Chapter 1

Overview

1.1 History of Communication Networks

Communication Networks enable users to transfer information in the form of voice, video, electronic mail or e-mail, and computer files. Users request the communication service they need by means of simple procedures using a telephone handset or cellular phone, set-up TV box, or through applications running on a host computer such as PC or workstation.

1.2 Telephone Networks

Figure-1.1 illustrates the telephone network around 1988. One major development at this stage is that the transmission of the voice signals between switches is digital, as indicated by the letter D, instead of analog. An electronic interface in the switch the converts analog signal traveling on the link from the telephone set to the switch into a digital signal called *bit stream*. The figure also shows another major development-*common channel signaling* (CCS). CCS is data communication network that switches use to exchange control information. This “conversation” between the switches serves as the same function as the conversation that took place between operators in the manual network. Thus CCS separates the functions of call control from the transfer of voice. Combined with the flexible computerized switches, this separation of function facilitates new services such as call waiting, call forwarding, and call back. In telephone networks, the bit streams in trucks (lines connecting switches) and access links (lines connecting subscriber telephone to the switch) are organized in the digital signal (DS) hierarchy. The “links” themselves-the “hardware”-are called *digital carrier systems.* Trunk capacity is divided into a hierarchy of logical channels. In North America these channels, listed in the Table-1.1 are called DS-1,.....DS-4 and have rates ranging from 1.544 to 274.176 Mbps. The basic unit is set by the DS-0 channel, which carries 64 Kbps and accommodates one voice circuit. Larger-capacity channels multiplex several voice channels. The rates in Japan and Europe are different. The most common channels are DS-1 and DS-3.

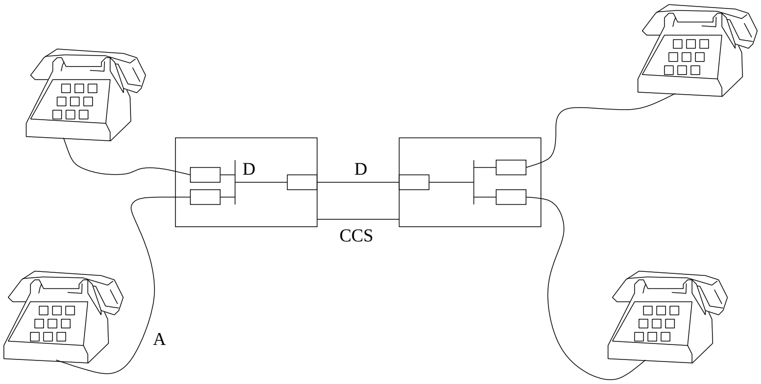


Figure 1.1: Telephone Networks around 1988. The transmissions are analog (A) or digital (D). The switches are electronic and exchange control information by using a data network called common channel signaling

Medium	Signal	No. of VC	Rate in Mbps		
			North America	Japan	Europe
T-1 paired cable	DS-1	24	1.5	1.5	2.0
T-1 C paired cable	DS-1C	48	3.1		
T-2 paired cable	DS-2	96	6.3	6.3	8.4
T-3 coax, fiber	DS-3	672	45.0	34.0	32.0
Coax,fiber	DS-4	4032	274.0		

Table 1.1: Digital carrier systems. This is hierarchy of digital signals that the telephone network uses. Note that the bit rates of the DS-1 signal is greater than 24 times the rate of a voice signal (64 Kbps) because of the additional framing bits required

Since the 1980s the transmission links of the telephone networks have been changing to the SONET, or Synchronous Optical Network, standard. SONET rates are arranged in the STS (Synchronous Transfer Signal) hierarchy shown in the Table-1.2.

Carrier	Signal	Rate in Mbps
OC-1	STS-1	51.840
OC-3	STS-3	155.520
OC-9	STS-9	466.560
OC-12	STS-12	622.080
OC-18	STS-18	933.120
OC-24	STS-24	1244.160
OC-36	STS-36	1866.240
OC-48	STS-48	2488.320
OC-192	STS-192	9853.280

Table 1.2: SONET rates of multiplexed STS-1 signals are exact multiples; no additional framing bits are used

1.3 Computer Networks

Computer or Data Networks has following key innovations: organization of data into packets, packet switching, the Internet Protocol hierarchy, multiple access methods, and service integration. In 1969 RS-232-C standard for *serial port* of computer device, was developed. Then in 1970, the *synchronous transmission* standards were introduced to increase the transmission rate and the usable length of transmission links. These are known as SDLC (Synchronous Data Link Control). The main idea of the SDLC is to avoid the time wasted by RS-232-C caused by gaps between successive characters. To eliminated that lost time, SDLC groups many data bits into *packets.* A packet is a sequence of bits preceded by a special bit patters called *header* and followed by another special bit pattern called *trailer.* The number of bits in a packet may be fixed or variable. SDLC uses an error-detection code called the *cyclic-redundancy check*, or CRC, that is more efficient and more powerful than the single-parity bit of RS-2332-C.

In the early 1960s, communication engineers proposed the *store-and-forward packet-switching* method as shown in the Figure-1.2. When computers use store-and-forward packet switching, they use a given link only when they send a packet. As a result, the same links can be used efficiently by a large number of intermittent transmissions is called *statistical multiplexing.* Statistical multiplexing contrasts with time-division multiplexing-based circuit switching, which reserves circuits for the duration of the conversation even though the parties connected by the circuit may not transmit continuously.

ARPANET, began operations in 1969 by connecting four computers. The rules of operations, or protocols, ARPANET used were published in the open literature.

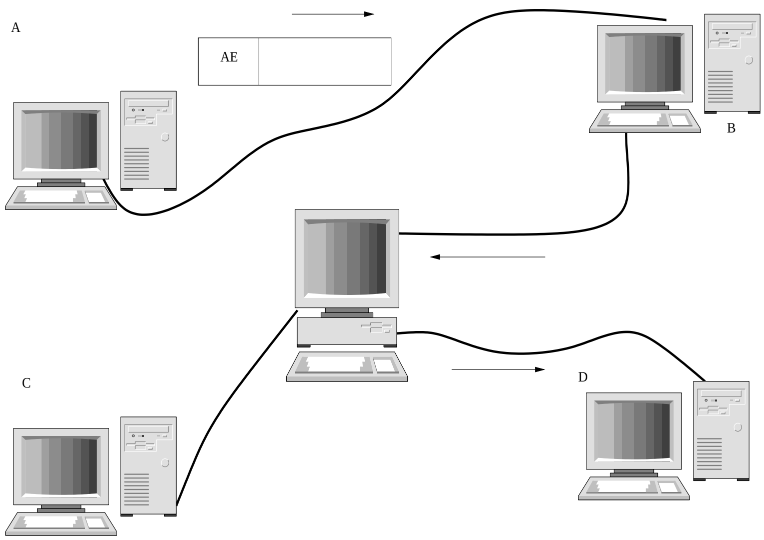


Figure 1.2: Store and Forward Network

By implementing these protocols, engineers in many research and educational institutions attached their computers to the ARPANET. The ARPANET implicitly implemented a three layered architecture consisting of,

1. the physical network that transfer bits,

2. groups of data encapsulated into packets with common format and addressing scheme, and

3. applications that assume transfer of packets with no regard to the underlying physical network.

This implicit layered architecture was subsequently elaborated and formalized in the Open Systems Interconnection, or OSI model. In late 1960s and early 1970ws, engineers proposed a new method for connecting computers. This method is called *multiple access*. It is dramatically reduced the cost of interconnecting nearby computers in a LAN as well as the cost of access to a wide area network (WAN). Figure-3 shows the popular implementation of multiple access called *Ethernet*. In an Ethernet network, computers are attached to a common coaxial cable via an interface that today consists of small chip set mounted on the main board. When computer A wants to send a packet to computer E, it puts the source address A and the destination address E into the packet header and transmits the packet cable. All the compute with the destination address indicated on the packet copies it. The original Ethernet transmission rate was 10 Mbps; now 100 Mbps and 1000 Mbps Ethernet are available.

1.4 What is Internet ?

We would like to give you a one-sentence definition of the Internet, a definition that you can take home and share with your family and friends. Alas, the Internet is very complex, both in terms of its hardware and software components, as well as the services it provides.

1.4.1 A Nuts and Bolts Description

Instead of giving a one-sentence definition, let's try a more descriptive approach. There are a couple of ways to do this. One way is to describe the nuts and bolts of the Internet, that is, the basic hardware and software components that make up the Internet. Another way is to describe the Internet in terms of a networking infrastructure that provides services to distributed applications. Let's begin with the nuts-and-bolts description, using Figure-1.3 to illustrate our discussion.

