

Chapter 19

Network Fundamentals

The next few chapters will concentrate on one of the hot communications subjects of today — digital networks and data transmission. In order to understand the subject — and especially the relationships between various techniques and methods — we need to understand the terminology, and also to put it all into context.

Since we are introducing a new topic, we have many new words to learn. In a sense, this chapter will be a glossary of terms, but we will group words and concepts by their similarity, rather than alphabetically.

Networks

Fig. 19-1(a) shows a simple connection between two devices. The devices could be computers, terminals, switchboards, etc.; for simplicity, they are simply called *nodes*. The connections between them are often called *links*. It is a *point-to-point* connection because it is a direct connection between two points or nodes. Because there is only one connection between the two nodes, it is called *singly-connected*.

Fig. 19-1(a) is too simple to really be called a network — a network requires more than two nodes and/or more than one link. For example, the *multiply-connected* connection in Fig. 19-1(b) would qualify as a network, though a very simple one.

A *network* is thus a set of connections between a number of nodes.

A *public* network is one that can be used by various customers, and is generally owned by one or more organizations called *common carriers*. A *private* network, on the other hand, is generally used by just one customer, and usually also owned by that customer. (There are exceptions, of course — for example, a company could own a private network, but let its customers or suppliers also use it, or it could route a part of its private network over someone else's facilities.)

There are also *networks-of-networks*. One example is the telephone network, which is a network which consists of other networks, connected together. For example, a long-distance call from New York to California may utilize the network of a Local Exchange Carrier

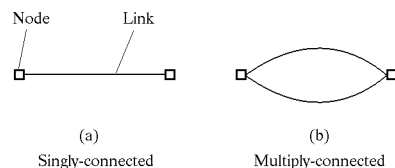


Fig. 19-1. Point-to-point network topology

(LEC) in New York, the network of an Interexchange Carrier (IXC) who carries the call from the east coast to the west coast, and the network of another LEC in California.

Another example of a network of networks is the *Internet*. Typically, large organizations (businesses, universities, research institutes) maintain individual private networks, which are tied together through a *backbone* network so they can communicate with each other. The entire system, including both the individual networks as well as the backbone connecting them, is then called the Internet. Initially, use of the Internet was limited to the large organizations and their staff; nowadays, the public can access it through ISPs or Internet Service Providers, who connect their own networks to the backbone.

Besides public and private networks, there are also *virtual private networks* or VPNs. To the user, a VPN looks and feels like a private network — it is a private connection, not accessible to the general public — but it is routed through a public network.

Network connections can also be permanent or switched. A *permanent* connection is one that is always connected. For example, consider a company having two offices A and B at opposite ends of town, with each office having its own telephone switchboard. To let people at A speak with people at B, the company could (for a monthly fee) rent a *tie-line* from the telephone company to permanently connect the two switchboards together. But if the connection is only needed a few times a day, it would be cheaper to place the calls by dialing the number and using the telephone company's switching network. This would then be a *switched* connection.

Networks can also be characterized by their size. For example, in digital networks there are LANs, WANs, and MANs.

The LAN is a *Local Area Network*. This is a digital network which is generally completely owned by one person or company, and usually limited to just one building.

A typical WAN or *Wide Area Network* would extend to more than one building. WANs often consist of two or more LANs, interconnected to each other but otherwise relatively independent. The connection between them might involve circuitry owned by the same owner as the LANs, or might be leased from another carrier.

A MAN or *Metropolitan Area Network*, on the other hand, extends over a wider area, such as a city. Like a WAN, it too usually consists of interconnected LANs, but here the interconnection is almost always through a common carrier.

Another difference between LANs, WANs, and MANs is their speed. The local nature of a LAN means

that high-speed links are feasible and cost-effective, so LANs tend to operate at very high speeds — 100 megabits per second is not unusual.

