Internet, WWW, Html

Contents

1. Introduction to the Internet

This chapter provides a brief history of the Internet. And yet, no exact history can be written about the Internet because the Internet is not an easily definable thing. It is a consensus of ideas, an agreement among friends and colleagues, a reflection of technological trends. It is evidence of the notion that communication among peoples is a good thing; it is a quiet affirmation of individual initiative. In short, the Internet is a very large concept.

Therefore, what you will read in the following pages is the history of a number of discrete events that, when combined in history, resulted in the Internet.

1.1 From the Cold War-A Hot Network

The 1960s were a peculiar time in the United States. 'Fhe start of the decade saw the arrival of nuclear missiles in Cuba. The simmering Cold War with Russia rose to a near boil; the threat of nuclear annihilation was a constant in nearly everyone's daily life.

Concurrent with the blockade of Cuba, the beginning of the Vietnam conflict, and political intrigue in many Third World countries, the Cold War was being fought in research labs, fueled by federal spending and public fear. It was thought that the ability to create and keep a technological edge would determine the winner of the war. Technological advances were coming in a rush, and nowhere were they coming more quickly than in the field of computers. By the late 1960s, every major federally° funded research center, including for-profit businesses and universities, had a computer facility equipped with the latest technology that America's burgeoning computer industry could offer.

The idea developed quickly that these various computer centers could be connected to share data. But the actual means by which they would be connected was colored by the ever-present Russian threat. Any network linking these defense-related centers had to be capable of withstanding disruption by a nuclear attack.

The Advanced Research Projects Agency (ARI'A) within the Department of Defense was charged with finding the best way to interconnect these various computer sites.
The government's research did not start in a vacuum. Both the National Physics Lab in the United Kingdom and France's Societe Internationale de Telecommunications Aeronautiques were experimenting with a means of intercomputer communications called packet switching, which provides tremendous flexibility and reliability in moving commands and data from one computer to another.

1.2 Packet switching versus circuit switching

Packet switching solved the difficult problem of creating a network that could survive attack while providing the greatest communications flexibility. To understand the advantages of the packet switching, consider the following analogy. Suppose that you work for a company that has three buildings (as shown in Figure 1); you want to link the computers in each building. You can string a telephone line from A to B, another from A to C, and another from B to C. When the computer in building A has a message for the computer in building C, it switches to Circuit AC and sends the message. A similar process occurs if the computer in building C has a message for the computer in building B. It turns on Circuit BC and sends the

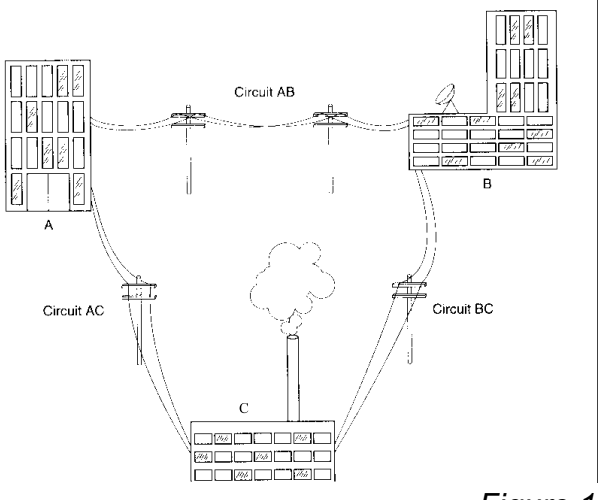

Figure 1

message. This is called circuit switching, a method that works just fine as long as all the circuits are in place and functioning.

But what happens if a large object falls from the sky, smashing one of the telephone poles between buildings A and B, thus destroying Circuit AB? The computer in building A can no longer communicate with the computer in building B.

There is another way: Instead of depending on a circuit to send a message between buildings A and B, you can stuff your message into an electronic envelope (called a packet), put building B's address on the outside of the packet, and, if B can't receive it, send it to any computer on the network that can receive it. In this case, building C would receive the packet, read the address on it, and send it out over its working lines to building B.

Packet switching does not rely on fixed connections between two computers. Rather, messages are contained in packets, which can be routed among computers until they reach their final destination. Very large messages are divided into several packets, each of which is addressed and contains a sequence number so that the message can be reassembled at its destination. In the early days of the Internet, every computer contained a list of all the other computers it knew about on the network. The list had to be updated on a regular basis and was difficult to maintain. Today, a number of computers throughout the world are responsible for keeping track of and registering new computer names on the Internet.

Packets can be nearly any size, but they rarely exceed 1,500 bytes in length. The packet "envelope" usually contains a "to" address, a "from" address, information about the size of the particular packet, and information on where the packet fits in the series of packets that make up a large message. The computer that receives the packet checks particular predefined locations within each packet to get this information.

Packets offer the following benefits:

- Information is divided into discrete chunks that can be routed independently, along various routes, to the destination and then reassembled.
- If a packet disappears or is corrupted during transmission, only the damaged packet must be re-sent, not the whole message.
- Packets can be encoded for security.
- Packets can be compressed to save transmission time.
- A packet can contain information about itself (a checksum) that the receiver can use to validate the accuracy of the contents.
- By use of a standard packet protocol (an agreement on exactly how the packets are to be handled and routed), computers and networks using different kinds of hardware and software can be linked together.
- The best use of communications links can be made because packets from various locations on the network can be intermixed. Instead of getting a "busy signal" until another site is totally finished sending packets, new packets can be slipped into small breaks in the traffic-sort of like carrying on a conversation between the words of someone else's conversation.

ARPA funded a study by the firm Bolt Beranek and Newman (BBN) to find out how communications between these research centers and military installations could be maintained in spite of a nuclear attack. By 1969, BBN had come up with a packet-switching network protocol called the Network Control Protocol and had designed a network controlling computer called an Information Message Processor (IMP), which could manage the network tasks for mainframe computers. The very first IMP was installed at UCLA that same year. By 1970, the first packet-switched computer network in the United States had been created. As shown in Figure 2, ARPAnet connected the University of California at Los Angeles, the University of California at Santa Barbara, Stanford University, and the University of Utah in Salt Lake City.

This was the start of the Internet-four universities connected by a packet-switching network funded by ARPA. If any one link of the network failed, information could still be routed along the remaining links. This satisfied the original criteria for developing a computer network that could withstand hostile attack. By using packets for communications, each computer was at a peer level with every other computer on the network. This arrangement decentralized network control. No one computer was the master and all had equal standing on the network. This fundamental design element was key in encouraging the growth of networks throughout the world and the eventual linking of many of these networks into one world-wide Internet.

Figure 2

By 1972, there were 40 different sites attached to ARPAnet. The electronic traffic between these sites included small text files sent between individual users-a transfer called electronic mail or e-mail. The University of Utah was the first to control a remote computer over the network-a process called remote login or rlogin. Large text and data files were transferred between computers on ARPAnet using File Transfer Protocol (FTP). Thus, by 1972, the core technology was in place.

1.3 Outcome of the First ICCC

In 1972, the first International Conference on Computer Communications was held in Washington, D.C. Attended by representatives from around the world, the conference sought an agreement about communication protocols between different computers and networks. Vinton Cerf, who was involved in the establishment of the ARPAnet at UCLA, was named the first chairman of the InterNetwork Working Group, a group that was charged with creating a protocol that could be used by nearly any computer network in the world to communicate with any other network.

The year following the IC,CC, ARPA, newly renamed Defense Advanced Research Projects Agency (DARI'A), began a program called the lnternetting Project to study how to link packetswitching networks together.

