

**Military College
of
Electrical and Mechanical Engineering**



FACULTY OF ELECTRONICS

Secunderabad-15

DISSERTATION

ON

**STUDY OF DATA NETWORKS
(LAN, MAN, WAN), SATELLITE AND
FIBRE OPTIC NETWORKS**

Guide :

**Lt Col K S Raju
Faculty of Electronics
MCEME**

Presented by :

Maj Sanjay Malik

Jawaharlal Nehru University, New Delhi

2000

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY
IN
ELECTRONICS

BY

MAJOR SANJAY MALIK

91P + figures

Faculty of Electronics
Military College of EME
Secunderabad

Guide : Lt Col K.S.Raju
Military College of EME
Secunderabad

CERTIFICATE

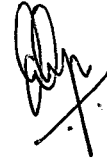
Certified that Major Sanjay Malik of Faculty of Electronics, Military College of Electronics and Mechanical Engineering carried out the dissertation work entitled "**Study of Data Networks, Satellite Networks, Fiber Optic Newtowrks and LAN, WAN & MAN**" in partial fulfilment for the award of Degree of Master of Technology of Jawaharlal Nehru University, under my guidance.

External Examiner

K P Reddy

COL K P REDDY (Retd)

Guide

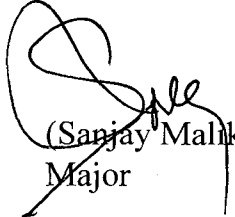


Signature :

Name : Lt Col K S Raju

ACKNOWLEDGEMENT

1. *I consider it pleasant duty to express my heartfelt gratitude, appreciation and indebtedness to Lt Col K S Raju for his able guidance for successful completion of this dissertation.*
2. *I am thankful to Dean Faculty, Brig Surinder Singh, Head C E Department, Col O A James and the staff of C E department, for their kind pursuance, coopertaion and providing necessary facilities for timely completion o fthis dissertation.*
3. *Last, but not the least, I express my sincere thanks to Lt Col D S Chotani, Course Control Officer, Faculty of Electronics for providing the necessary and timely help.*


(Sanjay Malik)
Major

INDEX

PARTICULARS	Page No.
INTRODUCTION	1 ~ 13
NETWORK STRUCTURE & ARCHITECTURE	14 ~ 19
COMMUNICATION BETWEEN & AMONG COMPUTERS & TERMINALS	20 ~ 39
THE OSI REFERENCE MODEL	40 ~ 51
SATELLITE NETWORKS	52 ~ 61
FIBER OPTIC NETWORKS	62 ~ 67
LOCAL / METROPOLITAN / WIDE AREA NETWORKS (LAN/WAN/MAN)	68 ~ 87
FUTURE OF NETWORKING	88 ~ 91
CONCLUSION	

INTRODUCTION

What was once extraordinary is now common place. Twenty years ago, the computer was considered by most people to be a mysterious, esoteric machine. Very few people understood even the most rudimentary aspects of the computer and most individuals viewed it with mistrust and suspicion. On countless occasions, the computer was blamed for problems that were actually the fault of humans. Those who used computers were said to be associated with a priesthood, a term that was used in a derogatory sense to connect a profession that operated in a world largely unknown to the general public.

In less than two decades, the computer has entered the mainstream of our personal and professional lives and has dramatically reshaped our society. The computer has literally created a technology revolution.