A MAN, on the other hand, requires leasing a long connection from another carrier, and here the cost increases with speed and distance. Hence MANs tend to operate at much slower speeds. (There are exceptions, of course.)

Topology

Mathematicians use the word *topology* to describe how points in an object are connected to each other. In communications, we use the same word to describe the connections in a network. It is easiest to explain with some examples.

Fig. 19-1 shows point-to-point connections between two nodes, singly-connected at (a) and multiply-connected at (b). Of more interest, however, are the connections between multiple nodes, as in Fig. 19-2.

Fully-connected network

Fig. 19-2(a) shows six nodes in a *fully-connected* network. Each of the six nodes is connected to each of the other five nodes. This sort of connection can be feasible if there are just a few nodes, but it quickly grows out of sight when the number of nodes is large, because with n nodes we need

$$\frac{n \times (n - 1)}{2}$$

links. For example, with the six nodes shown, we need

$$\frac{6 \times 5}{2} = 15 \text{ links;}$$

with 100 nodes, we would need

$$\frac{100 \times 99}{2} = 4950 \text{ links.}$$

This rapidly becomes unwieldy and expensive.

In most cases, the nodes need to communicate with each other in both directions. Thus each of the links in the fully-connected network needs to be bidirectional. In terms of wiring, this can be done with either two one-

directional cables, or with one two-directional cable. This leads us to three new terms:

A *simplex* connection communicates in just one direction.

A *half-duplex* connection sends data in both directions, but only one way at a time.

A *full-duplex* connection can send data in both directions at the *same* time.

Bus Network

Fig. 19-2(b) shows a *bus* connection. The heavy horizontal line is a common connection, called the bus, which runs through the premises. Each of the nodes connects to the bus at some point along its length.

Each of the nodes can be either a data source or a receiver, and data might go from any node to any other node. There is generally only one signal path in the bus, however, so it must be able to carry data in both directions. But this means that data can only go one way at a time, so the bus is half-duplex.

This points out one disadvantage of the bus over the fully-connected scheme — except for *broadcasts* (where one node sends data to all other nodes at the same time), only two nodes can communicate at any one time on a bus. In a fully connected network, all connections are separate, and so multiple pairs of nodes could communicate at the same time.

With so many sources (i.e., nodes) being able to feed the bus, there must be some way to avoid conflicts by preventing several nodes from sending data to the bus at the same time; we will discuss this in a few pages.

The bus can take many forms:

- Inside a computer, a data bus or address bus is simply a set of short, parallel conductors (usually copper paths on a printed circuit board) that carry parallel data back and forth between the central processor, memory, and input/output devices.
- In longer-distance systems, the bus might be a balanced wire pair. For example, some auto salvage yards share a voice communications bus system where a junk yard needing a specific part can “broadcast” a request which is heard at all other yards on the system.
- In LANs, the bus might be a coax cable running throughout the building and connecting the various computers on the system.

As we discussed in Chapter 4, the end of a transmission line will reflect signals if it is not properly terminated. This could be a problem in a bus, with the reflected pulses interfering with other data. Thus the bus has to be terminated; this is shown in Fig. 19-2(b) by putting a *terminator* at each end of the bus.

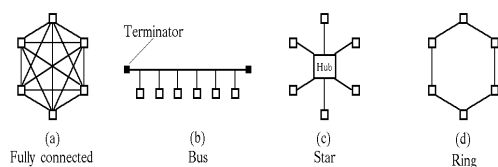


Fig. 19-2. Four common network topologies

A bus connection was fairly common in older local area networks, and there were two common methods used, called 10Base5 and 10Base2. The term “10Base” was used because both systems ran at 10 megabits per second, and because the digital pulses were sent as “baseband” signals — that is, they were not modulated onto a carrier, but were sent directly. The 5 and the 2 at the end described the type of cable and connection. 10Base5 used a thicker coaxial cable and allowed a bus length of 500 meters; 10Base2 used a thinner cable with a maximum length of about 200 meters (actually, 185 meters).