- The public Internet is a world-wide computer network, i.e., a network that interconnects millions of computing devices throughout the world. Most of these computing devices are traditional desktop PCs, Unix-based workstations,

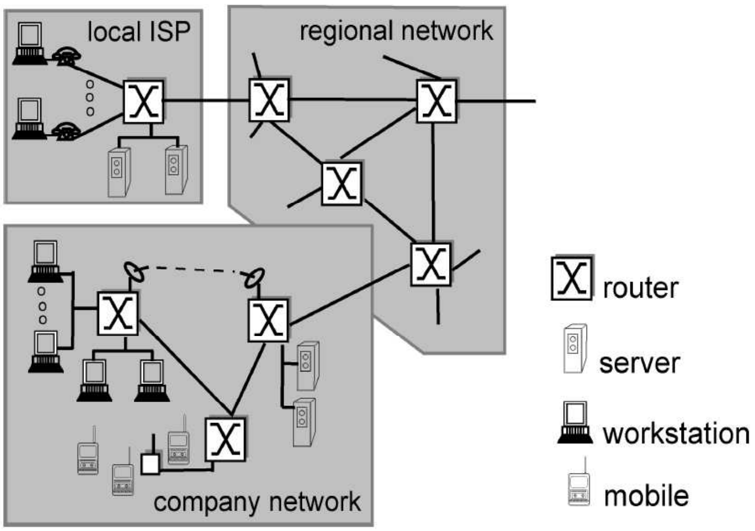


Figure 1.3: Some "pieces" of the Internet

and so called "servers" that store and transmit information such as WWW pages and e-mail messages. Increasingly, non-traditional computing devices such as Web TVs, mobile computers, pagers and toasters are being connected to the Internet. (Toasters are not the only rather unusual devices to have been hooked up to the Internet; see the The Future of the Living Room.) In the Internet jargon, all of these devices are called hosts or end systems. The Internet applications with which many of us are familiar, such as the WWW and e-mail, are network application programs that run on such end systems. We will look into Internet end systems in more detail later and then delve deeply into the study of network applications

- End systems, as well as most other "pieces" of the Internet, run protocols that control the sending and receiving of information within the Internet. TCP (the Transmission Control Protocol) and IP (the Internet Protocol) are two of the most important protocols in the Internet. The Internet's principle protocols are collectively known as TCP/IP protocols.

- End systems are connected together by communication links. We'll see later that there are many types of communication links. Links are made up of different types of physical media: coaxial cable, copper wire, fiber optics, and radio spectrum. Different links can transmit data at different rates. The link transmission rate is often called the link bandwidth, and is typically measured in bits/second.

- Usually, end systems are not directly attached to each other via a single communication link. Instead, they are indirectly connected to each other through intermediate switching devices known as routers. A router takes information arriving on one of its incoming communication links and then forwards that information on one of its outgoing communication links. The IP protocol specifies the format of the information that is sent and received among routers and end systems. The path that transmitted information takes from the sending end system, through a series of communications links and routers, to the receiving end system is known as a route or path through the network. We introduce routing in more detail later, and study the algorithms used to determine routes, as well as the internal structure of a router itself.

- Rather than provide a dedicated path between communicating end systems, the Internet uses a technique known as packet switching that allows multiple communicating end systems to share a path, or parts of a path, at the same time. We will see that packet switching can often use a link more "efficiently" than circuit switching (where each pair of communicating end systems gets a dedicated path). The earliest ancestors of the Internet were the first packet-switched networks; today's public Internet is the grande dame of all existing packet-switched networks.

- The Internet is really a network of networks. That is, the Internet is an interconnected set of privately and publicly owned and managed networks. Any network connected to the Internet must run the IP protocol and conform to certain naming and addressing conventions. Other than these few constraints, however, a network operator can configure and run its network (i.e., its little ”piece” of the Internet) however it chooses. Because of the universal use of the IP protocol in the Internet, the IP protocol is sometimes referred to as the Internet dial tone.

- The topology of the Internet, i.e., the structure of the interconnection among the various pieces of the Internet, is loosely hierarchical. Roughly speaking, from bottom-to-top, the hierarchy consists of end systems connected to local Internet Service Providers (ISPs) though access networks. An access network may be a so-called local area network within a company or university, a dial telephone line with a modem, or a high-speed cable-based or phone-based access network. Local ISP's are in turn connected to regional ISPs, which are in turn connected to national and international ISPs. The national and international ISPs are connected together at the highest tier in the hierarchy. New tiers and branches (i.e., new networks, and new networks of networks) can be added just as a new piece of Lego can be attached to an existing Lego construction. In the first half of 1996, approximately 40,000 new network addresses were added to the Internet - an astounding growth rate.

- At the technical and developmental level, the Internet is made possible through creation, testing and implementation of Internet Standards. These standards are developed by the Internet Engineering Task Force (IETF). The IETF standards documents are called RFCs (request for comments). RFCs started out as general request for comments (hence the name) to resolve architecture problems which faced the precursor to the Internet. RFCs, though not formally standards, have evolved to the point where they are cited as such. RFCs tend to be quite technical and detailed. They define protocols such as TCP, IP, HTTP (for the Web) and SMTP (for open-standards e-mail). There are more than 2000 different RFC's

IP, HTTP (for the Web) and SMTP (for open-standards e-mail). There are more than 2000 different RFC's The public Internet (i.e., the global network of networks discussed above) is the network that one typically refers to as the Internet. There are also many private networks, such as certain corporate and government networks, whose hosts are not accessible from (i.e., they can not exchange messages with) hosts outside of that private network. These private networks are often referred to as intranets, as they often use the same "Internet technology" (e.g., the same types of host, routers, links, protocols, and standards) as the public Internet.

1.5 Service Description

The discussion above has identified many of the pieces that make up the Internet. Let's now leave the nuts and bolts description and take a more abstract, service-oriented, view:

- The Internet allows distributed applications running on its end systems to exchange data with each other. These applications include remote login, file transfer, electronic mail, audio and video streaming, real-time audio and video conferencing, distributed games, the World Wide Web, and much much more. It is worth emphasizing that the Web is not a separate network but rather just one of many distributed applications that use the communication services provided by the Internet. The Web could also run over a network besides the Internet. One reason that the Internet is the communication medium of choice for the Web, however, is that no other existing packet-switched network connects more than 43 million computers together and has 100 million or so users. (By the way, determining the number of computers hooked up to the Internet is a very difficult task, as no one is responsible for maintaining a list of who's connected. When a new network is added to the Internet, its administrators do not need to report which end systems are connected to that network. Similarly, an exiting network does not report its changes in connected end systems to any central authority.)

- The Internet provides two services to its distributed applications: a connection-oriented service and a connectionless service. Loosely speaking, connection-oriented service guarantees that data transmitted from a sender to a receiver will eventually be delivered to the receiver in-order and in its entirety. Connectionless service does not make any guarantees about eventual delivery. Typically, a distributed application makes use of one or the other of these two services and not both. We examine these two different services later in detail.

- Currently the Internet does not provide a service that makes promises about how long it will take to deliver the data from sender to receiver. And except for increasing your access bit rate to your Internet Service Provider (ISP), you currently cannot obtain better service (e.g., shorter delays) by paying more – a state of affairs that some (particularly Americans!) find odd. Our second description of the Internet - in terms of the services it provides to distributed applications – is a non-traditional, but important, one. Increasingly, advances in the "nuts and bolts" components of the Internet are being driven by the needs of new applications. So it's important to keep in mind that the Internet is an infrastructure in which new applications are being constantly invented and deployed.

1.6 What is a Protocol ?

Now that we've got a bit of a feel for what the "Internet" is, let's consider another important word is the title of this book: "protocol." What is a protocol? What does a protocol do? How would you recognize a protocol if you met one?

A Human Analogy: It is probably easiest to understand the notion of a computer network protocol by first considering some human analogies, since we humans execute protocols all of the time. Consider what you do when you want to ask someone for the time of day. A typical exchange is shown in Figure-1.4. Human protocol (or good manners, at least) dictates that one first offers a greeting (the first "Hi" in Figure 4) to initiate communication with someone else. The typical response to a "Hi" message (at least outside of New York City) is a returned "Hi" message. Implicitly, one then takes a cordial "Hi" response as an indication that one can proceed ahead and ask for the time of day. A different response to the initial "Hi" (such as "Don't bother me!", or "I don't speak English," or an unprintable reply that one might receive in New York City) might indicate an unwillingness or inability to communicate. In this case, the human protocol would be to not ask for the time of day. Sometimes one gets no response at all to a question, in which case one typically gives up asking that person for the time. Note that in our human protocol, there are specific messages we send, and specific actions we take in response to the received reply messages or other events (such as no reply within some given amount of time). Clearly, transmitted and received messages, and actions taken when these message are sent or received or other events occur, play a central role in a human protocol. If people run different protocols (e.g., if one person has manners but the other does not, or if one understands the concept of time and the other does not) the protocols do not inter-operate and no useful work can be accomplished. The same is true in networking – it takes two (or more) communicating entities running the same protocol in order to accomplish a task. Let's consider a second human analogy.

Suppose you're in a college class (a computer networking class, for example!). The teacher is droning on about protocols and you're confused. The teacher stops to ask, "Are there any questions?" (a message that is transmitted to, and received by, all students who are not sleeping). You raise your hand (transmitting an implicit message to the teacher). Your teacher acknowledges you with a smile, saying "Yes ...." (a transmitted message encouraging you to ask your question - teachers love to be asked questions) and you then ask your question (i.e., transmit your message to your teacher). Your teacher hears your question (receives your question message) and answers (transmits a reply to you). Once again, we see that the transmission and receipt of messages, and a set of conventional actions taken when these messages are sent and received, are at the heart of this question-and-answer protocol.