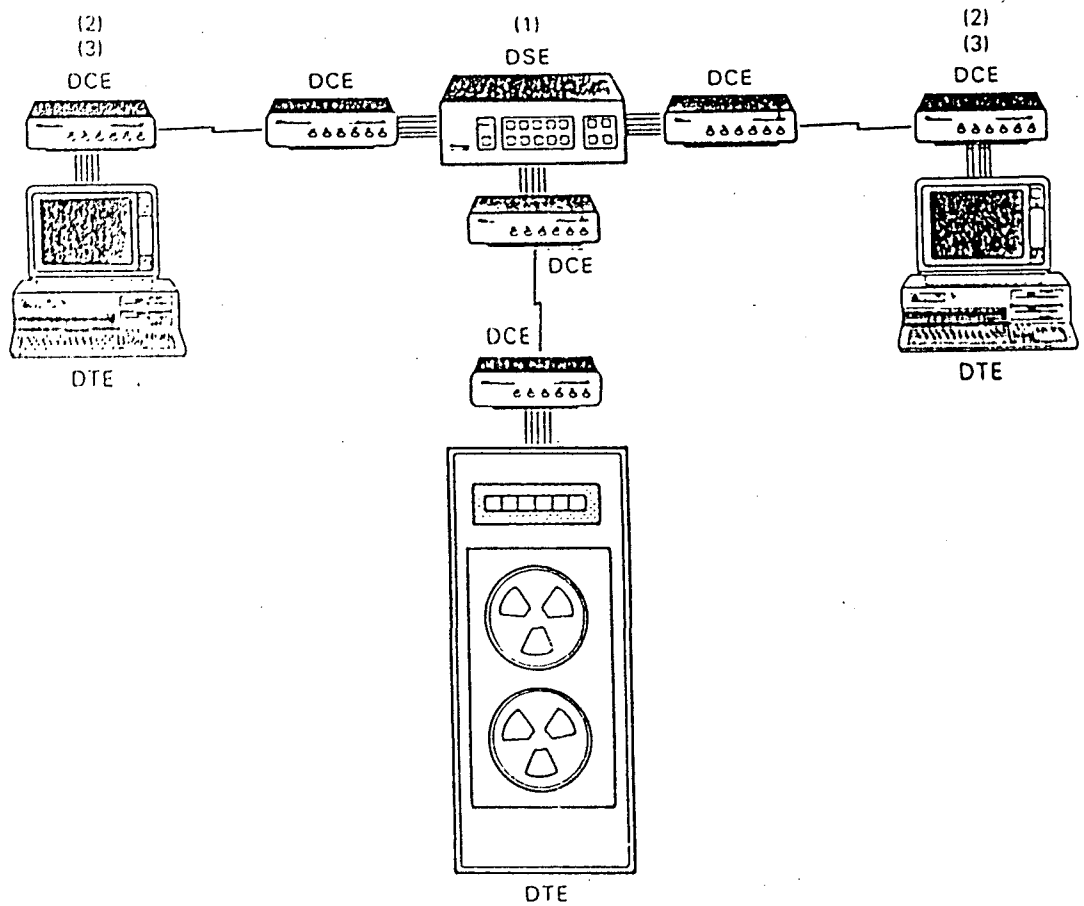
Today, the computer is accepted as simply another tool for doing work. Indeed, many younger people take the computer for granted. They are not aware of the B.C period (before computers). They view a computer commercial on television with the same nonchalance as an automobile advertisement. Discussions at parties often revolve around the latest software package for a personal computer. These now mundane events were simply unthinkable twenty years ago.

As the power and speed of the computer increase, its impact will be even more profound in the future that it is today. As humans become more computer literate, computers will become more people literate. The application of artificial intelligence (AI) will provide services to our society that will even surpass the imagination of writers of today's most far-fetched science fiction.

A vital factor in this information revolution is the use of communications systems to connect computers together. Facilities we take for granted today, such as automated teller machines, personal computer information service, automated factories, and space programs are all dependent upon communications facilities to support the transfer of information between computers' files and data bases. The information revolution also is dependent on communications systems.

Like the computer industry, the communications industry is growing and changing at a fantastic pace. Both computers and communications are being driven by the continuing progress in the development of integrated circuits, lightwave technology, and in the not too distant future, superconductors will bring about more change. It is difficult to think of other technologies that have changed at such a rapid rate. Consider the following:

- Twenty years ago, the maximum transfer rate on a communications line was 1.5 million bits per second. Today optical fiber is capable of a transfer rate of over 600 million bits per second.
- Today a small personal computer has the processing power of a room-size mainframe of the 1960s.



DTE: Data Terminal Equipment (User Station)
 DCE: Data Circuit Terminating Equipment (2)
 DSE: Data Switching Equipment (Switch)
 —: Communications Link

Notes: (1) Not used in some systems
 (2) DCE not present in some systems (short connections)
 (3) DCE may be placed inside the DTE

FIG 1: BASIC COMPONENTS OF A DATA COMMUNICATION SYSTEM

- Twenty years ago, a printer could produce a few hundred lines of print per minute. Today, laser printers produce output at the rate of over 20,000 lines per minute.
- Twenty years ago, a few hundred logic elements could be placed into a single hardware chip. Today, several thousand elements are on one chip.
- Optical switches are now being manufactured that are one fourth the size of the head of a dwarf ant. These switches process data at a rate of 200 billion bits per second. To relate this figure to something more meaningful, this switch can process (examine, switch, etc.) over 10,000 copies of the book you are currently reading – in one second.