These two projects resulted in the development and introduction of the two basic Internet protocols. In 1974, Vinton Cerf and Robert Kahn released the Internet Protocol (IP) and the Transmission Control Protocol (TCP). These two protocols defined the way in which messages (files or commands) are passed among computer networks on the Internet.

1.4 TCP/IP

Fundamentally, communications protocols are rules that govern the way one machine communicates with another. We can use the English language to demonstrate. When you are reading text written in English, you have some idea of what rules-protocolsgovern the language. For example, you know that most sentences start with a capitalized word, and that a sentence ends with some sort of punctuation mark. In between, sentence fragments are separated by commas, semicolons, or colons. Large thoughts are divided into paragraphs (unless you are Norman Mailer), and so on. Thus, there are commonly understood rules governing written communication.

The Internet Protocol (IP) is a similar body of rules that forms the foundation for all communications over the Internet. Among the rules the IP establishes are these:

- Every node (computer) on the Internet has an Internet address made up of four numbers, and

each number is less than 256. (For example, my Internet provider's address is 192.108.254.10) The address numbers are separated by periods when written out.
• All messages are divided into packets of information.
• Each message packet is assembled (electronically speaking) into an IP envelope.
• The IP envelope contains the address to which the envelope is being sent and the address of the computer sending the message.

Some of the computers that make up the Internet are called routers. These computers are responsible for directing packets sent out on the Internet to the correct destination. Not every computer on the Internet is a muter, nor is it necessary that every computer on the Internet know the location and path to the computer that the packet is being sent to. It is analogous to your mail carrier picking up an envelope you have addressed to Aunt Flo who lives on 220 E. West St. in Walla Walla, Washington. The person who picked up your mail may not even know where Walla Walla is. But your mail carrier carries the envelope to a post office that routes the envelope to a central post office near Walla Walla, which, in turn, routes the envelope to Aunt Flo's local post office, where a postal carrier picks it up and delivers it to 220 E. West St.
The IP address contains finer and finer location information as you read from left to cigltt. The first IP number indicates which major part of Internet the destination network is on; the right-most number indicates the specific machine being addressed. Again using my Internet provider as an example, the right-most number (10) in the IP address 192.108.254.10 is a Sun workstation named `Kelly`.
Most protocols have layers, and the Internet protocols are no exception. The Internet Protocol (IPj is the foundation; laying on top of the IP is yet another protocol called the Transmission Control Protocol (TCP). Most often, you see these two protocols referred to together as TCP/IP.
TCP is used to handle large amounts of data and to handle situations in which the transmitted data is corrupted. TCP divides large messages into multiple packets. Each packet is then assembled into a TCP envelope, which is in turn assembled into an IP envelope. At the receiving end, the TCP envelopes are separated from the IP envelopes and the original message or data is reassembled. If one or more packets are corrupted (as indicated by bad checksums), the originating computer is sent a request to issue a replacement for the bad packet.

The work done by Kahn and Cerf continues to serve the Internet community. TCP/IP is the protocol of choice in most new networks established today. The approach used in TCP/IP is so straightforward that the original goal of creating a communications pathway among many different kinds of networks using their own internal protocols continues to be met.
It was, however, a curious, counter-intuitive event (given the times and the Cold War fears) that truly made the Internet broadly available: DARPA decided to release TCP/IP to the world, free of charge, with no restrictions. In other words, a core technology that solved the problem of computer-network reliability in times of war was suddenly released to the world.

1.5 UNIX and Digital Equipment Corporation

The next part of the Internet story involves the development of a new concept in computer operating systems and a low-cost minicomputer.

Digital Equipment Corporation (DEC) was one of the early developers of the minicomputer, a breakthrough in relatively low-cost computers for the masses (as opposed to the large mainframes from IBM and Control Data that cost hundreds of thousands or even millions of dollars). DEC developed the PDP series of computers, followed in the early 1970s by the VAX family of computers. These moderately powerful computers could be afforded by many colleges, universities, and high-tech businesses. Originally, the VAX computers were only shipped with operating system software called VMS, but that was soon to change.

About the same time, researchers at AT&T Bell Labs were experimenting with a home-grown, multitasking, operating system that ran on DEC minicomputers: a system called UNIX.

UNIX was, from the beginning, an operating system that understood networking. In 1976, Mike Lesk at AT&T Bell Labs created a software package called the UNIX-to-UNIX Copy Program, or UUCP. With UUCP, any UNIX computer with a modem could call any other UNIX computer with a modem and transfer files. AT&T Bell Labs starting shipping UUCP with UNIX version 7 in 1977.
Here was a widely available and affordable computer that could run an operating system that actually had built-in support for networking. The UNIX/DEC combination spread like wildfire throughout industry and academia. Networking was no longer an esoteric act performed on expensive, government-sponsored computer facilities. All those slightly renegade UNIX users quickly understood and adopted the idea of networking.

UNIX was the original "open" system, and it promoted an anarchistic attitude toward computing. Clashes between traditional data processing organizations (with their rightful focus on limited access and security) were the antithesis of the UNIX approach. As much as anything else, UNIX was a game, and its users were global players.

1.6 The underground network

The community was held in some disrepute by the data-processing community in many companies. While working at Tektronix, Inc. in the late 1970s and early 1980s, I had the privilege of watching a bit of anarchistic behavior unfold.
During that time, a new business unit inside Tektronix was started for developing products to serve the emerging microcomputer marketplace. During the course of creating the business unit, the company purchased several DEC VAX computers and the decision was made to use the UNIX operating system. In addition, the computers were networked to one another internally, and a few modems were purchased to provide dial-up capability. Of course, if users could dial in, they could usually dial out, too.
There was a certain amount of fear at the corporate level about having a computer network-used to develop new products-communicating with other computer sites. Because UNIX was a fairly open operating system, it was difficult to guarantee that outsiders could not break into the development computers and steal designs. Therefore, Tektronix had a list of approved external sites. Any site not on the list could not be dialed up, nor could any site that was not on the list dial in to Tektronix's computers. Within 20 miles or so of the Tektronix site was a private institution of higher learning called Reed College. Because of the rambunctious nature of college students, it was determined that there would be no computer-to-computer communications between Reed College and Tektronix.

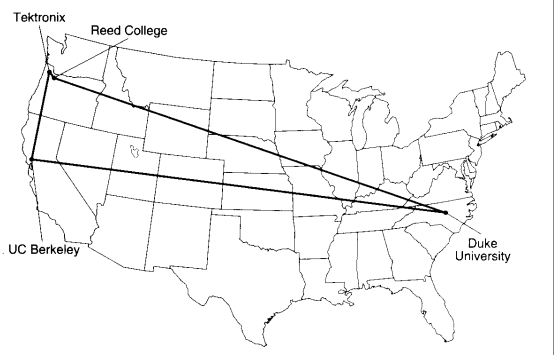

Figure 3

Not everyone at the lower levels believed in this policy. In fact, there were those determined to demonstrate the folly of placing artificial limitations on a computer network. Work was immediately undertaken to establish an alternative, "underground," connection to Reed College. This connection is shown in Figure 3. Tektronix could not communicate directly with Reed College. However, the Universiry of California at Berkeley was on the approved list. UC Berkeley could, in turn, establish a link between its UNIX computers and Duke University in North Carolina. Duke had a link to Reed. Therefore, a long-distance link was established between the VAX computers in Beaverton, Oregon, and Reed College's computers in Portland, Oregon.
This UUCP network was carried over the public telephone system, which was drafted to serve as the information superhighway of the moment.

The wide distribution of DEC minicomputers running the UNIX operating system created a very large, casual network of computers running over the public telephone systems. This was the epitome of a decentralized, ungoverned network.