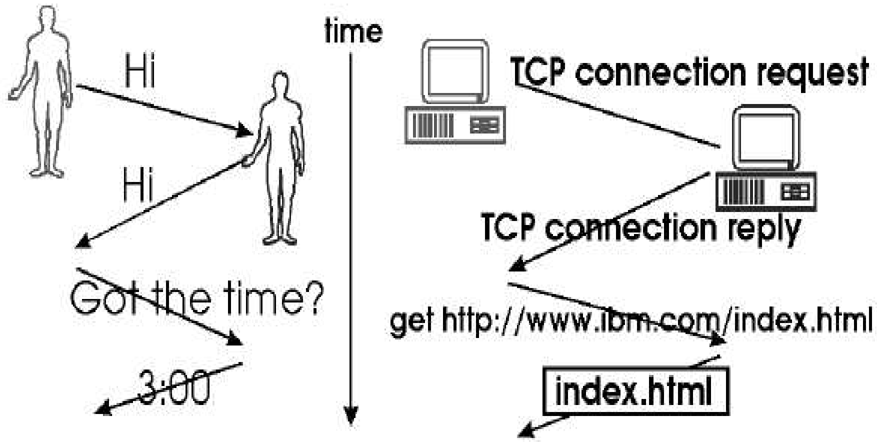


Figure 1.4: A human protocol and a computer network protocol

1.7 Network Protocol

A network protocol is similar to a human protocol, except that the entities exchanging messages and taking actions are hardware or software components of a computer network, components that we will study shortly in the following sections. All activity in the Internet that involves two or more communicating remote entities is governed by a protocol. Protocols in routers determine a packet's path from source to destination; hardware-implemented protocols in the network interface cards of two physically connected computers control the flow of bits on the "wire" between the two computers; a congestion control protocol controls the rate at which packets are transmitted between sender and receiver.

As an example of a computer network protocol with which you are probably familiar, consider what happens when you make a request to a WWW server, i.e., when you type in the URL of a WWW page into your web browser. The scenario is illustrated in the right half of Figure 4. First, your computer will send a so-called "connection request" message to the WWW server and wait for a reply. The WWW server will eventually receive your connection request message and return a "connection reply" message. Knowing that it is now OK to request the WWW document, your computer then sends the name of the WWW page it wants to fetch from that WWW server in a "get" message. Finally, the WWW server returns the contents of the WWW document to your computer.

Given the human and networking examples above, the exchange of messages and the actions taken when these messages are sent and received are the key defining elements of a protocol:

*A **protocol** defines the format and the order of messages exchanged between two* *or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message.*

The Internet, and computer networks in general, make extensive use of protocols. Different protocols are used to accomplish different communication tasks. As you read through this book, you will learn that some protocols are simple and straightforward, while others are complex and intellectually deep. Mastering the field of computer networking is equivalent to understanding the what, why and how of networking protocols.

1.8 The Network Edge

In the previous sections we presented a high-level description of the Internet and networking protocols. We are now going to delve a bit more deeply into the components of the Internet. We begin in this section at the edge of network and look at the components with which we are most familiar – the computers (e.g., PCs and workstations) that we use on a daily basis. In the next section we will move from the network edge to the network core and examine switching and routing in computer networks. Then later we will discuss the actual physical links that carry the signals sent between the computers and the switches.

1.8.1 End Systems, Clients and Servers

In computer networking jargon, the computers that we use on a daily basis are often referred to as or hosts or end systems. They are referred to as “hosts” because they host (run) application-level programs such as a Web browser or server program, or an e-mail program. They are also referred to as “end systems” because they sit at the “edge” of the Internet, as shown in Figure 5. Throughout this book we will use the terms hosts and end systems interchangeably, that is, host = end system. Hosts are sometimes further divided into two categories: clients and servers. Informally, clients often tend to be desktop PC's or workstations, while servers are more powerful machines. But there is a more precise meaning of a client and a server in computer networking. In the so-called client-server model, a client program running on one end system requests and receives information from a server running on another end system. This client-server model is undoubtedly the most prevalent structure for Internet applications. The Web, e-mail, file transfer, remote login (e.g., Telnet), new groups and many other popular applications adopt the client-server model. Since a client typically runs on one computer and the server runs on another computer, client-server Internet applications are, by definition, distributed applications. The client and the server interact with each other by communicating (i.e., sending each other messages) over the Internet. At this level of abstraction, the routers, links and other "pieces" of the Internet serve as a "black box" that transfers messages between the distributed, communicating components of an Internet application. This is the level of abstraction depicted in Figure-1.5.

1.9 Connectionless and Connection-Oriented Services

We have seen that end systems exchange messages with each other according to an application-level protocol in order to accomplish some task. The links, routers and other pieces of the Internet provide the means to transport these messages between the end system applications. But what are the characteristics of this communication service that is provided? The Internet, and more generally TCP/IP networks, provide two types of services to its applications: **connectionless service** and **connection-oriented service**. A developer creating an Internet application

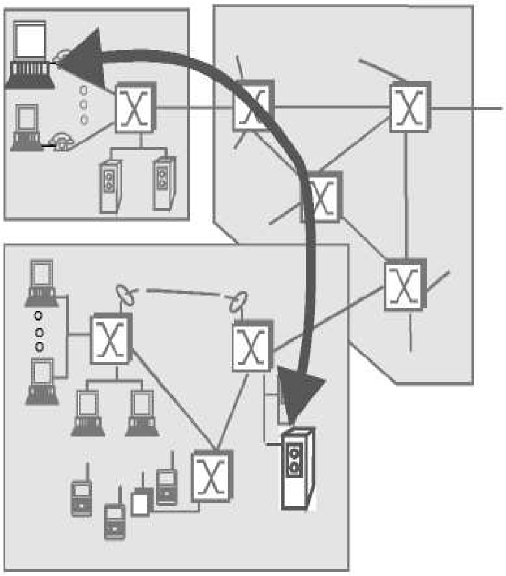


Figure 1.5: End System Interaction

(e.g., an email application, a file transfer application, a Web application or an Internet phone application) must program the application to use one of these two services. Here, we only briefly describe these two services;

1.9.1 Connection-Oriented Service

When an application uses the connection-oriented service, the client and the server (residing in different end systems) send control packets to each other before sending packets with real data (such as e-mail messages). This so-called handshaking procedure alerts the client and server, allowing them to prepare for an onslaught of packets. It is interesting to note that this initial hand-shaking procedure is similar to the protocol used in human interaction. The exchange of "hi's" we saw in Figure 4 is an example of a human "handshaking protocol" (even though handshaking is not literally taking place between the two people). The two TCP messages that are exchanged as part of the WWW interaction shown in Figure 4 are two of the three messages exchanged when TCP sets up a connection between a sender and receiver. The third TCP message (not shown) that forms the final part of the TCP three-way handshake is contained in the get message shown in Figure 4. Once the handshaking procedure is finished, a "connection" is said to be established between the two end systems. But the two end systems are connected in a very loose manner, hence the terminology "connection-oriented". In particular, only the end systems themselves are aware of this connection; the packet switches (i.e., routers) within the Internet are completely oblivious to the connection. This is because a TCP connection is nothing more than allocated resources (buffers) and state variables in the end systems. The packet switches do not maintain any connection state information.

The Internet's connection oriented service comes bundled with several other services, including reliable data transfer, flow control and congestion control. By **reliable data transfer**, we mean that an application can rely on the connection to deliver all of its data without error and in the proper order. Reliability in the Internet is achieved through the use of acknowledgments and retransmissions. To get a preliminary idea about how the Internet implements the reliable transport service, consider an application that has established a connection between end systems A and B. When end system B receives a packet from A, it sends an acknowledgment; when end system A receives the acknowledgment, it knows that the corresponding packet has definitely been received. When end system A doesn't receive an acknowledgment, it assumes that the packet it sent was not received by B; it therefore retransmits the packet.Flow control makes sure that neither side of a connection overwhelms the other side by sending too many packets too fast. Indeed, the application at one one side of the connection may not be able to process information as quickly as it receives the information. Therefore, there is a risk of overwhelming either side of an application. The flow-control service forces the sending end system to reduce its rate whenever there is such a risk. The Internet's congestion control service helps prevent the Internet from entering a state of grid lock. When a router becomes congested, its buffers can overflow and packet loss can occur. In such circumstances, if every pair of communicating end systems continues to pump packets into the network as fast as they can, gridlock sets in and few packets are delivered to their destinations. The Internet avoids this problem by forcing end systems to diminish the rate at which they send packets into the network during periods of congestion. End systems are alerted to the existence of severe congestion when they stop receiving acknowledgments for the packets they have sent.

We emphasize here that although the Internet's connection-oriented service comes bundled with reliable data transfer, flow control and congestion control, these three features are by no means essential components of a connection-oriented service. A different type of computer network may provide a connection-oriented service to its applications without bundling in one or more of these features. Indeed, any protocol that performs handshaking between the communicating entities before transferring data is a connection-oriented service. The Internet's connection-oriented service has a name – TCP (Transmission Control Protocol); the initial version of the TCP protocol is defined in the Internet Request for Comments RFC 793. The services that TCP provides to an application include reliable transport, flow control and congestion control. It is important to note that an application need only care about the services that are provided; it need not to worry about how TCP actually implements reliability, flow control, or congestion control.

1.9.2 Connectionless Service

There is no handshaking with the Internet's connectionless service. When one side of an application wants to send packets to another side of an application, the sending application simply sends the packets. Since there is no handshaking procedure prior to the transmission of the packets, data can be delivered faster. But there are no acknowledgments either, so a source never knows for sure which packets arrive at the destination. Moreover, the service makes no provision for flow control or congestion control. The Internet's connectionless service is provided by UDP (User Datagram Protocol); UDP is defined in the Internet Request for Comments RFC 768. Most of the more familiar Internet applications use TCP, the Internet's connection-oriented service. These applications include Telnet (remote login), SMTP (for electronic mail), FTP (for file transfer), and HTTP (for the Web). Nevertheless, UDP, the Internet's connectionless service, is used by many applications, including many of the emerging multimedia applications, such as Internet phone, audio-on demand, and video conferencing.