Fig. 19-3 shows how a typical coax bus connected to the NIC or *network interface card* in a 10Base2 system. The NIC was installed in the computer, and had a BNC connector on its back. A BNC “T” connector was plugged into the NIC card, and the coax cable (shown at the bottom right in the photograph) connected to the “T”. In this photo, the computer would be at the end of the bus, so a BNC terminator (containing simply a resistor) would plug into the other side of the “T”; otherwise, a second coax connector would plug in there and continue the bus to the next computer.

The bus topology has a problem if there is a break somewhere in the bus connection. At first glance, you would think that a break somewhere along the line would simply break the system into two parts, with all of the computers in each part still being able to communicate together, but no communication between the two parts. This turns out not to be so. If there is a break in the bus, the two halves lose their terminations on one end. Hence there are now reflections, and the entire sys-

tem stops working. The bus connection is therefore no longer popular for LANs (beside the fact that newer, faster LANs do not support coax cable and BNC connectors at all.)

Star Network

The nodes in Fig. 19-2(c) use a *star connection*. There is now a central connection point, and all the nodes connect to this central point.

How the network works depends on how much “smarts” is in the center of the star. Here are some examples:

- In the telephone network, the center point would be a switch, a fairly intelligent device. A node that wants to communicate with another node simply asks the switch to put through a direct connection from one to the other. The switch can handle many such simultaneous connections, so there can be many conversations going on at the same time.
- Or the central point could be a computer, which acts as the controller for a network of relatively dumb nodes. For example, a central computer could be interfaced to a number of cash registers. The computer sends commands out to the cash registers, and the registers only respond to these commands.
- In a LAN, on the other hand, the central point might be a dumb device called a *hub*. The simplest hub takes any input from any node, and sends it out to all the other nodes at the same time. Since all signals are mixed, only one node should send out data at a time.

Modern LANs use unshielded twisted pair wire (UTP), rather than coax cable, and use an RJ-45 connector as shown in Fig. 19-4. It has room for eight connections and usually uses eight-wire cable, but only four wires are actually used — two to send data toward the hub, and two to send data back out from the hub. The wiring can support full-duplex operation, but many older networks still use half-duplex.