2. What is a network?

This chapter is a brief dip into the network pool. There are literally hundreds of books available that detail how networks work. This chapter, however, gives you an overview of how networks function, and how the Internet moves millions of bits of information around the world every hour of every day. Previous chapters explained how the networks that make up the Internet networked together with the aid of TCP/IP protocols. The networks that make up the Internet, themselves, use a variety of techniques and protocols to make links within their own networks. This chapter is about how these networks work.

Keep in mind that the Internet is actually a constantly metamorphosing collection of many different kinds of networks. Internet traffic races down telephone lines, flashes from a mountaintop to a high tower on the on the plains using microwave relays, bounces off geosynchronous communications satellites orbiting the Earth 22,000 miles away, enters a laser beam stuck in one end of a fiber-optic cable, comes out the other end, zips around an Ethernet cable strung from one point to another in your office, and speeds through a converter straight to your computer.

2.1 A Network Is...

Of course, nothing with computer networks is quite as simple and straightforward as you might hope. The first concept to adopt is that of layering. Every network has several layers of functionaliry. Accepting the risk of insulting your intelligence, I'll describe how these layers work by using a very primitive two-node networg.

You have two tin cans and a length of string. Poke a hole in the bottom of each can, feed one end of the string through each hole, and tie a knot in each end. If you and a friend (each holding a can) move apart until the string is taut between the cans, you create the ftrst layer in the network: the physical layer. When either of you speaks into a can, the sound waves are transformed into mechanical vibrations in the bottom of the can (the diaphragm) and then are transj muted down the length of string to the other can where they are converted back into sound waves by the bottom of the second can (now a speaker).

If you pull the string tight and both begin to talk at once, you create what's known in the network trade as a data collision. In other words, if everybody talks and nobody listens, people don't communicate. Therefore, you have to build your first communications protocol layer: an agreement between you and your friend that before you speak, you will listen.

Having agreed how to start, you listen, hear nothing, and then begin speaking into your node (can). Your friend listens and then hears you start to speak. But, taking great exception to some,, thing you say, he begins to respond without waiting for you to stop speaking. Another data collision occurs. How do you solve this? You create another protocol layer that says, "When I am done talking, I will say 'over,' just like in the movies."

These are the two aspects of any network: physical and metaphysical. The physical aspect concerns the actual transmission medium. If it's a wire, how quickly can you change voltages on the wire, and what are the minimum and maximum voltage levels? The metaphysical aspect is represented by various protocol layers in which network software running on machine A understands what machine B is trying to say.

Networks vary widely in the physical aspect. A network can be as simple as two I'Cs connected by their serial ports and an RS-232 cable, or multiple PCs sharing a printer by way of an infrared beam bounced off the ceiling. Some networlts rely on a pair of wires twisted around each other, and others use coaxial

shielded cable or fiber-optic cable.

2.2 Network Mediums

Using wire to transmit signals bas been the norm since the days of the telegraph. A single pair ofwires strung between poles can carry a low-quality telephone conversation for some distance. Wires such as these, however, have a couple problems: electro-mechanical interference (EMI) and crosstalk. A magnetic field can create an electrical current-noise (FMI)-in the line, generated, for example, by a lightning storm or an electric motor. If a network uses two linesone to carry one conversation and another to carry a different conversation-and the wires are in close physical proximity, crosstallc occurs. The magnetic field created by one wire causes noise in the other wire.

Several methods have been tried to overcome these two deficiencies. The first is to twist the wires together in pairs, called, obviously enough, twisted pairs. Twisted pairs have keen used for years by the telephone company and have the bandwidth (see the "Bandwidth Determines Quantity of Data" sidebar) to handle several simultaneous telephone conversations.

2.3 Bandwidth

Any communications medium can be discussed in terms of bandwidth: the range of frequencies that can be passed from one end of the medium to the other. For example, in the early days of public telephone systems, bandwidth was very limited. Each conversation required a pair of telephone wires, and only one conversation could occur at a time. People who lived in rural America most likely shared a parry line with their neighbors. When someone called, people listened to the number of rings to determine who was being called.

However, as twisted pairs came into use, available bandwidth went well beyond the 3,500 Hz needed for one conversation. This enabled the creation of fnultiplexed telephone lines. A fairly narrow range of frequencies (300 to 3,500 Hz) is required to carry voice signals. Several of these relatively low-frequency voice signals can be used to modulate one higher, carrier frequency. (In other words, the voice signals are added to, or incorporated within, the higher "carrier" frequency in a manner similar to what radio stations do with a radio frequency carrier and voice frequencies.) The process of combining several voice signals into one is called multiplexing. Suppose that a twistedpair cable has a bandwidth of 1,000 Hz to 18,000 Hz, as shown in Figure 4. You could then broadcast four carrier frequencies at 3, 7, 11, and 15 KHz, and modulate them with the audio frequencies. This creates four voice channels over a single pair of wires. Filters at the receiving end separate the signals. Twisted pairs can actually handle upwards of 100 voice channels.

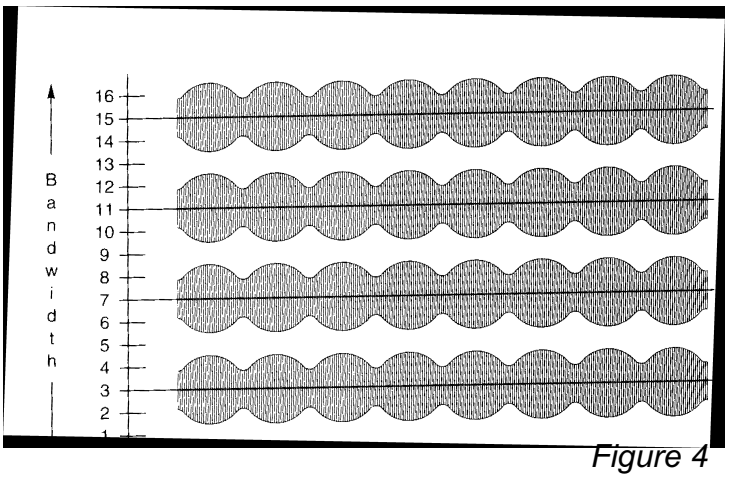

Figure 4

Another method (better than twisted pairs) oFovercoming EMI and crosstalk is to place one wire inside the other, separated by an insulator-the coaxial cable. Coaxial cable can handle a bandwidth several orders of magnitude greater than twisted-pair wires: nearly 10,000 analog voice channels and superior rejection of EMI and crosstalk.

The bandwidth champion, however, is fiber-optic cabling. Because of its nearly complete immunity to outside interference, and a bandwidth of several gigahertz, fiber-optic cabling can carry a tremendous amount of data. Fiber-optic cables contain one or more very thin glass rods (the thickness of a human

hair) with the useful characteristic of being able to conduct a light shown into one end to the other end of the cable, with very little loss. Laser light is modulated to carry the signals at rates above 140 million bits per second (bps). Compare this to the upper limit of 28,800 bps transmission speeds available from currently popular modems used over normal phone lines (the fiber optics are about a thousand times faster).

2.4 Network Topologies

Connecting one computer with another computer is a straightforward task; when several computers are linked, however, there are a number of different ways they can be connected-and each method has advantages and disadvantages. Figure 5 shows several network topologies in use throughout the Internet. The star, hierarchical, and loop arrangements are all point-to point topologies. In this kind of topology, each computer can communicate with its nearest neighbors, but depends on those neighbors to relay data or commands to other computers on the network. The star topology is an extreme example of this configuration because every computer on the network must communicate with the central computer. The other extreme in point-to-point topologies is when all computers are connected to all other computers in a fully connected network. The bus, ring, and satellite configurations are broadcast topologies. The fundamental concept behind this topology is that a message is placed on the bus, in the ring, or broadcast from a satellite. The message contains the name of the intended receiving computer node. All computers listen constantly; when a message addressed to a specific computer arrives, the message is captured and stored. Only one node can broadcast a message at a time. There are two very popular broadcast topologies in use today: the bus-based Ethernet, and the ring-based Token Ring network.