Both technical and social scientists believe these trends will continue. The items in the list above will probably be common place in a few short years. The rate of change seems to be increasing exponentially.

This dissertation is a modest attempt to gain an understanding of how communications systems fulfill the vital role of linking computers together. With the introduction behind us, let us now examine the basic components of the data communication system.

Basic Components Of A Data Communications System

The purpose of a data communications system is to transport data between user's computers, terminals and application programs. While the concept is simple, the actual effort involves many steps. Figure 1 shows the basic components of a data communications system.

The user application resides in the data terminal equipment or DTE. DTE is a general term used to describe the end-user machine, which is usually a computer or terminal. The DTE could be a large computer or a small machine such as a terminal or personal computer. The term user station (or station) is used here to describe the DTE. The end-user applications process (AP) and data files reside at the DTE. The AP is another name for a user program.

The function of a data communications system is to interconnect the DTEs so that they can share resources, exchange data, and provide back-up for each other.

The path between the DTEs is called a line, link, circuit, or channel. It may consist of wires, radio signals, or light transmissions. A telephone company often provides this link between the user devices.

Figure 1 also shows the data circuit-terminating equipment (DCE). Its function is to connect the DTEs into the communication line. The DCEs designed in the 1960s and 1970s were strictly communication devices. However, in the last few years, the machines have incorporated more user functions, and today some DCEs contain a portion of a user application process. Nonetheless, the primary function of the DCE is to provide an interface of the DTE into the communication link.

The DCE may be located inside the DTE or stand alone as a separate unit. Wherever it is located, it is used primarily to convert the signals representing user data to a form acceptable to the receiving channel. As a simple example, it might convert an electrical signal from a terminal to a light signal for an optical fiber link. As we shall see, the DCE performs many other important tasks for the data communications link. The DTEs and DCEs communicate with protocols. Protocols are agreements on how the machines “converse” with each other. They may include the logic and codes which stipulate a required or recommended convention or technique. Typically, several levels of protocols are required to support an end-user application, and may be implemented in both software and hardware. The lower level protocols are usually found in hardware, and the upper levels use software.

The link is often connected by another component, the DSE (data switching equipment), or switch. As figure 1 shows, the switch allows the DTEs to use different channels at different times to communicate with different user stations. This arrangement is known as networking. The DSE provides other important functions such as routing around failed or busy devices and channels. The DSE may also route data to the final destination through other switches.

Many computer and terminals can be interconnected to form networks. This arrangement may or may not use a switch. In figure 1, a switch performs the interconnections that move the data through the network. This type of network is called a switched network.

Another approach is the broadcast network. All stations share a common channel, and a station transmits its signal to all other stations. The stations “copy” the signal if it is destined for them. Radio systems, such as CB, are examples of broadcast networks. Television and commercial radio are other examples.

Codes

Codes are the symbols used by the machines to direct their actions. The codes are based on binary numbers. Most of us are familiar with the decimal numbering system, consisting of the numbers 0-9. However, machines and the interconnecting channels are designed to support only two signal state : 0 or 1. For example, the binary equivalent of 394 is 110001010. Its decimal value can be established through positional notation:

Each binary digit is called a bit. A group of eight bits make up a byte or octet (although in some systems seven bits comprise a byte).

Binary numbers and codes are represented by several signaling techniques. The data can be represented by simply switching a current on or off; by changing the direction of current flow, or by measuring a current and its associated electromagnetic field. It is also possible to measure the voltage state of the line (such as on/off voltage, or positive or negative voltage) to represent 1s and 0s. Increasingly, optical fiber systems are used to transmit light pulses to represent binary 1s and 0s.