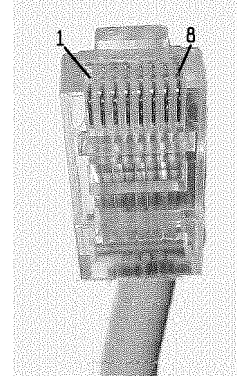


Fig. 19-4. RJ-45 connector

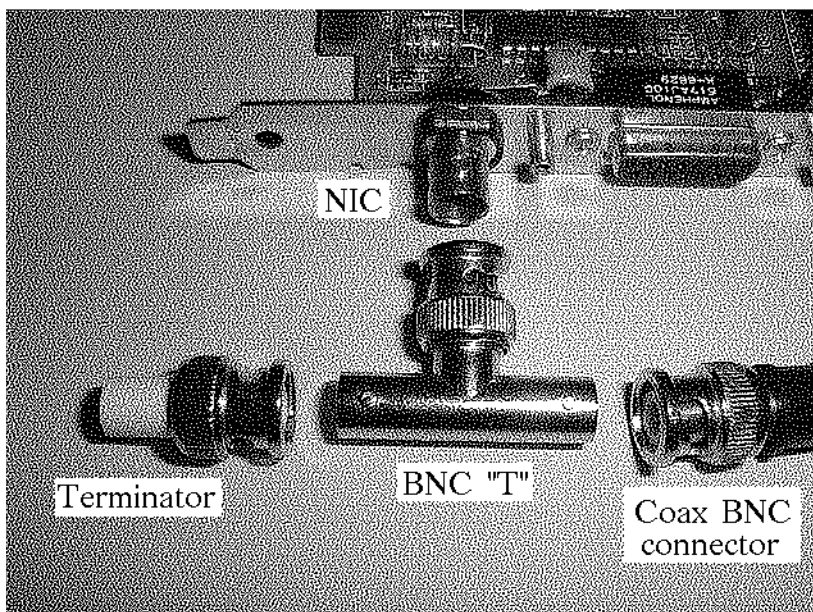


Fig. 19-3. 10Base2 coax bus termination at a NIC

The half-duplex hub generally contains no switching; it is just a distribution system. It takes all of the inputs, combines them together into one data stream, and sends it out to all of the outputs.

Even though the star connection is physically very different from a bus connection, logically it still works the same — all of the nodes on the system can hear each other all the time. As with the bus, if two or more nodes output data at the same time, there will be problems. But the advantage of the star over the bus is that a break somewhere in the network will only affect the one node that it connects to, not the entire network.

Ring Network

Still another way to connect a number of nodes is with a ring network, shown in Fig. 19-2(d). In this configuration, adjacent nodes can still talk to each other directly, but nodes farther away have to use intermediate nodes for relaying data.

In its simplest form, the connection between nodes might only be one-way. For example, data may flow around the node in only a clockwise direction or only counterclockwise. But a cable break anywhere obviously kills the entire network, so there are two common solutions:

One is to provide for two-way links between nodes. In case of a break, data can go around the ring in the opposite direction.

A second solution is to rewire the ring as shown in Fig. 19-5(a). It now *looks* like a star, but is really still a ring. The difference is that the connections between nodes, instead of going directly from one node to the next, loop through a central box called a MAU or *Multi-station Access Unit*.

The MAU is really the secret of the system. In addition to the digital data, each node also sends a DC voltage to the MAU; this DC voltage tells the MAU whether the node is operating and properly connected, or not. If the MAU fails to get this DC voltage (because of a cable break, for example), it simply bypasses that node and takes it out of the ring, as shown in Fig. 19-5(b). But the rest of the ring still stays connected.

This system has two advantages — not only does it allow the ring to continue working if a wire is cut, but it

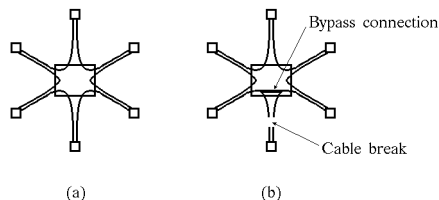


Fig. 19-5. Another way to wire a ring

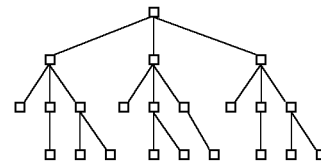


Fig. 19-6. Hierarchical or tree network

also allows the nodes to be turned off or otherwise taken out of service. This would be perfectly OK in a bus or normal star connection, but would break the ring if it were connected as in Fig. 19-2(d).

More Complex Network Topologies

When there is a relatively small number of nodes, any of the network topologies shown so far is adequate. But when there are many nodes, then slightly different connections may be used for either physical reasons (to simplify wiring, for example) or logical reasons (to improve performance.)

Some network configurations are simply combinations of the basic types from Fig. 19-2. For example, there might be several star networks, all connected together via a bus. Alternatively, Fig. 19-6 shows a hierarchical or tree topology. There are other possible combinations.

Network Interconnections

Most network connections have a limit on their length. Because of signal loss (attenuation) and signal distortion (dispersion and noise), there is a limit on how far a signal can travel before it is too weak or too distorted to be reliably recognized by the receiver. This is a problem in both analog and digital networks, and is usually solved by putting some sort of an amplifier into the line.

In an analog circuit, that amplifier is called a *repeater*. It amplifies the signal, but it unfortunately also amplifies any noise or distortion that has been added to the desired signal during transmission. Thus each time the signal is amplified on a long trip, it gets worse and worse.