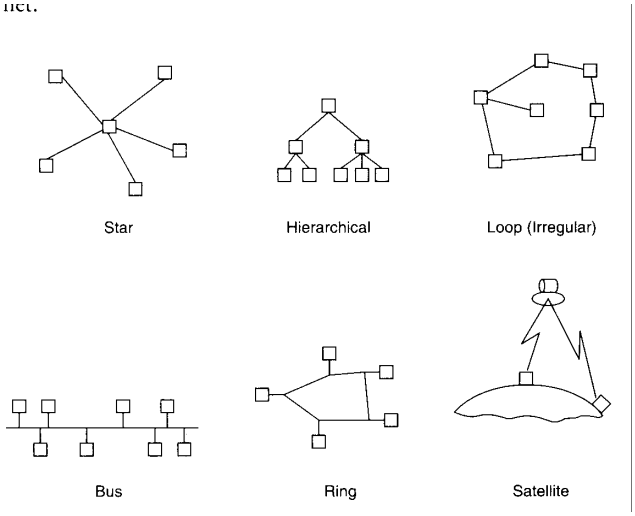

Figure 5

2.4.1 Ethernet

Ethernet is a popular bus-based broadcast topology that gained wide acceptance in the 1980s. Ethernet can transfer data at up to 10 million bps. A four-node Ethernet network is shown in Figure 6.

The network starts with a single cable (which may be coaxial, twisted pairs, or fiber optic). Each computer that taps into the cable must use a transceiver; the computer connects to the transceiver. Transceivers may be circuit boards that plug into a computer's motherboard; other types of transceivers may be connected to the computer's serial or parallel port.

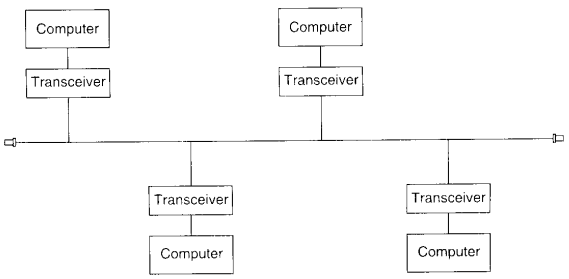

Figure 6

When the message is composed, the sending computer must "listen" to the network (through the transceiver) to see whether any other computer is sending a message at the moment. If not, the sending computer can begin transmitting the message onto the bus. If other traffic is on the bus, the computer must wait before trying again. Finally, while transmitting the message, the computer also simultaneously listens to the bus to see whether the message the bus is carrying is the same as the one it is broadcasting. If not, another computer has begun transmission at almost the same time, and a collision is occurring. When that happens, both computers stop transmitting and wait a random amount of time before trying again. This procedure of listening, and then sending while listening, is called carrier sense multiple access with

collision detection (CSMA/CD). When a transceiver loads a message onto the bus, it creates a carrier signal, which all the other transceivers listen for.

2.4.2 Token Ring

Another broadcast topology in wide use is the Token Ring network. Unlike the loop, which requires each computer in the loop to load every message into itself before passing the message on or keeping it, the ring requires that each computer be attached to a repeater, as shown in Figure 7. Although Token Ring networks appear to be self contained and isolated from the rest of the Internet world, one computer in the ring is usually assigned the task of serving as the gateway computer. The gateway is responsible for converting incoming, external traffic into a form acceptable to the Token Ring. Likewise, outbound traffic is converted to the appropriate communications protocol. The Token Ring, developed by IBM, has an interesting way of letting each node on the ring tell when the ring is available to broadcast a message. An electronic token is created and placed on the ring by one of the computers (usually, a single computer node has the responsibility of creating and issuing the token). This token (which is really a packet containing a fixed data word) loops around the ring, passing through each repeater. If no node issues a message, the token continues its circuitous route. However, when a computer node does transmit a message, the token is captured by that computer's repeater, modified, and sent back out onto the ring. This newly modified token passes around the ring, bur is not recognized by any of the other repeaters; therefore, it indicates to them that the ring is busy and cannot be used.

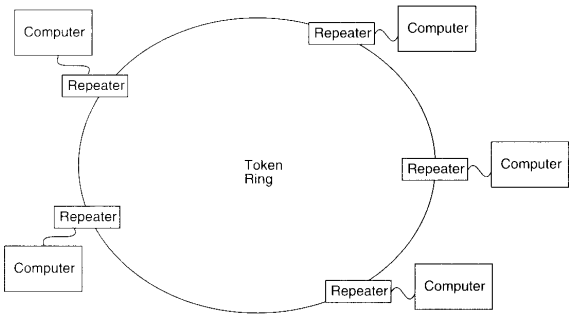

Figure 7

The sending computer prepares a message with a destination address and places the message on the ring. As the packet passes through all the other repeaters in the ring, each repeater reads the packet header to see whether the message is addressed to its node. If the message is not addressed to its node, the repeater passes the message to the next repeater in the ring. If the message is addressed to its node, that repeater sends a copy of the message to its node and passes the message on around the ring. When the message makes a complete circuit around the ring to the originator, the originator reinstates the token, indicating that it is done with the ring. The ring then becomes available to any other node.
As with Ethernet, there is no built-in way for the sender of a message to verify that it was received, beyond the fact that the packet made a complete circuit around the ring; if the receiver was listening, the receiver should have captured the message. Passing an acknowledgment message around the ring is the responsibility of the next higher protocol level.

2.5 Network Protocols

Mentioned in the preceding sections were different protocol "layers" used in network communications. The specifics of each layer, and the number of layers, varies from type to type. The protocol architecture provided in the following example is typical of many networks.

The International Standards Organization (ISO) has developed an architecture that defines seven layers of network protocol (used by many network developers) as a base definition. These seven layers are shown in Figure 8. Keep in mind that protocol layers are, fundamentally, agreements between computers about how to communicate with one another. They are the "rules of the road."

The advantage of layering communication protocols is two-fold:
1. New versions or updates can be written for each layer without affecting the layer before or after. For

example, improvements to the Network layer should not require any changes to the Data Link layer, and any improvements made at this level ripple upward so that higher levels benefit without having to be modified.

2. Two computers on the network need to use only the layers appropriate for the task they are doing.

Each layer in a protocol system such as this uses the previous layer or layers to take action.

The first layer is the Physical layer. This is the definition of how is and Os are passed over the nemvork medium, what control signals are used, and the mechanical properties of the network itself (cable size, connector, and so on). In fact, this is the only layer in which actual communications occur. All the other layers exist as software within the computer that directs and modifies the behavior of the Physical layer.

