the switches communicate with each other and, without any outside interaction, turn off one port on the loop (Figure 3.16).

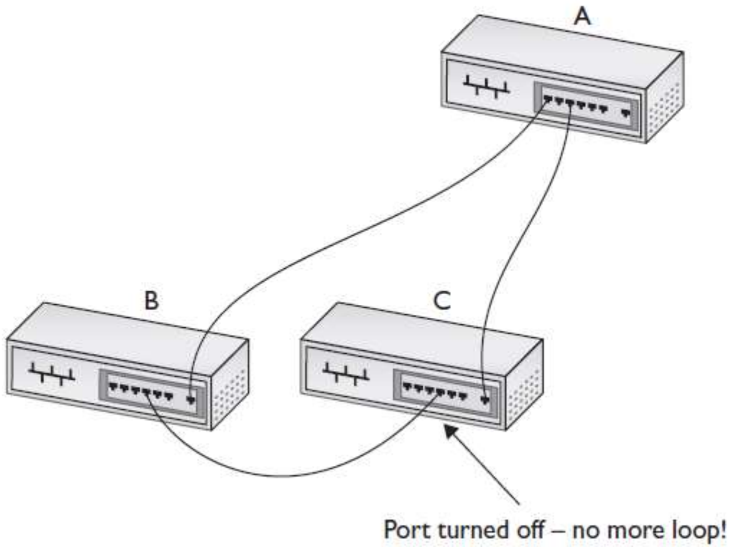


Figure 3.16 - Port turned off, disaster averted