A simple repeater could also be used in a digital circuit, but a better solution is to use a *regenerator* (unfortunately, regenerators are also often called repeaters, and so the terminology can be confusing.) The regenerator receives incoming digital data, analyzes it to recognize the ONES and ZEROS in the signal, and then generates a new digital signal which duplicates the bits, but without the noise or distortion, and with the timing properly restored. With proper design and placement, the regenerator cleans up the signal and makes it “as good as new”. The digital signal can thus travel huge

distances without in any way becoming degraded as it goes along.

Modern digital networks, however, often need more than just a repeater/regenerator, and there is a variety of more sophisticated equipment that fits the bill:

- A *bridge* is a regenerator, but one that can connect different types of networks together. For example, it could connect a star network to a ring. It essentially transmits everything from one network to the other, but it may make necessary changes in timing or organization to make it match the new network.
- A *router* is an intelligent device that can do the work of a bridge, but it also decides on the route that a digital message takes. As an example, consider a company with a LAN in the sales department, and another LAN in the engineering department, and a router connecting the two LANs together. Messages between salesmen don't need to go to the engineering LAN, and vice versa, whereas messages from a salesman to an engineer do. The router looks at the addresses in the messages and decides where and how (and whether) to route them through.

In some ways, the router could be called a *store-and-forward* node. It receives an incoming message, temporarily stores it, possibly acknowledges receipt to the sender, analyzes the address, decides where and how to forward it, and sends it on to the next recipient. It may also check for errors and/or take part in error correction.

Collisions and Timing

In a half-duplex bus or star network, only one message can obviously travel at a time. (But even in more complex full-duplex or ring networks, sending messages into the network at the right time is essential.) If two different messages appear on the same circuit at the same time, we get a *collision*.

You can think of a data bus as a one-lane two-way highway where only one car is allowed at a time because two cars will cause a collision. Just as on a highway, there are three possible ways to avoid a problem:

- Use a *bus controller* — a sort of traffic cop or traffic light, which controls who can send data to the bus and when, *or*
- Put a stop sign at each entrance, where anyone who wants to enter the highway must first stop and look whether there is another car already there, *or*
- Use a round-robin approach, where a node must wait until its turn.

All three of these methods are used in networks of various kinds.

Bus Controller

The simplest example of the bus controller approach is the data bus inside a simple computer system. Here the processor controls access to the data bus by sending out control signals to the various devices in the system. These signals often carry names like MEMORY READ, MEMORY WRITE or I/O READ. This is practical in a computer system, but it needs extra wiring and so would not be practical for a larger network.

A more practical network example is the case of a single large computer connected, most likely by a bus, to a number of cash registers. The central computer sends commands to the cash registers, and they respond to commands. “Don't speak until spoken to” would be a good description for the system. In a case like this, there can be no collisions.

Carrier Sense Multiple Access (CSMA)

Consider a number of devices all feeding a single bus connection — “Multiple Access” to one bus. Each device monitors the bus, and will only send data to it if it sees that the bus is free and there is no data on it from someone else.

If the data on the bus is modulated onto a carrier, then each device is trying to “Sense a Carrier” before itself transmitting. This explains the letters CS in CSMA, but actually the same abbreviation is used even if the bus carries baseband data — i.e., data that is put on the bus in bare form as pulses, without a carrier. Either way, the device will only send data to the bus if the bus is free.

The problem is that occasionally two (or more) devices will listen at the same time, both decide that the bus is free, and then both start to transmit at the same time. This will lead to a *collision*. In pure CSMA, neither device will know that there is a collision, and so will finish transmitting its message as if nothing happened. This might occur, for instance, in a simple wireless system, where a device can either receive or transmit, but not both at the same time. Once it starts to transmit, it has no way of knowing that some other device is transmitting also.