1.10 The Network Core

Having examined the end systems and end-end transport service model of the Internet, let us now delve more deeply into the "inside" of the network. In this section we study the network core – the mesh of routers that interconnect the Internet's end systems. Figure-1.6 highlights the network core.

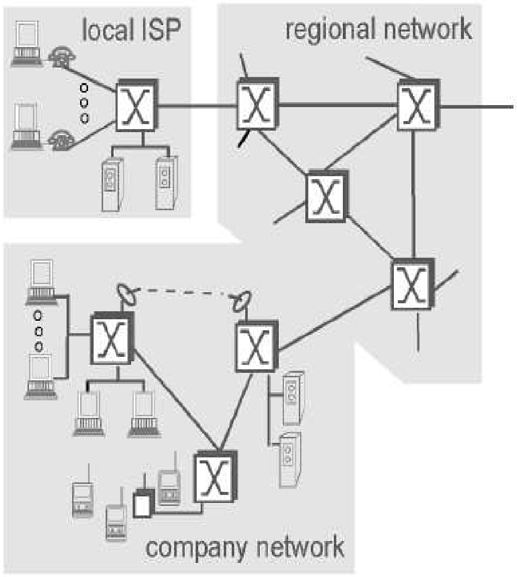


Figure 1.6: The Network Core

1.11 Circuit Switching, Packet Switching, and Message Switching

There are two fundamental approaches towards building a network core: circuit switching and packet switching. In circuit-switched networks, the resources needed along a path (buffers, link bandwidth) to provide for communication between the end systems are reserved for the duration of the session. In packet-switched networks, these resources are not reserved; a session's messages use the resource on demand, and as a consequence, may have to wait (i.e., queue) for access to a communication link. As a simple analogy, consider two restaurants – one which requires reservations and another which neither requires reservations nor accepts them. For the restaurant that requires reservations, we have to go through the hassle of first calling (or sending an e-mail!) before we leave home. But when we arrive at the restaurant we can, in principle, immediately communicate with the waiter and order our meal. For the restaurant that does not require reservations, we don't need to bother to reserve a table. But when we arrive at the restaurant, we may have to wait for a table before we can communicate with the waiter. The ubiquitous telephone networks are examples of circuit-switched networks. Consider what happens when one person wants to send information (voice or facsimile) to another over a telephone network. Before the sender can send the information, the network must first establish a connection between the sender and the receiver. In contrast with the TCP connection that we discussed in the previous section, this is a bona fide connection for which the switches on the path between the sender and receiver maintain connection state for that connection. In the jargon of telephony, this connection is called a **circuit**.

When the network establishes the circuit, it also reserves a constant transmission rate in the network's links for the duration of the connection. This reservation allows the sender to transfer the data to the receiver at the guaranteed constant rate. Today's Internet is a quintessential packet-switched network. Consider what happens when one host wants to send a packet to another host over a packet-switched network. As with circuit-switching, the packet is transmitted over a series of communication links. But with packet-switching, the packet is sent into the network without reserving any bandwidth whatsoever. If one of the links is congested because other packets need to be transmitted over the link at the same time, then our packet will have to wait in a buffer at the sending side of the transmission line, and suffer a delay. The Internet makes its best effort to deliver the data in a timely manner. But it does not make any guarantees. Not all telecommunication networks can be neatly classified as pure circuit-switched networks or pure packet-switched networks. For example, for networks based on the ATM technology, a connection can make a reservation and yet its messages may still wait for congested resources! Nevertheless, this fundamental classification into packet- and circuit-switched networks is an excellent starting point in understanding telecommunication network technology.

1.11.1 Circuit Switching

This course is about computer networks, the Internet and packet switching, not about telephone networks and circuit switching. Nevertheless, it is important to understand why the Internet and other computer networks use packet switching rather than the more traditional circuit-switching technology used in the telephone networks. For this reason, we now give a brief overview of circuit switching. Figure-1.7 illustrates a circuit-switched network. In this network the three circuit switches are interconnected by two links; each of these links has n circuits, so that each link can support n simultaneous connections. The end systems (e.g., PCs and workstations) are each directly connected to one of the switches. (Ordinary telephones are also connected to the switches, but they are not shown in the diagram.) Notice that some of the hosts have analog access to the switches, whereas others have direct digital access. For analog access, a modem is required. When two hosts desire to communicate, the network establishes a dedicated end-to-end circuit between two hosts. (Conference calls between more than two devices are, of course, also possible. But to keep things simple, let's suppose for now that there are only two hosts for each connection.) Thus in order for host A to send messages to host B, the network must first reserve one circuit on each of two links.

A circuit in a link is implemented with either frequency division multiplexing (FDM) or time-division multiplexing (TDM). With FDM, the frequency spectrum of a link is shared among the connections established across the link. Specifically, the link dedicates a frequency band to each connection for the duration of the

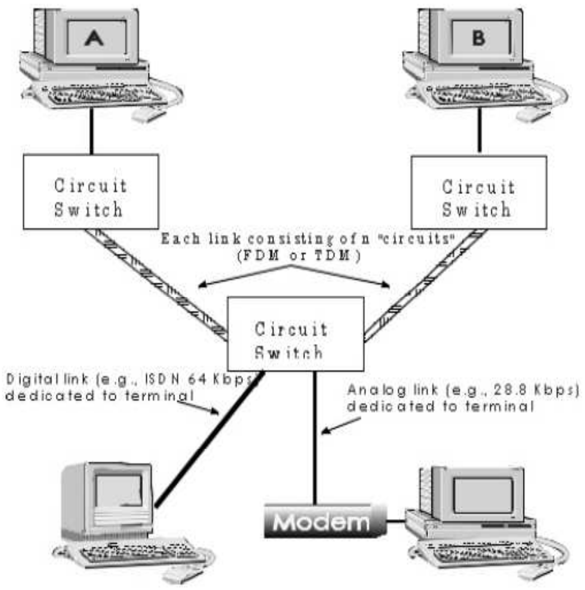

A
B
Circuit Switch
Circuit Switch
Each link consisting of n "circuits" (FDM or TDM)
Circuit Switch
Digital link (e.g., ISDN 64 Kbps) dedicated to terminal
Analog link (e.g., 28.8 Kbps) dedicated to terminal
Modem

Figure 1.7: A simple circuit-switched network consisting of three circuit switches interconnected with two links. Each link has n circuits; each end-to-end circuit over a link gets the fraction $1/n$ of the link's bandwidth for the duration of the circuit. The n circuits in a link can be either TDM or FDM circuits.

connection. In telephone networks, this frequency band typically has a width of 4 kHz. The width of the band is called, not surprisingly, the bandwidth. FM radio stations also use FDM to share microwave frequency spectrum. The trend in modern telephony is to replace FDM with TDM. The majority of the links in most telephone systems in the United States and in other developed countries currently employ TDM. For a TDM link, time is divided into frames of fixed duration and each frame is divided into a fixed number of time slots. When the network establish a connection across a link, the network dedicates one time slot in every frame to the connection. These slots are dedicated for the sole use of that connection, with a time slot available for use (in every frame) to transmit the connection's data. Figure 8 illustrates FDM and TDM for a specific network link. For FDM, the frequency domain is segmented into a number of circuits, each of bandwidth 4 KHz (i.e., 4,000 Hertz or 4,000 cycles per second). For TDM, the time domain is segmented into four circuits; each circuit is assigned the same dedicated slot in the revolving TDM frames. The transmission rate of the frame is equal to the frame rate multiplied by the number of bits in a slot. For example, if the link transmits 8,000 frames per second and each slot consists of 8 bits, then the transmission rate is 64 Kbps.

Proponents of packet switching have always argued that circuit switching is wasteful because the dedicated circuits are idle during silent periods. For example, when one of the conversant in a telephone call stops talking, the idle network resources (frequency bands or slots in the links along the connection's route) cannot be used by other ongoing connections. As another example of how these resources can be underutilized, consider a radiologist who uses a circuit-switched network to remotely access a series of x-rays. The radiologist sets up a connection, requests an image, contemplates the image, and then requests a new image. Network resources are wasted during the radiologist's contemplation periods. Proponents of packet

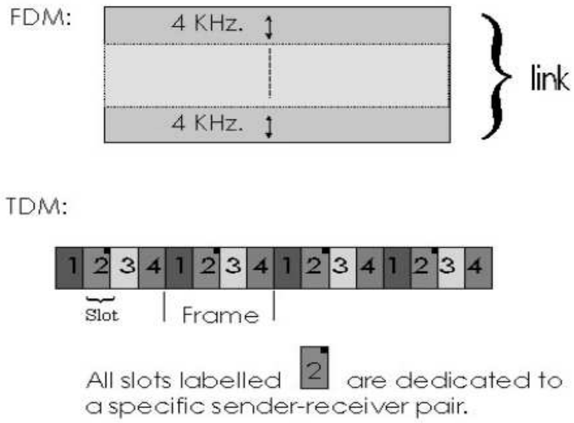

FDM:
4 KHz.
4 KHz.
link
TDM:
1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
Slot
Frame
All slots labelled 2 are dedicated to a specific sender-receiver pair.