Bit Positions				4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
				3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1
				2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
				1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
8	7	6	5																		
0	0	0	0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI		
0	0	0	1	DLE	DC ₁	DC ₂	DC ₃	RES	NL	BS	IL	CAN	EM	CC		IFS	IGS	IRS	IUS		
0	0	1	0	DS	SOS	FS		BYP	LF	EOB	PRE			SM			ENQ	ACK	BEL		
0	0	1	1			SYN		PN	RS	UC	EOT					DC ₄	NAK		SUB		
0	1	0	0	SP										¢	.	<	'	+			
0	1	0	1	&										!	\$	•)	;	~		
0	1	1	0	-	/										'	%	-	>	?		
0	1	1	1											:	#	@	,	=	"		
1	0	0	0		a	b	c	d	e	f	g	h	i								
1	0	0	1		j	k	l	m	n	o	p	q	r								
1	0	1	0			s	t	u	v	w	x	y	z								
1	0	1	1																		
1	1	0	0		A	B	C	D	E	F	G	H	I								
1	1	0	1		J	K	L	M	N	O	P	Q	R								
1	1	1	0			S	T	U	V	W	X	Y	Z								
1	1	1	1	0	1	2	3	4	5	6	7	8	9						□		

FIG 2: EBCDIC CODE

Bit Positions				7	0	0	0	0	1	1	1	1
				6	0	0	1	1	0	0	1	1
4	3	2	1	5	0	1	0	1	0	1	0	1
0	0	0	0	NUL	DLE	SP	0	@	P	\	p	
0	0	0	1	SOH	DC1		1	A	Q	a	q	
0	0	1	0	STX	DC2	"	2	B	R	b	r	
0	0	1	1	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	.	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	;	K	[k	{	
1	1	0	0	FF	FS	'	<	L	\	l	:	
1	1	0	1	CR	GS	-	=	M]	m	}	
1	1	1	0	SO	RS	.	>	N	^	n	~	
1	1	1	1	SI	US	/	?	0	-	o	DEL	

FIG 3: ASCII / IAS CODE

<i>Typical Speed in Bits</i>	<i>Typical Uses</i>
0-600	Telegraph, older terminals; telemetry
600-2400	Human-operated terminals; personal computers
2400-19,200	Applications requiring fast response and/or throughput; some batch and file transfer applications
32,000-64,000	Digital voice; high-speed applications; some video
64,000-1,544,000	Very high speed for multiple users; computer-to-computer traffic; backbone links for networks; video
greater than 1,544,000	Backbone links for networks; high-quality video; multiple digital voice

TABLE 1: LINK SPEED AND USES

<i>Multiplication Factor</i>	<i>Prefix</i>	<i>Symbol</i>	<i>Meaning</i>
1 000 000 000 000 000 000 = 10 ¹⁸	exa	E	Quintillion
1 000 000 000 000 000 = 10 ¹⁵	peta	P	Quadrillion
1 000 000 000 000 = 10 ¹²	tera	T	Trillion
1 000 000 000 = 10 ⁹	giga	G	Billion
1 000 000 = 10 ⁶	mega	M	Million
1 000 = 10 ³	kilo	K	Thousand
100 = 10 ²	hecto	h	Hundred
10 = 10 ¹	deka	da	Ten
0.1 = 10 ⁻¹	deci	d	Tenth
0.01 = 10 ⁻²	centi	c	Hundredth
0.001 = 10 ⁻³	milli	m	Thousandth
0.000 001 = 10 ⁻⁶	micro	μ	Millionth
0.000 000 001 = 10 ⁻⁹	nano	n	Billionth
0.000 000 000 001 = 10 ⁻¹²	pico	p	Trillionth
0.000 000 000 000 001 = 10 ⁻¹⁵	femto	f	Quadrillionth
0.000 000 000 000 000 001 = 10 ⁻¹⁸	atto	a	Quintillionth

TABLE 2: BASE TEN NUMBERING SYSTEM

AND TERMS

In addition to number representations, data communications systems must also represent another symbols, such as the letters of the alphabet or special characters (like the question mark, ?).

The early codes used in data communications were designed for telegraphic transmission. For example, the antiquated Morse code consists of dots and dashes in a particular sequence to represent characters, numbers and special characters. The dots and dashes represent how long the telegraph operator presses the key on the transmitter to produce an electrical current.

Figures 2 and 3 are examples of two codes in wide use today: the EBCDIC code, developed and sponsored by IBM, and the ASCII code, published by the American National Standards Institute (ANSI). The ASCII code is an international standard, in that it is in conformance with the International Alphabet 5 or IA5. The EBCDIC is widely used primarily because of IBM's position in the industry. EBCDIC is an eight-bit code. The bit position in figure 1 are arranged to show the first four bits at the top of the table with the remaining four bits to the side of the table. The ASCII code is a seven-bit code, although many vendors add an eighth bit for error-checking purpose and is called a parity bit.