CSMA networks usually depend on the receiver acknowledging receipt of the message. If there is a collision, that acknowledgement never arrives. Each sender then eventually gets the idea that perhaps the message didn't get through, and so sends it again.

CSMA-CD — CSMA With Collision Detection

On a typical wired bus network, each sender can listen to the bus as well as transmit. It can therefore detect a collision if it sees that the data on the bus is either gar-

bled, or simply different from what it sent. It can now take corrective steps immediately.

When it detects a collision, each device proceeds like this:

- (a) Stop transmitting,
- (b) Wait a random time interval,
- (c) If the bus is free, try again; otherwise wait for it to become free.

The reason for the random wait is to prevent an immediate second collision. Suppose two devices collide. If each immediately tried again, they would collide again. Even if each waited a while and then tried, they would still collide if they waited the same amount of time. By waiting a random time, one of them is likely to retry sooner than the other, thus avoiding another collision.

CSMA-CD is used in many networks; the most common LAN type — Ethernet — uses it. When the amount of traffic is low, it allows almost instantaneous access to the network for any device. But when there is a lot of traffic to be carried, collisions occur so often that the network rapidly becomes overwhelmed. For example, many LAN hubs have an LED indicator which lights when the hub detects a collision. Under heavy usage, that LED may be lit almost continuously, and the actual bit-per-second rate may be just a fraction of the theoretical network capacity.

Token Ring

Although it uses the word “ring”, a token ring network is not necessarily wired in a ring topology. Rather, the term *token ring* describes how the various sources share the network.

Each time a new node is added to a token ring network, all the nodes go through a discovery process, during which they check who is connected, and assign themselves a number. Once this is finished, they constantly pass a dedicated message, called a *token*, back and forth between each other. Node 1 sends it to node 2, node 2 then sends it on to node 3, and so on, until the last one sends it back to node 1. If at any time, one of the nodes is turned off, or if a new node is connected, then this process stops, the nodes go through another discovery process, assign themselves new numbers, and go back to sending the token around and around.

To avoid collisions, nodes which have a message to send must wait until they get the token. At that point, they immediately send their traffic, and then go back to waiting for a token before they can send more.

Token ring networks therefore never have collisions. Even when there is a lot of traffic to send, there is an orderly method of deciding who gets to send and when, so

they tend to be much more efficient in periods of heavy traffic. On the other hand, they also have some disadvantages:

- (a) Even when there is very little traffic, a node has to wait until it gets the token before it can transmit.
- (b) The tokens use up time; they are essentially an unproductive overhead.
- (c) When a node is turned off, or when a new node enters the system, it takes a while before the discovery process is finished. During this time, the network cannot be used for traffic.

Compared with CSMA-CD networks, token ring networks are somewhat slower when the traffic load is small, but better when the traffic load is high. There is thus some disagreement as to which is better, but it turns out that token ring networks are rarely installed these days; most new networks are CSMA-CD.

Ethernet LAN Cabling

Most modern local area networks use *Ethernet*. Ethernet was invented in the early 1970's, but has gone through a number of changes since then:

- Original speed was 3 million bits per second (3 million bps or 3 Mbps)
- Soon speeded up to 10 Mbps, nicknamed 10BASE2 and 10BASE5, depending on the type of coaxial cable used (described earlier) or 10BASE-T, where the T stands for unshielded twisted pair cabling
- Most common speed today is 100 Mbps, often called Fast Ethernet or 100BASE-T
- Giga-Bit speeds of either 1000 Mbps (1 Gbps) or 10,000 Mbps (10 Gbps) are available, but not common.

Since 100BASE-T is the most common, let's describe it with more detail.

100BASE-T uses unshielded twisted pair or UTP cable, along with the RJ-45 connectors shown in Fig. 19-4. Both the cable and connectors have eight conductors, but only four are used; the other four are generally unused, although they sometimes carry either regular telephone signals, or else power for remote devices such as surveillance cameras; the latter is called PoE or Power over Ethernet.