Figure 1.8: With FDM, each circuit continuously gets a fraction of the bandwidth. With TDM, each circuit gets all of the bandwidth periodically during brief intervals of time (i.e., during slots).

switching also enjoy pointing out that establishing end-to-end circuits and reserving end-to-end bandwidth is complicated and requires complex signaling software to coordinate the operation of the switches along the end-to-end path.

Before we finish our discussion of circuit switching, let's work through a numerical example that should shed further insight on the matter. Let us consider how long it takes to send a file of 640 Kbits from host A to host B over a circuit-switched network. Suppose that all links in the network use TDM with 24 slots and have bit rate 1.536 Mbps. Also suppose that it takes 500 msec to establish an end-to-end circuit before A can begin to transmit the file. How long does it take to send the file? Each circuit has a transmission rate of (1.536 Mbps)/24 = 64 Kbps, so it takes (640 Kbits)/(64 Kbps) = 10 seconds to transmit the file. To this 10 seconds we add the the circuit establishment time, giving 10.5 seconds to send the file. Note that the transmission time is independent of the number links: the transmission time would be 10 seconds if the end-to-end circuit passes through one link or one-hundred links.

1.11.2 Packet Switching

We saw in previous sections that application-level protocols exchange messages in accomplishing their task. Messages can contain anything the protocol designer desires. Messages may perform a control function (e.g., the ”hi” messages in our handshaking example) or can contain data, such as an ASCII file, a Postscript file, a Web page, a digital audio file. In modern packet-switched networks, the source breaks long messages into smaller packets. Between source and destination, each of these packets traverse communication links and packet switches (also known as routers). Packets are transmitted over each communication link at a rate equal to the full transmission rate of the link. Most packet switches use store and forward transmission at the inputs to the links. **Store-and-forward** transmission means that the switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link. Thus store-and-forward packet-switches introduce a store-and-forward delay at the input to each link along the packet's route. This delay is proportional to the packet's length in bits. In particular, if a packet consists of L bits, and the packet is to be forwarded onto an outbound link of R bps, then the store-and-forward delay at the switch is L/R seconds. Within each router there are multiple buffers (also called queues), with each link having an input buffer (to store packets that have just arrived to that link) and an output buffer. The output buffers play a key role in packet switching. If an arriving packet needs to be transmitted across a link but finds the link busy with the transmission of another packet, the arriving packet must wait in the output buffer. Thus, in addition to the store-and-forward delays, packets suffer output buffer queuing delays. These delays are variable and depend on the level of congestion in the network. Since the amount of buffer space is finite, an arriving packet may find that the buffer is completely filled with other packets waiting for transmission. In this case, packet loss will occur - either the arriving packet or one of the already queued packets will be dropped.

Returning to our restaurant analogy from earlier in this section, the queuing delay is analogous to the amount of time one spends waiting for a table. Packet loss is analogous to being told by the waiter that you must leave the premises because there are already too many other people waiting at the bar for a table.

Figure-1.9 illustrates a simple packet-switched network. Suppose Hosts A and B are sending packets to Host E. Hosts A and B first send their packets along 28.8 Kbps links to the first packet switch. The packet switch directs these packets to the 1.544 Mbps link. If there is congestion at this link, the packets queue in the link's output buffer before they can be transmitted onto the link. Consider now how Host A and Host B packets are transmitted onto this link. As shown in Figure-1.9, the sequence of A and B packets does not follow any periodic ordering; the ordering is random or statistical – packets are sent whenever they happen to be present at the link. For this reason, we often say that packet switching employs statistical multiplexing. Statistical multiplexing sharply contrasts with time-division multiplexing (TDM), for which each host gets the same slot in a revolving TDM frame.

Let us now consider how long it takes to send a packet of L bits from host A to another host across a packet-switched network. Let us suppose that there are Q links between A and E, each of rate R bps. Assume that queuing delays and end-to-end propagation delays are negligible and that there is no connection establishment. The packet must first be transmitted onto the first link emanating from host A; this takes L/R seconds. It must then be transmitted on each of the $Q-1$ remaining links, that is, it must be stored-and-forwarded $Q-1$ times. Thus the total delay is QL/R.

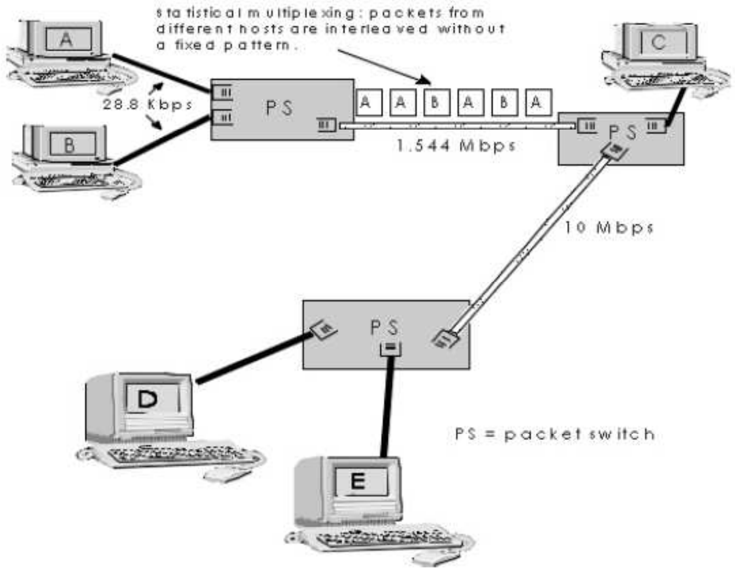

statistical multiplexing: packets from different hosts are interleaved without a fixed pattern.
A
B
C
D
E
28.8 Kbps
PS
A
A
B
A
B
A
1.544 Mbps
PS
10 Mbps
PS
PS = packet switch

Figure 1.9: Packet Switching

1.11.3 Packet Switching Vs. Circuit Switching

Having described circuit switching and packet switching, let us compare the two. Opponents of packet switching have often argued that the packet switching is not suitable for real-time services (e.g., telephone calls and video conference calls) due to its variable and unpredictable delays. Proponents of packet switching argue that (1) it offers better sharing of bandwidth than circuit switching and (2) it is simpler, more efficient, and less costly to implement than circuit-switching. Generally speaking, people who do not like to hassle with restaurant reservations prefer packet switching to circuit switching. Why is packet-switching more efficient? Let us look at a simple example. Suppose users share a 1 Mbps link. Also suppose that each user alternates between periods of activity (when it generates data at a constant rate of 100Kbits/sec) and periods of inactivity (when it generates no data). Suppose further that a user is active only 10 % of the time (and is idle drinking coffee during the remaining 90 % of the time). With circuit-switching, 100 Kbps must be reserved for each user at all times. Thus, the link can support only ten simultaneous users. With packet switching, if there are 35 users, the probability that there are 10 or more simultaneously active users is less than .0004. If there are 10 or less simultaneously active users (which happens with probability .9996), the aggregate arrival rate of data is less than 1 Mbps (the output rate of the link). Thus, users' packets flow through the link essentially without delay, as is the case with circuit switching. When there are more than 10 simultaneously active users, then the aggregate arrival rate of packets will exceed the output capacity of the link, and the output queue will begin to grow (until the aggregate input rate falls back below 1 Mbps, at which point the queue will begin to diminish in length). Because the probability of having ten or more simultaneously active users is very very small, packet-switching almost always has the same delay performance as circuit switching, but does so while allowing for more than three times the number of users. Although packet switching and circuit switching are both very prevalent in today's telecommunication networks, the trend is certainly in the direction of packet switching. Even many of today's circuit-switched telephone networks are slowly migrating towards packet switching. In particular, telephone networks often convert to packet switching for the expensive overseas portion of a telephone call.

1.11.4 Message Switching

In a modern packet-switched network, the source host segments long messages into smaller packets and sends the smaller packets into the network; the receiver reassembles the packets back into the original message. But why bother to segment the messages into packets in the first place, only to have to reassemble packets into messages? Doesn't this place an additional and unnecessary burden on the source and destination? Although the segmentation and reassembly do complicate the design of the source and receiver, researchers and network designers concluded in the early days of packet switching that the advantages of segmentation greatly compensate for its complexity. Before discussing some of these advantages, we need to introduce some terminology. We say that a packet-switched network performs message switching if the sources do not segment messages, i.e., they send a message into the network as a whole. Thus message switching is a specific kind of packet switching, whereby the packets traversing the network are themselves entire messages. Figure 10 illustrates message switching in a route consisting of two packet switches (PSs) and three links. With message switching, the message stays in tact as it traverses the network. Because the switches are store-and-forward packet switches, a packet switch must receive the entire message before it can begin to forward the message on an outbound link.

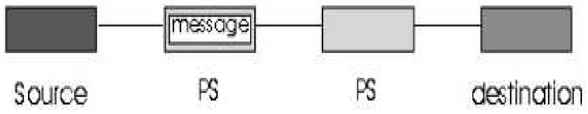


Figure 1.10: A Simple message Switched Network

Figure-1.10 illustrates packet switching for the same network. In this example the original message has been divided into five distinct packets. In Figure 11, the first packet has arrived at the destination, the second and third packets are in transit in the network, and the last two packets are still in the source. Again, because the switches are store-and-forward packet switches, a packet switch must receive an entire packet before it can begin to forward the packet on an outbound link.