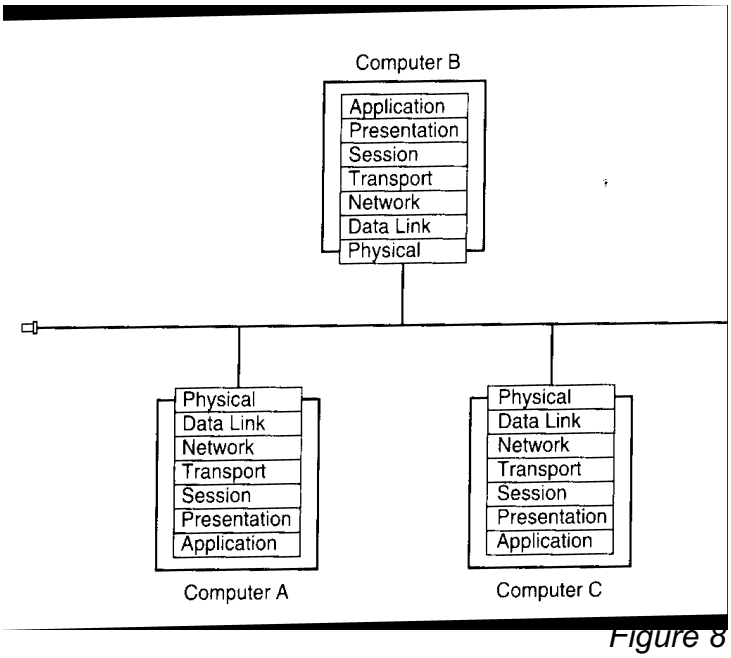

Figure 8

The following chart briefly describes each of the remaining layers.

Layer	Description
Data Link	This layer provides the low-level error detection and correction functions of the network. For example, if a packet is corrupted during transmission, the Data Link layer is responsible for retransmitting the packet.
Network	This layer is responsible for routing packets across the network. If you want to e-mail a large file to another site, for example, the Network protocol layer divides your file into packets, addresses the packets, and sends them out over the network.
Transport	This is an intermediate layer that higher layers use to communicate with the network. This layer hides all the gruesome details necessary to actually make a connection between two computers.
Session	This layer manages the current connection, or session, between two computers. Keep in mind that in packet-switched networks, your computer does not have a full-time connection to a remote computer (even though it may seem so). Your commands to the remote computer are broken into packets and transmitted to the remote machine where they are reassembled and responded to. The Session layer keeps communications flowing until you're done. This layer also validates users that log on your computer through the Internet.

Presentation	The Presentation layer does all the necessary conversion to make sure that both computers are speaking the same language. For example, you may be logged on a Digital Equipment Corporation workstation that uses the ASCII representation for text. If you want to send text to a friend who works with an older IBM mainframe, you have two choices: you can convert the file to an EBCDIC representation and then send the file, or you can let your friend's computer's Presentation layer take care of the conversion for you. Presentation layers also can be used to automatically encode and decode data for transport over the Net.
Application	This is the highest layer in the ISO standard and is represented by the programs you use directly.

Network protocols are critical to intermachine communication. Fortunately, most of us don't have to worry about these various layers. Systems developers have done all the worrying for us. The original Internet protocol, which is still used today, is the set developed by Vinton Cerf and Robert Kahn in 1974: Internet Protocol (IP) and Transmission Control Protocol (TCP), known collectively as TCP/IP.

The TCP/IP protocols have layering similar to the ISO protocols and are, in fact, used more widely throughout the Internet than the ISO protocols. A simplistic view is that the Internet, Protocol (IP) takes care of addressing packets, and the Transmission Control Protocol (TCP) takes care of dividing your message into packers-and then relies on IP to mail them. When a message is received, the reverse happens. The IP captures the various packets and feeds them to TCP-which makes sure that they are all there-and then reassembles the packets into a single message.

2.6 Routing

Now that you understand how messages are sent from one computer node to another within a network, the next question is, "How can an e-mail message get from your network to a single user on a network to a single user on a network accross the country?"

You may recall From earlier chapters that the success of the Internet depends a lot on the fact that it is a packet-switched network. Messages between computers arc converted to small packets that are rapidly toured to their destinations. Each packet contains destination and source addresses, as well as other information that makes routing possible.

In a simple ring topology, a packet is toured around the ring until it gets back to the sender. The assumption is that the receiver saw the packet with its address on it and copied the packet as it went by. However, not every network is a ring. In fact, the Internet is made up of all the possible types of topologies. There must be a more rational way of getting packets from point A to point B. Consider the mixed network shown in Figure 9. Suppose that you want to send a message From your computer on a small network in Portland, Oregon, to a friend in Richardson, Texas. What are the options?

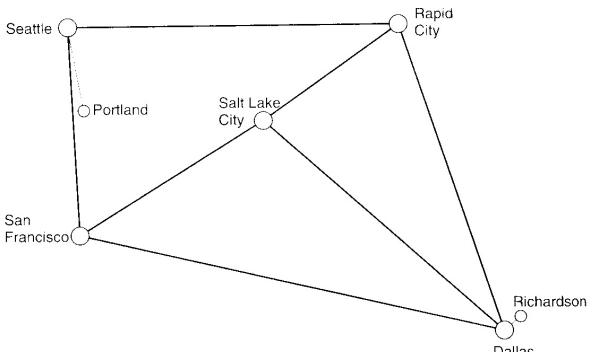

Figure 9

- You could sit in Portland with a network map and lav out a specific route to Richardson. But what happens if one of the links is broken?
- You could keep a list of all possible routes (a routing table) to Richardson in your computer. Your armpurer could keep truing different routes until it succeeds. However, keeping the list up to date is a bother; it requires that every computer on every network keep a routing table up

to date.

• All network nodes could declare one centralized point (Salt Lake City, For example) in the entire network; this point would be responsible for rerouting packets to their appropriate destinations. But what happens if that centralized point goes down?

• One node in each local network could be responsible for keeping track of all the other remote networks.

The fact is that all these approaches (and even more) have been used in the past. Foday, the Internet is so large-consisting of thousands of networks-that keeping track of routing in formation is difficult. Some years ago, every computer on the I nternet was responsible for keeping routing information. Now, computers called routers are used to forward packets in the appropriate direction.

For example, a router in Portland may know that all packets destined for anywhere near Dallas have to go first to Seattle. In Seattle, another router may know that it has two options to route packets to Dallas: by way of San Francisco or by way of Rapid City.The router in Seattle may pick the route with the least traffrc at the moment-for example, to Rapid City. In Rapid City, another roister knows there are a couple of paths available. Once the packet arrives in the Dallas roister, the packet is sent over a local line to your friend's computer in Richardson (or to his or her Internet access provider's computer). This wav, no one computer must keep track of all possible destinations. The routers are responsible for making the major moves; the local machines manage to get the packer to its final destination.

Keep in mind that a large message is broken into a number of packets. Not all packets are necessarily sent out over the same route. The route selected depends on traffic loads and what backbones are working at the moment.

Another method of routing is the "nearest neighbor" method, or centralized adaptive routing. A central node within each network knows onlv about its direct connections to the outside world. For example, in Figure 5.6, Seattle knows about Portland, San Francisco, and Rapid City; Rapid City knows about Seattle, Salt Lake City, and Dallas, but doesn't know about San Francisco.

These are some of the many routing strategies that remain in use today.

3. Domain Names and Internet Addresses

In 1963, the United States Postal Service divided the country into small geographical zones and assigned each zone a ZIP code, which is a five-digit number that enables the postal service to very quickly determine how to route mail.

Your local post office sends all its nonlocal outgoing mail to its nearby main office, where the mail is sorted by ZIP code. The sorted mail is then sent to a distribution center, where it is gathered together and shipped out to the appropriate receiving distribution center (see Figure 10).