The eight wires in the cable are grouped into four pairs; each pair is then tightly twisted to reduce noise pickup and transmission, and then the four pairs are further twisted around each other. Each pair uses similar colors — for example, the orange pair has one plain orange wire, while the other is orange with a white stripe.

Although electrical signals don't care about wire colors, EIA/TIA standard 564 (meaning the Electronic Industry Association / Telecommunications Industry

Association document number 564) specifies the colors and which wire is connected to which pin in the RJ-45 connector. If you just buy premade cables, then you really don't care very much about this, but if you occasionally make your own cables, or if you need to recognize the difference between a straight-through cable and a crossover cable, then the colors and standards become important.

There are two versions of the standard: 564A and 564B. Look at Fig. 19-4 to see how the pins in the connector are numbered from 1 at the top left to 8 at the top right. Standard 564B, the more common one, assigns the following wire colors to the pins:

Pin no.	Color
1	Orange and White
2	Orange
3	Green and White
4	Blue
5	Blue and White
6	Green
7	Brown and White
8	Brown

LAN data uses the orange and green pairs; that is, wires 1, 2, 3, and 6. For example, the LAN connector on the back of a computer sends data out on pins 1 and 2, and receives incoming data on pins 3 and 6.

A normal or "straight-through" cable connects pin 1 to 1, pin 2 to 2, and so on, using the colors in Table 19-1 above. If you connect a computer to a hub or router, the computer *outputs* on pins 1 and 2, so the hub or router must *receive* data on pins 1 and 2, and vice versa. Hub and router jacks are normally wired like that.

But what if you want to connect one computer to another? You must now swap wires, so that pins 1 and 2 of one computer connect to pins 3 and 6 of the other. This requires a *crossover cable* that swaps the wires. In the crossover cable, one connector is wired according to 564B as above, but the other end is wired according to 564A, as shown in Table 19-2:

Pin no.	Color
1	Green and White
2	Green
3	Orange and White
4	Blue
5	Blue and White
6	Orange
7	Brown and White
8	Brown

As you can see, the orange and green pairs are swapped.

Since the orange color is easiest to see, just look for the orange: if they are in the same place at both ends of a cable, you have a straight-through cable; if they are different, then it is a crossover cable.

The UTP cables used for networks are also often identified with a CAT or category number. Originally, the UTP cables were very loosely twisted, but this limited their speed. As LAN speeds have increased, so has the amount of twisting per foot of cable increased. Thus CAT 6 cable, suitable for speeds up to 100 Mbps and above, is more tightly twisted than CAT 3 cable, which was only good up to 10 Mbps.

If you were buying LAN cable, you might also be interested in knowing whether it used solid or stranded wire. Stranded wire is more flexible, and thus better for patch cables or temporary connections; solid wire is considered better for permanent installation, such as inside walls. The connectors for the two types are also slightly different.

There is also the matter of insulation. When plastic insulation burns, it often releases poisonous gases. When LAN cables are run in air conditioning ducts, the must be *plenum-rated* — that is, made of plastic which does not generate poisonous gases in case of a fire.

Ethernet Signals

100Base-T Ethernet uses PAM-5 modulation. That is, it uses pulse amplitude modulation, with the pulses having five different voltage levels. In this way, it squeezes several bits into each signal pulse (remember our discussion of baud vs. bit per second in earlier chapters?)

Much more interesting, however, is how those bits are organized and what they mean.

The MAC address

The actual circuit used to send or receive an Ethernet signal is called a *Network Interface* or *NI*. In many cases, computers use a *NIC* or *Network Interface Card* to hold that interface circuit.

Every manufacturer of a network interface puts a unique binary serial number into the device. That number is called a *MAC address* or *Media Access Control* number. The address consists of 48 bits, divided into six 8-bit bytes. It is usually written as six two-digit hexadecimal numbers separated by dashes, such as 6C-12-4A-3B-28-F5. The MAC address is unique — the first few bits of the number identify the manufacturer, while the remaining bits are the actual serial number.