One major advantage of packet switching (with segmented messages) is that it

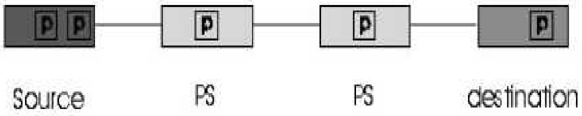


Figure 1.11: A Simple message Switched Network

achieves end-to-end delays that are typically much smaller than the delays associated with message-switching. We illustrate this point with the following simple example. Consider a message that is 7.5 Mbits long. Suppose that between source and destination there are two packet switches and three links, and that each link has a transmission rate of 1.5 Mbps. Assuming there is no congestion in the network, how much time is required to move the message from source to destination with message switching? It takes the source 5 seconds to move the message from the source to the first switch. Because the switches use store-and-forward, the first switch cannot begin to transmit any bits in the message onto the link until this first switch has received the entire message. Once the first switch has received the entire message, it takes 5 seconds to move the message from the first switch to the second switch. Thus it takes ten seconds to move the message from the source to the second switch. Following this logic we see that a total of 15 seconds is needed to move the message from source to destination. These delays are illustrated in Figure-1.12.

Continuing with the same example, now suppose that the source breaks the message into 5000 packets, with each packet being 1.5 Kbits long. Again assuming that there is no congestion in the network, how long does it take to move the 5000 packets from source to destination? It takes the source 1 msec to move the first packet from the source to the first switch. And it takes the first switch 1 msec to move this first packet from the first to the second switch. But while the first

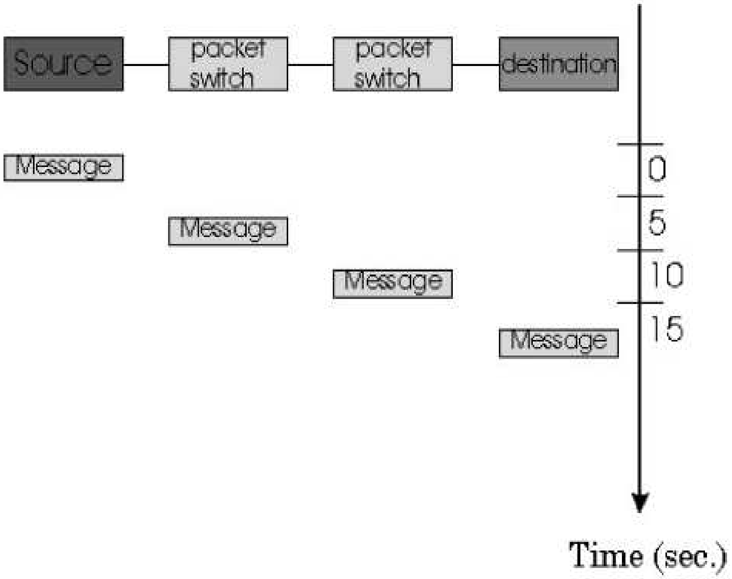


Figure 1.12: Timing of message transfer of a 7.5 Mbit message in a message-switched network

packet is being moved from the first switch to the second switch, the second packet is simultaneously moved from the source to the first switch. Thus the second packet reaches the first switch at time = 2 msec. Following this logic we see that the last packet is completely received at the first switch at time = 5000 msec = 5 seconds. Since this last packet has to be transmitted on two more links, the last packet is received by the destination at 5.002 seconds.

Amazingly enough, packet-switching has reduced the message-switching delay by a factor of three! But why is this so? What is packet-switching doing that is different from message switching? The key difference is that message switching is performing sequential transmission whereas packet switching is performing parallel transmission. Observe that with message switching, while one node (the source or one of the switches) is transmitting, the remaining nodes are idle. With packet

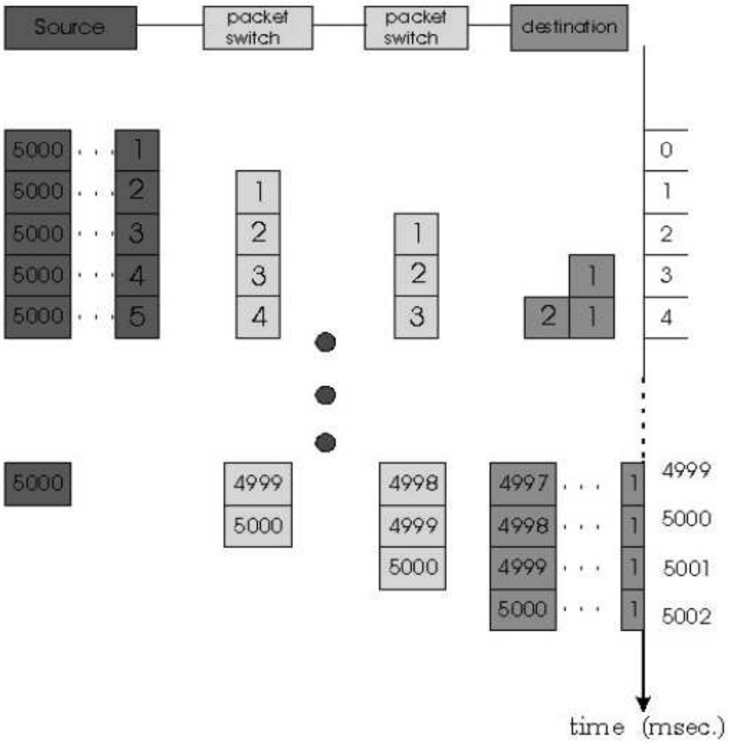


Figure 1.13: Timing of packet transfer of a 7.5 Mbit message, divided into 5000 packets, in a packet-switched network

switching, once the first packet reaches the last switch, three nodes transmit at the same time. Packet switching has yet another important advantage over message switching. As we will discuss later, bit errors can be introduced into packets as they transit the network. When a switch detects an error in a packet, it typically discards the entire packet. So, if the entire message is a packet and one bit in the message gets corrupted, the entire message is discarded. If, on the other hand, the message is segmented into many packets and one bit in one of the packets is corrupted, then only that one packet is discarded. Packet switching is not without its disadvantages, however, with respect to message switching. We will see that each packet or message must carry, in addition to the data being sent from the sending application to the receiving application, an amount of control information. This information, which is carried in the packet or message header, might include the identity of the sender and receiver and a packet or message identifier (e.g., number). Since the amount of header information would be approximately the same for a message or a packet, the amount of header overhead per byte of data is higher for packet switching than for message switching.

1.12 Routing in Data Networks

There are two broad classes of packet-switched networks: datagram networks and virtual-circuit networks. They differ according to whether they route packets according to host destination addresses or according to virtual circuit numbers. We shall call any network that routes packets according to host destination addresses a datagram network. The IP protocol of the Internet routes packets according to the destination addresses; hence the Internet is a datagram network. We shall call any network that routes packets according to virtual-circuit numbers a virtual-circuit network. Examples of packet switching technologies that use virtual circuits include X.25, frame relay, and ATM.

1.12.1 Virtual Circuit Networks

A virtual circuit (VC) consists of (1) a path (i.e., a series of links and packet switches) between the source and destination hosts, (2) virtual circuit numbers, one number for each link along the path, and (3) entries in VC-number translation tables in each packet switch along the path. Once a VC is established between source and destination, packets can be sent with the appropriate VC numbers. Because a VC has a different VC number on each link, an intermediate packet switch must replace the VC number of each traversing packet with a new one. The new VC number is obtained from the VC number translation table. To illustrate the concept, consider the network shown in Figure 14. Suppose host A requests that the network establish a VC between itself and host B. Suppose that the network chooses the path A - PS1 - PS2 - B and assigns VC numbers 12, 22, 32 to the three links in this path. Then, when a packet as part of this VC leaves host A, the value in the VC number field is 12; when it leaves PS1, the value is 22; and when it leaves PS2, the value is 32. The numbers next to the links of PS1 are the interface numbers.