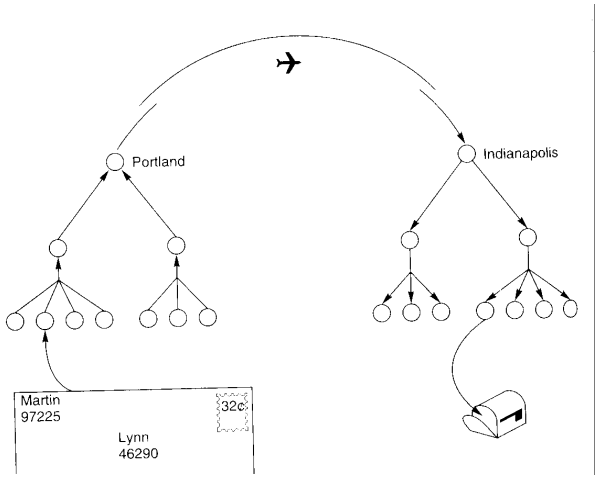

Figure 10

If I address an envelope to my pal Lynn at ZIP code 46290, my mail carrier takes the envelope to the local post office (where they check to see whether the envelope is addressed to a local ZIP code); if it doesn't have a local ZIP, they pass it on to a collection center in Portland. There, envelopes are sorted: any with 46as the first two digits of the ZIP

code are destined for Indianapolis in central Indiana. My envelope is dropped into the Indianapolis bag, and off it goes. Once in Indianapolis, the reverse happens, and the envelope lands in Lynn's local post office. Only then does someone look at the street address. A valid five-digit ZIP code removes the necessity of including a city and state on your envelope (although you should include them in case the post office can't read the ZIP code numbers). The final sorting by postal carrier area is done at the local post office, and then Lynn receives the envelope.

More recently, in 1983, the postal service added another four digits to the ZIP code, enabling the post office to determine your street with the nine-digit number.

3.1 Internet Addresses

The ZIP code is decoded left to right, with the first number identifying the largest geographical area. Each next number (moving right) signifies smaller areas, and the last four digits of the nine-digit number represent the actual street.

A similar thing happens over the Internet. Every computer on the Internet has a specific address called the Internet Protocol (IP) address. Each person that uses a computer on the Internet also has a user name that is combined with the IP address to make their full Internet address. The IP address is made up of four numbers; each number is less than 256. (For example, my Internet provider's address is 192.108.254 10.). The address numbers are separated by periods when written out. As with ZIP codes, IP addresses are decoded from left to right. The first three digits identify the largest divisions ofthe network. (The numeric IP addresses are not based on geographic location.) The next numbers (moving right) signify smaller virtual locations; the final numbers represent the computer you use to access the Internet.

A computer with an IP address is part of the Internet; but if you use your personal computer to dial in to an Internet access provider, your PC does not need an IP address (unless you are making a SLIP or PIT connection). Your PC serves as an intelligent terminal connected to your Internet provider's computer with a phone Line and modem. Your PC is, in effect, just one of several terminals using the common address.

For example, if you want to send me mail, you can do so on some systems by e-mailing to this address:

```
martinm@192.108.254.10
```

The `martinm` is my user name, given to me by the system operator that runs the computer I use to access the Internet. It could just as well have been mmoore, or even just a number, as long as it uniquely identifies me as a user on the system.

The at (`@`) symbol is used to separate the user name from the IP address. The IP address is the four-part number, described previously. Much more commonly, however, addresses are made up of your user name attached to the front of a domain name: `foo@bar.com`.

3.2 Domain Names

As mentioned in the preceding section, on some systems, you use the IP address to send me mail. More commonly, you use the domain name of the Internet provider's computer. Most people have a hard time remembering a long string of numbers containing periods. For those of us who are challenged by long numbers, the Internet provides domain names.

Domain names represent the IP address and are intended to be simpler to understand and easier to remember. If you want to use a domain name to send me e-mail (instead of the IP address used in the preceding section), you address the e-mail as follows:

```
martinm@teleport.com
```

The `teleport` part of the domain name is the unique name of the computer attached to the Internet. The `.com` portion of the domain name identifies what kind of operation this computer serves-in this case, a commercial (com) service.

Domain names can have many parts, each separated by a period (usually referred to on the Internet as a dot). However, instead of providing the largest group on the left (as with IP addresses), domain names provide the largest grouping on the right. In other words, in an IP address, the number to the left of the first period is the largest grouping possible within Internet. It defines the network being used. The rightmost number in an IP address identifies one actual computer. Domain names are just the opposite. The largest grouping within the name is the rightmost part of the name, and the specific computer's name is the leftmost part.

Here are some examples:

Domain Name	What It Means
`cs.wisc.edu`	This is a computer in the computer science (`cs`) department at the University of Wisconsin (`wisc`), which is an educational (`edu`) institution
`xcf.berkeley.edu`	The University of California at Berkeley (`berkeley`) has a computer named `xcf` somewhere in its hallowed halls. Again, UC Berkeley is an educational (`edu`) institution.
`spacelink.msfc.nasa.gov`	The federal government has its own domain name (`gov`). This is a computer named `spacelink` at Marshall Space Flight Center (`msfc`), which is part of NASA (`nasa`).
`prep.ai.mit.edu`	A computer named `prep` is probably in the Artificial Intelligence (`ai`) lab at the Massachusetts Institute of Technology (`mit`), an educational (`edu`) institution.

These domain names are certainly more readily remembered than the IP addresses they represent. Here are several well-recognized, high-level domains in use within the Internet:

Domain Name	Meaning
`com`	Commercial domains used by corporations or companies that have Internet access. The commercial domain also is used by Internet providers such as teleport.com.
`edu`	This domain name is used primarily in the United States to identify educational sites.
`gov`	The U.S. government.
`mil`	U.S. military sites use the mil domain name.
`net`	Some networks choose to use this domain name in identifying themselves. For example, some state-run networks use the net domain name because, although educational sites may be tied into the network, the network serves state and local offices as well.

org	This name is used by organizations. The Internet Society, for example, uses the domain name isoc.org.

You will discover other high-level domain names in use around the world. For example, Internet traffic from Australia usually uses .au as the final element of the domain name; .nz means the computer is located in New Zealand; and .de is used by Germany.

All these domain names are part of what is called the Domain Name Systern. The Domain Name System uses a number of computers scattered throughout the Internet to keep track of which computers are located within its geographical area. When you send e-mail to teleport.com, for example, your host computer sends a request to the nameserver to check its database for the IP number for teleport.com. When the IP number is received back by your host computer, it matches that IP number with your message and sends it out on the Internet.

Since the inception of the Domain Name System in 1983 at the University of Wisconsin, the Internet has been easier to use.

The Internet Network Information Center (NIC) manages the naming of Internet nodes. Anyone who wants a domain name must apply for the name.

4. What are WWW, hypertext and hypermedia?

WWW stands for "World Wide Web." The WWW project, started by Tim Berners-Lee while at CERN (the European Laboratory for Particle Physics), seeks to build a "distributed hypermedia system." In practice, the web is a vast collection of interconnected documents, spanning the world. Tim Berners-Lee continues his pioneering work with the W3 Consortium at MIT.

The advantage of hypertext is that in a hypertext document, if you want more information about a particular subject mentioned, you can usually "just click on it" to read further detail. In fact, documents can be and often are linked to other documents by completely different authors - much like footnoting, but you can get the referenced document instantly!

To access the web, you run a browser program. The browser reads documents, and can fetch documents from other sources. Information providers set up hypermedia servers which browsers can get documents from.

The browsers can, in addition, access files by FTP, NNTP (the Internet news protocol), gopher and an ever-increasing range of other methods. On top of these, if the server has search capabilities, the browsers will permit searches of documents and databases.

The documents that the browsers display are hypertext documents. Hypertext is text with pointers to other text. The browsers let you deal with the pointers in a transparent way -- select the pointer, and you are presented with the text that is pointed to.

Hypermedia is a superset of hypertext -- it is any medium with pointers to other media. This means that browsers might not display a text file, but might display images or sound or animations.