With 48 bits, there are 2^{48} possible different MAC addresses, which is about 2.81×10^{14} , so in theory, at least, no two Ethernet devices in the whole world should

Start flag	Address	Frame type etc.	Data	Error check	End flag
------------	---------	-----------------	------	-------------	----------

(a) Typical frame

Preamble/ Start bits	Dest. address	Source address	Type / Length	Data	CRC
-------------------------	------------------	-------------------	------------------	------	-----

(b) Ethernet frame

Fig. 19-7. Ethernet frame structure

have the same MAC address. In reality, there is some duplication, partially because some of the smaller manufacturers don't play by the rules, but also because many network interfaces allow the MAC address to be changed.

The Ethernet Frame

Information sent on an Ethernet connection is organized into frames. In Chapter 15, we defined a frame as a group of bits which travel together, and used the drawing of Fig. 19-7(a) as a typical example. The frame used for Ethernet data, shown as Fig. 19-7(b), is almost a perfect match. It consists of the following parts:

- The preamble is just a group of bits that identify the beginning of a frame. It also allows the receiving NI to synchronize its clock to the incoming data.
- The destination address is the MAC address of the NI that is supposed to receive the data, and is 6 bytes (48 bits) long.
- The source address is the 6-byte MAC address of the NI which is sending the frame.
- The type or length is a 2-byte number that can have several different meanings, depending on its value. A value of decimal 1500 or less signifies the length of the data that follows in bytes; larger values are sometimes used to denote other types of frames.
- The data can be anywhere from 46 to 1500 bytes long, and contains the actual data that the sender wants to send.
- Finally, the CRC is a 4-byte Cyclic Redundancy Check used to check whether the frame has been properly received. (See Appendix D for a discussion of CRC codes and how they are used.)

Depending on the length of the data portion, the overall frame can thus be anywhere from 64 bytes long to 1518 bytes long.

The Job of the Network Interface

The Network Interface or NI can be both a source and destination for Ethernet frames.

When sending data, the NI usually receives information from some type of microprocessor, and translates it into the electrical signals that travel on the LAN cable. A better description is this:

(a) The microprocessor gives the NI the destination and source addresses, the length or type code, and the actual data.

(b) The NI calculates the CRC that should follow the above.

(c) The NI waits until there is 'quiet' on the LAN cable, sends out the preamble to synchronize the receiver, and then starts to send the frame.

(d) At the same time, it 'listens' to the bits on the cable to monitor for a collision.

(e) If it detects a collision, it stops sending the frame, waits a random amount of time (generally a few microseconds), and then goes back to step (c).

(f) If it has tried this several times and it still gets collisions, it gives up.

The receiving NI does the following:

(a) It monitors the LAN cable at all times. Whenever it hears a preamble, it synchronizes its clock and then receives the frame. If the destination address in the frame doesn't match the MAC address of the NI, then it ignores the rest of the frame and goes back to monitoring.

(b) If the destination address matches the network interface's own MAC address, then it receives the rest of the frame.

(c) At the end of the frame, the NI checks the received CRC. If the CRC is wrong, then it simply throws away the frame.

(d) If the CRC is correct, then it passes the frame up the line to its own microprocessor.

This description is necessarily a bit vague — we have to wait for a discussion of the OSI Model in the next chapter to fill in the gaps. But it gives the general idea: the network interface is a fairly dumb device. It converts the signal back and forth between its electrical form on the cable, and does a simple CRC check that it is correct. But if there is any kind of error, then it simply throws the frame away. In other words, there is no guarantee that the frame will get through — it is a "best effort only." It doesn't do any error correction, and does not ask for a retransmission if there is a problem. The NI takes a simple approach in cases of problems: "Too Bad — Not My Job!"