How does the switch determine the replacement VC number for a packet traversing the switch? Each switch has a VC number translation table; for example, the VC number translation table in PS-1 might look something like shown in Figure 15:

You might be wondering why a packet doesn't just keep the same VC number on each of the links along its route? The answer to this question is twofold. First, by replacing the number from link to link, the length of the VC field is reduced. Second, and more importantly, by permitting a different VC number for each link along the path of the VC, a network management function is simplified. Specifically, with the multiple VC numbers, each link in the path can choose a VC number independently of what the other links in the path chose. If a common number were required for all

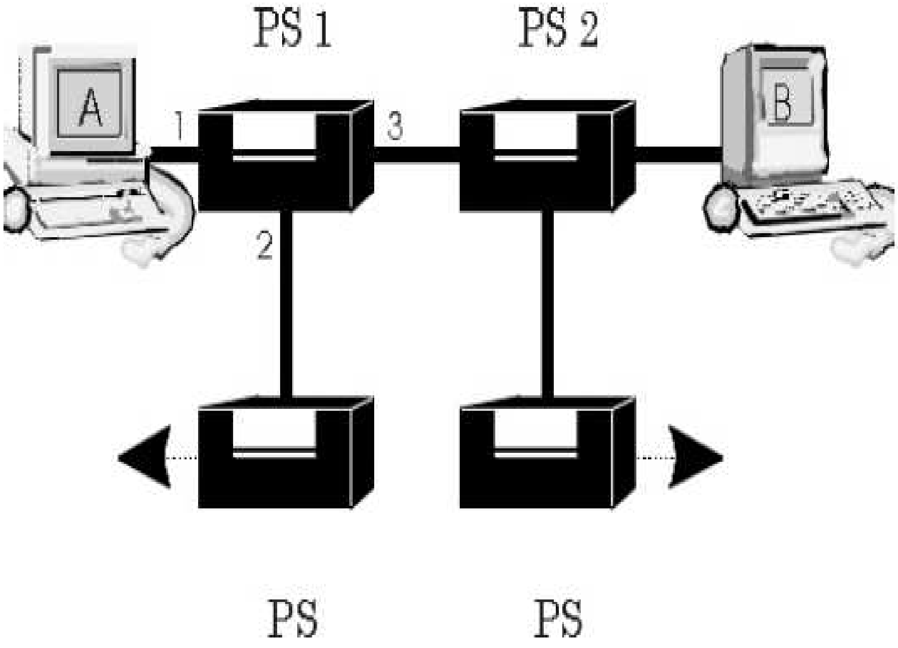
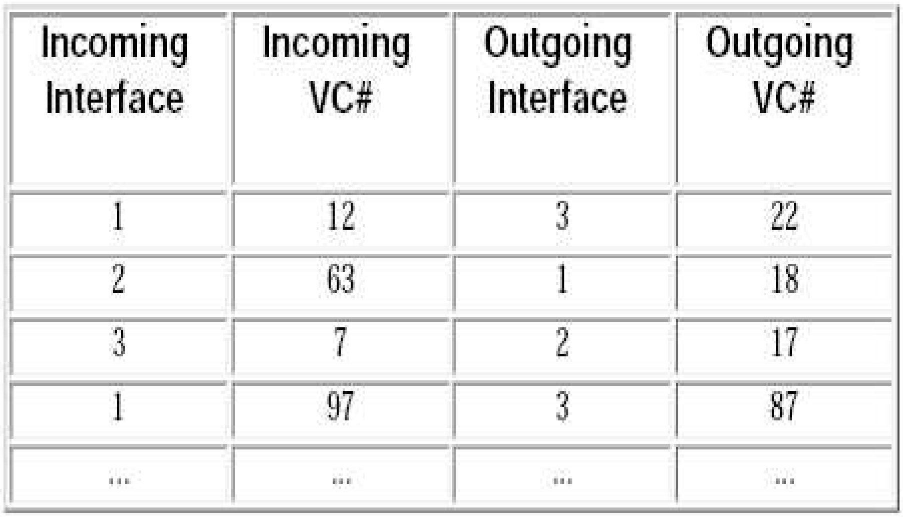


Figure 1.14: A Simple Virtual Circuit Network



Incoming Interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Figure 1.15: Table

links along the path, the switches would have to exchange and process a substantial number of messages to agree on the VC number to be used for a connection.

If a network employs virtual circuits, then the network's switches must maintain state information for the ongoing connections. Specifically, each time a new connection is established across a switch, a new connection entry must be added to the switch's VC-number translation table; and each time a connection is released, an entry must be removed from the table. Note that even if there is no VC number translation, it is still necessary to maintain state information that associates VC numbers to interface numbers. The issue of whether or not a switch or router maintains state information for each ongoing connection is a crucial one - one which we return to shortly below.

1.12.2 Datagram Networks

Datagram networks are analogous in many respects to the postal services . When a sender sends a letter to a destination, the sender wraps the letter in an envelope and writes the destination address on the envelope. This destination address has a hierarchical structure. For example, letters sent to a location in the United States include the country (the USA), the state (e.g., Pennsylvania), the city (e.g., Philadelphia), the street (e.g., Walnut Street) and the number of the house on the street (e.g., 421). The postal services use the address on the envelope to route the letter to its destination. For example, if the letter is sent from France, then a postal office in France will first direct the letter to a postal center in the USA. This postal center in the USA will then send the letter to a postal center in Philadelphia. Finally a mail person working in Philadelphia will deliver the letter to its ultimate destination. In a datagram network, each packet that traverses the network contains in its header the address of the destination. As with postal addresses, this address has a hierarchical structure. When a packet arrives at a packet switch in the network, the packet switch examines a portion of the packet's destination address and forwards the packet to an adjacent switch. More specifically, each packet switch has a routing table which maps destination addresses (or portions of the destination addresses) to an outbound link. When a packet arrives at switch, the switch examines the address and indexes its table with this address to find the appropriate outbound link. The switch then sends the packet into this outbound link. The whole routing process is also analogous to the car driver who does not use maps but instead prefers to ask for directions. For example, suppose Joe is driving from Philadelphia to 156 Lakeside Drive in Orlando, Florida. Joe first drives to his neighborhood gas station and asks how to get to 156 Lakeside Drive in Orlando, Florida. The gas station attendant extracts the Florida portion of the address and tells Joe that he needs to get onto the interstate highway I-95 South, which has an entrance just next to the gas station. He also tells Joe that once he enters Florida he should ask someone else there. Joe then takes I-95 South until he gets to Jacksonville, Florida, at which point he asks another gas station attendant for directions. The attendant extracts the Orlando portion of the address and tells Joe that he should continue on I-95 to Dayton Beach and then ask someone else. In Dayton Beach another gas station attendant also extracts the Orlando portion of the address and tells Joe that he should take I-4 directly to Orlando. Joe takes I-4 and gets off at the Orlando exit. Joe goes to another gas station attendant, and this time the attendant extracts the Lakeside Drive portion of the address, and tells Joe the road he must follow to get to Lakeside Drive. Once Joe reaches Lakeside Drive he asks a kid on a bicycle how to get to his destination. The kid extracts the 156 portion of the address and points to the house. Joe finally reaches his ultimate destination.

We will be discussing routing in datagram networks later. But for now we mention that, in contrast with VC networks, datagram networks do not maintain connection state information in their switches. In fact, a switch in a pure datagram network is completely oblivious to any flows of traffic that may be passing through it – it makes routing decisions for each individual packet. Because VC networks must maintain connection state information in their switches, opponents of VC networks argue that VC networks are overly complex. These opponents include most researchers and engineers in the Internet community. Proponents of VC networks feel that VCs can offer applications a wider variety of networking services. Many researchers and engineers in the ATM community are outspoken advocates for VCs.

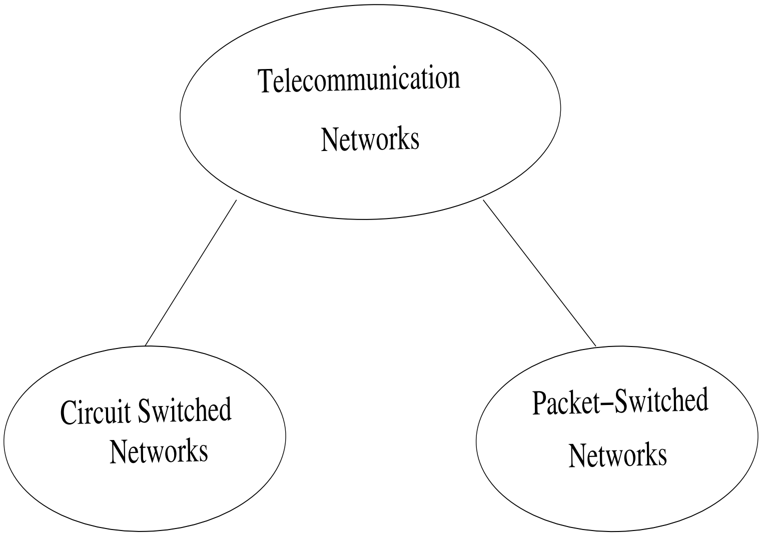


Figure 1.16: highest-level distinction among telecommunication networks: circuit-switched or packet switched?

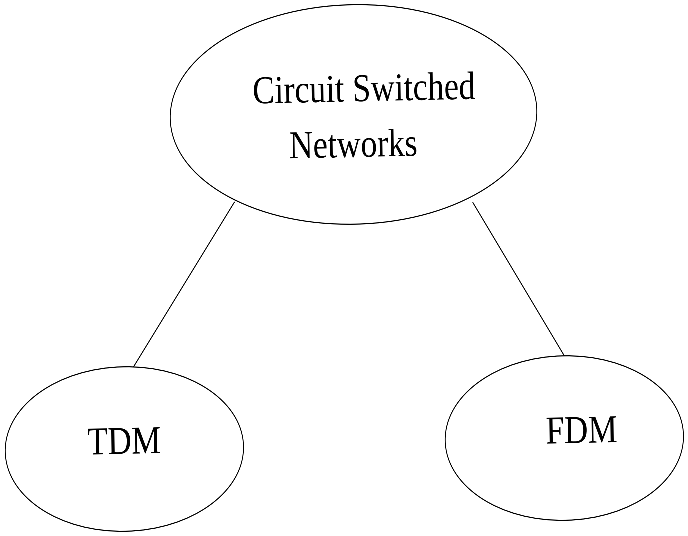


Figure 1.17: Circuit switching implementation: FDM or TDM?