4.1 What is a URL?

URL stands for "Uniform Resource Locator". It is a draft standard for specifying an object on the Internet, such as a file or newsgroup.

URLs look like this: (file: and ftp: URLs are synonymous.)

```
file://wuarchive.wustl.edu/mirrors/msdos/graphics/gifkit.zip
ftp://wuarchive.wustl.edu/mirrors
http://www.w3.org:80/default.html
news:alt.hypertext
telnet://dra.com
```

The first part of the URL, before the colon, specifies the access method. The part of the URL after the colon is interpreted specific to the access method. In general, two slashes after the colon indicate a machine name (machine:port is also valid).

When you are told to "check out this URL", what to do next depends on your browser; please check the help for your particular browser. For the line-mode browser at CERN, which you will quite possibly use first via telnet, the command to try a URL is "GO URL" (substitute the actual URL of course). In Lynx you just select the "GO" link on the first page you see; in graphical browsers, there's usually an "Open URL" option in the menus.

4.2 What are SGML and HTML?

Documents on the World Wide Web are written in a simple "markup language" called HTML, which stands for Hypertext Markup Language.

SGML is a much broader language which is used to define particular markup languages for particular purposes. HTML is just a specific application of SGML. You can learn more about SGML, and the rationale behind HTML, by reading A Gentle Introduction to SGML (URL is <URL:http://etext.virginia.edu/bin/tei-tocs?div=DIV1&id=SG>), a document provided by the Text Encoding Initiative.

4.3 How can I search through ALL web sites?

Several people have written robots which create indexes of web sites - including sites which have not arranged to be mentioned in the newspapers and catalogs above. (Before writing your own robot, please read the entry in the authoring section regarding robots.)

Here are a few such automatic indexes you can search:

Alta Vista

(URL is <URL:http://www.altavista.digital.com>) is probably the most powerful web searching facility at this time, with an exhaustive database and the capability to search USENET newsgroups as well as web sites. The query language is also powerful.

Yahoo

(URL is <URL:http://www.yahoo.com/>) is probably the most complete hierarchical, topical index of web sites, and also features a sophisticated search facility.

Lycos

(URL is <URL:http://fuzine.mt.cs.cmu.edu/mlm/lycos-home.html>) is another web-indexing robot, which includes the ability to submit the URLs of your own documents by hand, ensuring that they are available for searching.

WebCrawler

(URL is <URL:http://webcrawler.com.html>) builds an impressively complete index; on the other hand, since it indexes the content of documents, it may find many links that aren't exactly what you had in mind. However, it does a good job of sorting the

documents it finds according to how closely they match your search.

World Wide Web Worm

(URL is http://www.cs.colorado.edu/home/mcbryan/WWWW.html) builds its index based on page titles and URL contents only. This is somewhat less inclusive, but pages it finds are more

likely to be an exact match with your needs.

InfoSeek

<URL:http://www.infoseek.com/> is a commercial search service which also offers a free web search facility <URL:http://www2.infoseek.com>. You can specify phrases to locate, among other query operations, and InfoSeek's commercial service can search more than just web pages (newsgroups, for instance). InfoSeek's commercial service charges 10 cents per query and offers a free trial to new users. (Increasing load on the free search servers makes this sound better every day.)

OpenText

(URL is <URL:http://www.opentext.com>) also offers a robust web searching facility.

4.4 Producing HTML documents

HTML is the simple markup system used to create hypertext documents.

HTML is not intended to be a comprehensive page-layout system. Instead, HTML aims to let you describe the structure of your document by indicating headings, emphasis, links to other documents and so forth. The more you work with HTML rather than against it, the happier you'll be.

You can include images and other multimedia objects in your documents, but it should be remembered that not all web users have graphical clients, and many web users voluntarily turn graphics off to save downloading time! If you try to spite such users, you will only lose readers (and customers).

You can in fact specify a great deal about the appearance of your document in the latest web browsers. There is no harm in taking advantage of these features, but as a rule of thumb, always make sure your document looks good in a text-based browser such as Lynx as well as in the graphical browser of your dreams.

This is more than a simple matter of taste. Keep in mind that not all users can see!

There are three ways to produce HTML documents: writing them yourself, which is not a very difficult skill to acquire, using an HTML editor, which assists in doing the above, and converting documents in other formats to HTML. The following three sections cover these possibilities in sequence:

- Writing HTML yourself;
- HTML editing tools;
- Conversion tools.

4.4.1 HTML basics.

HTML is the "markup language" used by web browsers to display documents. A web browser treats text as a continuous sequence of words separated by "white space" (one or more spaces, tabs or line breaks) and displays it according to the width of the display window, using "word wrapping" to fit as many words as will fit on a line before starting the next line. Changing the width of the window will reformat the text so it still fits inside the window (try it!).

Since line breaks are ignored, your document will end up as one long continuoous paragraph if you don't do anything about it, regardless of how you laid it out when you wrote it. To tell the browser to start a new paragraph, you have to use markup tags which will be interpreted specially. HTML markup tags are written inside angle brackets "<...>"; the tag to tell a browser to start a new paragraph is <P>. It doesn't matter if you use capitals or not for tags, so <p> means the same thing as <P>.

You can also use markup tags to tell the browser about special formatting requirements (bold or italic text, and so on):

```
<B> ... </B>      Text between <B> and </B> will be
                  displayed as bold text
<I> ... </I>      Text between <I> and </I> will be
                  displayed as italic text
```

HTML tags are almost always used in pairs, like brackets; the closing tag is the same as the opening tag but preceded by "/", so <B> is the opening "boldface" tag and </B> is the closing "boldface" tag, and so on.

Because the characters "<" and ">" and a few others are treated specially by browsers, you have to encode them like this:

```
To display this:    write this:
      <                &lt;
      >                &gt;
      &                &amp;
      "                &quot;
```

Any tags that a browser doesn't recognise will just be ignored, so that if you forget to encode "<" as "<" the browser will treat what follows as a tag. If it doesn't recognise the text after "<" as a valid tag, everything up to the next occurrence of ">" will be ignored, which means that a chunk of your text will just disappear completely. The easiest way to write HTML is to use an HTML editor, which will take care of all these details automatically.

4.4.2 Structure of an HTML document.

An HTML document is actually divided into two parts: a header (which is not displayed) and a body (the text that is actually displayed in the browser window). The overall structure looks like this:

```
<HTML>                  -- start of HTML document
   <HEAD>               -- start of document header
      ...               -- header contents
   </HEAD>              -- end of header
   <BODY>               -- start of document body
      ...               -- body contents
   </BODY>              -- end of body
</HTML>                 -- end of document
```

The only thing the document header needs to contain is a document title which will be displayed in the browser's title bar. A title is enclosed in <TITLE> ... </TITLE> like this:

```
<TITLE>This is a document title</TITLE>
```

In fact, the document structure tags given above (<HTML>, <HEAD> and <BODY>) are normally ignored by browsers; usually, as soon as a browser sees anything which can't be part of the document header, it assumes that it's got to the document body and starts displaying text in the browser window. All the same, it's good practice to put these tags in since some browsers might require them.