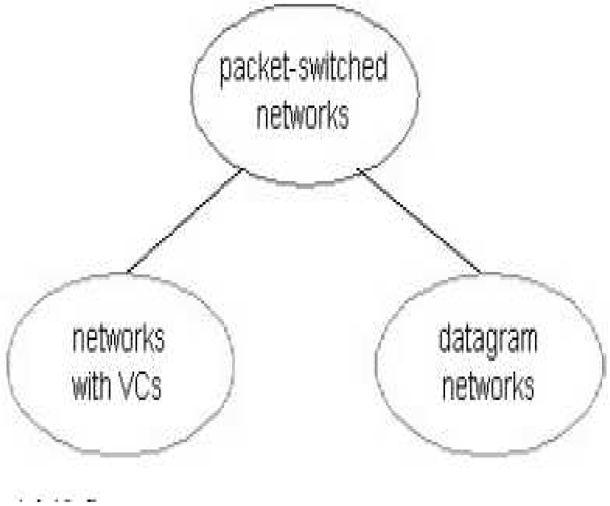


Figure 1.18: Packet switching implementation: virtual circuits or datagrams?

1.13 Access Networks and Physical Media

In sections 1.8 and 1.10 we have examined the roles of end systems and routers in a network architecture. In this section we consider the access network - the physical link(s) that connect an end system to its edge router, i.e., the first router on a path from the end system to any other distant end system.. Since access network technology is closely tied to physical media technology (fiber, coaxial pair, twisted pair telephone wire, radio spectrum), we consider these two topics together in this section.

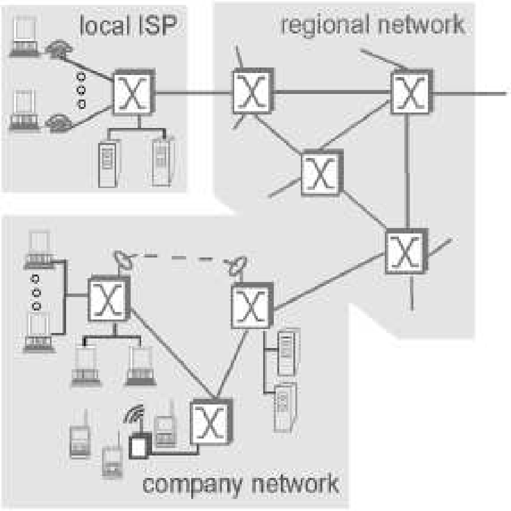


Figure 1.19: Access Networks

1.13.1 Access Networks

Figure-1.19 shows the access networks links highlighted. Access networks can be loosely divided into three categories:

- **residential access networks**, connecting a home end system into the network;

- **institutional access networks**, connecting an end system in a business or educational institution into the network;
- **mobile access networks**, connecting a mobile end system into the network.

These categories are not hard and fast; some corporate end systems may well use the access network technology that we ascribe to residential access networks, and vice versa. Our descriptions below are meant to hold for the common (if not every) case.

Residential access networks

A residential access network connects a home end system (typically a PC, but perhaps a Web TV or other residential system) to an edge router. Probably the most common form of home access is using a modem over a POTS (plain old telephone system) dialup line to an Internet service provider (ISP). The home modem converts the digital output of the PC into analog format for transmission over the analog phone line. A modem in the ISP converts the analog signal back into digital form for input to the ISP router. In this case, the ”access network” is simply a point-to-point dialup link into an edge router. The point-to-point link is your ordinary twisted-pair phone line. (We will discuss twisted pair later in this section.) Today’s modem speeds allow dialup access at rates up to 56 Kbps. However, due to the poor quality of twisted-pair line between many homes and ISPs, many users get an effective rate significantly less than 56 Kbps. For an in depth discussion of the practical aspects of modems see the Institute for Global Communications (IGC) web page on Modems and Data Communications.

While dialup modems require conversion of the end system’s digital data into analog form for transmission, so-called narrowband ISDN technology (Integrated Services Digital Network) allows for all-digital transmission of data from a home end system over ISDN "telephone" lines to a phone company central office. Dialup modems and narrowband ISDN are already widely deployed technologies. Two new technologies, Asymmetric Digital Subscriber Line (ADSL) and hybrid fiber coaxial cable (HFC) [Cable 1998] are currently being deployed. Figure-1.20, fiber optics (also to be discussed soon) connect the cable head end to neighborhood-level junctions, from which traditional coaxial cable is then used to reach individual houses and apartments. Each neighborhood juncture typically supports 500 to 5000 homes.

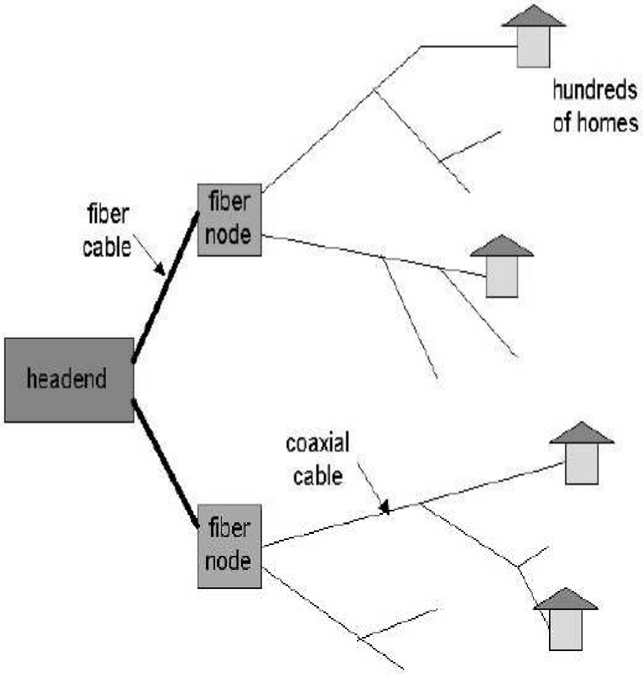


Figure 1.20: A hybrid fiber-coax access network

Enterprise Access Networks

In enterprise access networks, a local area network (LAN) is used to connect an end system to an edge router. However, Ethernet technology is currently by far the most prevalent access technology in enterprise networks. Ethernet operates 10 Mbps or 100 Mbps (and now even at 1 Gbps). It uses either twisted-pair copper wire are coaxial cable to connect a number of end systems with each other and with an edge router. The edge router is responsible for routing packets that have destinations outside of that LAN. Like HFC, Ethernet uses a shared medium, so that end users share the the transmission rate of the LAN. More recently, shared Ethernet technology has been migrating towards switched Ethernet technology. Switched Ethernet uses multiple coaxial cable or twisted pair Ethernet segments connected at a "switch" to allow the full bandwidth an Ethernet to be delivered to different users on the same LAN simultaneously.

Mobile Access Networks

Mobile access networks use the radio spectrum to connect a mobile end system (e.g., a laptop PC or a PDA with a wireless modem) to a base station, as shown in Figure-1.19. This base station, in turn, is connected to an edge router of a data network. An emerging standard for wireless data networking is Cellular Digital Packet Data (CDPD). As the name suggests, a CDPD network operates as an overlay network (i.e., as a separate, smaller "virtual" network, as a piece of the larger network) within the cellular telephone network. A CDPD network thus uses the same radio spectrum as the cellular phone system, and operates at speeds in the 10's of Kbits per second.

1.13.2 Physical Media

In the previous subsection we gave an overview of some of the most important access network technologies in the Internet. While describing these technologies, we also indicated the physical media used. For example, we said that HFC uses a combination of fiber cable and coaxial cable. We said that ordinary modems, ISDN, and

ADSL use twisted-pair copper wire. And we said that mobile access network use the radio spectrum. In this subsection we provide a brief overview of these and other transmission media that are commonly employed in the Internet.

In order to define what is meant by a “physical medium”, let us reflect on the brief life of a bit. Consider a bit traveling from one end system, through a series of links and routers, to another end system. This poor bit gets transmitted many, many times! The source end-system first transmits the bit and shortly thereafter the first router in the series receives the bit; the first router then transmits the bit and shortly afterwards the second router receives the bit, etc. Thus our bit, when traveling from source to destination, passes through a series of transmitter-receiver pairs. For each transmitter-receiver pair, the bit is sent by propagating electromagnetic waves across a physical medium. The physical medium can take many shapes and forms, and does not have to be of the same type for each transmitter-receiver pair along the path. Examples of physical media include twisted-pair copper wire, coaxial cable, multimode fiber optic cable, terrestrial radio spectrum and satellite radio spectrum.

Physical media fall into two categories: *guided media* and *unguided media*. With guided media, the waves are guided along a solid medium, such as a fiber-optic cable, a twisted-pair cooper wire or a coaxial cable. With unguided media, the waves propagate in the atmosphere and in outer space, such as in a digital satellite channel or in a CDPD system.

Chapter 2

Packet Switched Networks

2.1 Open Systems Interconnection (OSI) Model

2.1.1 History of OSI

The Open Systems Interconnection (usually abbreviated to OSI) was an effort to standardize networking that was started in 1982 by the International Organization for Standardization (ISO), along with the ITU-T.

Prior to OSI, according to its proponents, networking was largely vendor-developed and proprietary, with protocol standards such as SNA, Appletalk, NetWare and DECnet. OSI was an industry effort, attempting to get everyone to agree to common network standards to provide multi-vendor interoperability. It was common for large networks to support multiple network protocol suites, with many devices unable to talk to other devices because of a lack of common protocols between them. However while OSI developed its networking standards, TCP/IP came into widespread use on multi vendor networks, while below the network layer, both Ethernet and token ring played much the same role.