4.4.3 Document headings.

To provide headings like the one immediately above, you can use the tag <H1> ... </H1>. The text in between is displayed as a separate paragraph in a large font. For example, if you write this:

```
<H1>A Level 1 Heading</H1>
```

it will be displayed like this:

```
A Level 1 Heading
```

Level 1 headings like this are normally only used at the start of a document. There are five other levels for subheadings:

```
<H2>A Level 2 Heading</H2>
<H3>A Level 3 Heading</H3>
<H4>A Level 4 Heading</H4>
<H5>A Level 5 Heading</H5>
<H6>A Level 6 Heading</H6>
```

which will be displayed like this:

```
A Level 2 Heading

A Level 3 Heading

A Level 4 Heading

A Level 5 Heading

A Level 6 Heading
```

4.4.4 Preformatted text.

Sometimes you want text to be displayed exactly as you've written it (e.g. program code). To do this, enclose the text in <PRE> ... </PRE> like this:

```
<PRE>
   This text will be displayed exactly as it was typed
                     including any indentation
   or alignment      into columns
   like              this
   Blank lines       are also possible

   You can still use <B>bold text</B> or <I>italic text</I> in
   preformatted text.
</PRE>
```

This will be displayed as:

```
   This text will be displayed exactly as it was typed
                     including any indentation
   or alignment      into columns
   like              this
   Blank lines       are also possible

   You can still use bold text or italic text in
   preformatted text.
```

4.4.5 Lists.

If you want to write a bulleted list, you enclose the entire list in <UL> ... </UL> and then start individual list items with <LI>. For example:

```
<UL>
   <LI>List item 1
   <LI>List item 2
</UL>
```

will be displayed like this:

```
  List item 1
  List item 2
```

To produce a numbered list istead of a bulleted list, use <OL> ... </OL> instead of <UL> ... </UL>:

```
<OL>
   <LI>List item 1
   <LI>List item 2
</OL>
```

will be displayed like this:

```
1.List item 1
2.List item 2
```

You can also produce definition lists using <DL> ... </DL>. Each entry in a definition list is in two parts: a definition term which begins with <DT> and a definition part which begins with <DD>. For example, here is an extract from a glossary of terms
elsewhere on this CD:

```
 <DL>
   <DT>BTW
   <DD>&quot;By the way&quot;
   <DT>RTFM
   <DD>&quot;Read the f***ing manual&quot; (yes, really...)
</DL>
```

which will be displayed like this:

```
BTW
     "By the way"
RTFM
     "Read the f***ing manual" (yes, really...)
```

4.4.6 Miscellaneous tags.

Here are a couple more useful tags to round things off:

```
<HR>      A horizontal rule (like the one above the heading for this section)
<BR>      A line break
```

The line break starts a new line, but doesn't put a gap between lines the way that starting a new paragraph would.

4.4.7 Including images in your text.

To include an image, you need to have the image available in a .GIF or .JPG (JPEG) file. To reference the file you use an IMG tag, like this:

```
<IMG SRC="filename.gif">
```

This will display the image in the file filename.gif as part of the current paragraph. If you want the image to be displayed as a separate paragraph, start a new paragraph before and after the IMG tag, or put line breaks (
) before and after.

There's an example of this at the very beginning of the document. Slightly simplified (use "Frame source" from the "View" menu to see the whole truth), it looks like this:

 A Beginner's Guide to HTML
A good introduction to HTML from NCSA. It's a single HTML document, so it's easy to save a copy for offline viewing.

which is produced by the following markup:

```
<DL>
   <DT><IMG SRC="../../../link.gif"> A Beginner's Guide to HTML
   <DD>A good introduction to HTML from NCSA. It's a single HTML
       document, so it's easy to save a copy for offline viewing.
</DL>
```

The image is in the file `link.gif` in the directory three levels above the current one (standard Unix filename conventions are used, so directory names are separated by "/" and ".." means "the directory above this one").

In fact, the filename can be any URL (Uniform Resource Locator) so that it can be on any accessible machine anywhere in the world. URLs are described more fully below.

4.4.8 Hypertext links.

Hypertext links are what make web documents so powerful. A link like this can be used to reference another document, which can be another local file or (like an image) it can be another document anywhere in the world.

Links are generated by using anchor tags. The link above is written like this in HTML:

```
<A HREF="../.././welldone.htm">like this</A>
```

The text between <A> and </A> is highlighted by the browser, and when you click on it the browser goes to the file specified by the HREF part of the tag (in this case, the file welldone.htm in the directory two levels above this one). Simple, isn't it?

You can also use images as hypertext links:

 Press me!

Pressing the "button" will take you to another document. This was done with the following markup:

```
<A HREF="../../../welldone.htm"><IMG SRC="../../../link.gif"></A>
Press me!
```

If you want to link to a specific section in a document, you need to put #section after the filename, which will go to the section called section in the specified document:

```
<A HREF="somefile.htm#index">The index in some file</A>
```

If the reference is to a section of the current document, you just use #section on its own:

```
<A HREF="#contents">Go to the table of contents</A>
```

which will be displayed like this:

Go to the table of contents

To attach a section name to part of a document, you need to use another variation of the <A> tag:

```
<A NAME="section-name">Some text</A>
```

For example, the bookmark "contents" was attached to the heading for the table of contents at the beginning of this document like this:

```
<P><B><A NAME="contents">Contents:</A></B>
```

This has no visible effect on the text. All the section headings in this document have bookmarks attached, which are referenced from the table of contents at the start of the document.

4.4.9 URLs.

As I mentioned earlier, images and hypertext links can both use Uniform Resource Locators (URLs) which can reference documents all over the world. A typical URL looks like this:

```
http://www.comp.it.bton.ac.uk/je/burks.html
```

which references the front page for the online copy of BURKS at the University of Brighton. The URL consists of:

- a protocol specification, which says which Internet protocol the browser needs to use to access the document (in this case http, the HyperText Transfer Protocol);
- the internet address of a server (in this case the server is a machine called `www.comp.it.bton.ac.uk`); and
- the document's filename on the server, in this case `burks.html` in the directory je.

In general, a URL looks like this:

```
protocol://server/document
```

HTML supports many different Internet protocols: FTP, mail and Usenet news are among the commonest. The formats for these are as follows:

```
ftp://server/filename     -- transfer filename from server
                             using anonymous FTP
mailto:user@site          -- send email to the email address
                             user@site
news:groupname            -- connect to the newsgroup groupname
```

For example:

```
ftp://ftp.brighton.ac.uk/pub/je/adacraft/adacraft.zip
                          -- get the file adacraft.zip from
                             the directory pub/je/adacraft
                             by anonymous FTP from
                             ftp.brighton.ac.uk
mailto:je@brighton.ac.uk  -- send email to John English (je)
                             at Brighton University
                             (brighton.ac.uk)
news:comp.lang.ada        -- read the newsgroup comp.lang.ada
```

If you leave out the protocol and server name, the protocol and server name from the current URL will be assumed. So by leaving out the protocol and server name and just providing a file name, you end up referring to a file whose location is relative to the document containing the link.

4.4.10 Summary.

Here's a quick roundup of the HTML tags covered in this document:

Paragraph types:

`<P>`	Paragraph break
`<H1> ... </H1>`	Heading level 1
`<H2> ... </H2>`	Heading level 2
`<H3> ... </H3>`	Heading level 3
`<H4> ... </H4>`	Heading level 4
`<H5> ... </H5>`	Heading level 5
`<H6> ... </H6>`	Heading level 6
`<UL> ... </UL>`	Bulleted (unordered) list
`<OL> ... </OL>`	Numbered (ordered) list
`<LI>`	List item in a bulleted or numbered list
`<DL> ... </DL>`	Definition list
`<DT>`	Definition term
`<DD>`	Definition

Text formatting

`<B> ... </B>`	Bold text
`<I> ... </I>`	Italic text

Miscellaneous

`<TITLE> ... </TITLE>`	Document title
` `	Line break
`<HR>`	Horizontal rule

Hyperlinks

`<IMG SRC="url">`	Inline image
`<A HREF="url"> ... </A>`	Hyperlink to another document
`<A NAME="tag"> ... </A>`	Bookmark within a document