Transmission Speeds

Data are transmitted between machines using bit sequences to represent codes. The speed of the data transmission is described in bits per second (or bit/s). Typical speeds of data communications systems are shown in Table 1. Table 2 is also provided to explain commonly used terms. For example 9600 bit/s is often shortened to 9.6 kbit/s (that is, 9.6 kilobit/s).

The data communications world is fairly slow relative to the computer world. For example, a conventional data processing system with disk files attached to computers operates at 10 megabits per second (Mbit/s) and up. The slow speeds stem from the fact that computers usually communicate through the telephone line, which was the most convenient and readily available path when the industry developed computers and began to interface them with terminals and other computers in the 1960s. The telephone channel is not designed for fast transmission between high-speed computers, but for voice transmission between people, which does not require the speed associated with data transmission.

The Error-Laden Channel

The computers (DTEs) can be programmed to communicate with very little ambiguity. (Without question, the task is difficult because machines are relatively "unintelligent.") Yet, data transmitted correctly from the source DTE often arrive incorrectly at the destination (sink) DTE. Due to a myraid of factors, the data can be damaged on the channel en route, such that the binary 1s and 0s representing codes and symbols are misinterpreted by the receiver.

To gain an understanding of the magnitude of the problem, consider some performance measurements of a dial-up connection between two user stations on

long-distance carriers in the United States (MICR86). A 1.2 Kbit/s speed transmission experiences a bit error rate (BER) of 1 in 10^5 (moderate quality) to $1:10^3$ (very poor). In other words, we may expect one bit to be damaged in every 1000 to 100,000 bits transmitted. It takes little imagination to recognize that bit errors are frequent.

Bit errors are also often random; they cannot be predicted. Even through an error may or may not occur in a block of data, studies reveal that a dial-up telephone line experiences an incidence of 0.7 to 142.6 errors in every one thousand blocks sent (each block consisting of 1000 bits).

Is this a problem? It depends on the need of the user. The transmission of certain textual data may not require extensive error-detection efforts, since an occasional corrupted character can be ignored as no more serious than a typing error. On the other hand, an electronic funds transfer system can ill afford an error. The distortion of a decimal place of zero could have either disastrous or serendipitous consequences: disastrous for the individual losing the decimal place, and serendipitous for the individual gaining it.

One solution to the problem is the use of a circuit that is as error-free as possible, and substantial relief results from the use of conditioned, dedicated circuits and other high-quality media. Moreover, circuits using optical fiber offer superior performance over conventional media. Nonetheless, errors will occur, and some method must be used to deal with them – regardless of how frequent or infrequent they are. After the preventive maintenance efforts of using high-quality circuits, and well-designed hardware and software, the next line of defence is to (a) check for transmission errors at the receiver, (b) attempt to correct the error at the receiver, and (c) request the sender to retransmit the damaged data.

The specific component in a data communications system that performs this vital function is called a data link control or a line protocol.

Data Link Control – An Overview

Data link controls (DLCs) are so named because they control the data flow between stations on one physical communications link. All traffic on the link is controlled by the link protocol. For example, if a communications link has several users accessing it, the DLC is responsible for the data to be transported error-free to the receiving user station on the channel (or at least as error-free as possible). Data link controls follow well-ordered steps in managing a communications channel.

- **Link establishment.** Once the DCE has a physical connection to the remote DCE, the DLC (residing usually in the DTE) “handshakes” with the remote DLC logic to ensure that both systems are ready to exchange user data.
- **Information transfer.** User data is exchanged across the link between the two machines. The DLC checks the data for possible transmission errors and sends acknowledgments back to the transmitting machine. In the event an error is detected, the receiver requests the transmitter to retransmit the data.

- Link termination. The DLC relinquishes control of the link (channel), which means no data can be transferred until the link is reestablished. Typically, a DLC keeps a link active as long as the user community wishes to send data to the other stations.

Synchronizing Data Communications Components

The data link control (DLC) assumes the devices on the link are already physically connected and communicating with each other. As brought out earlier, this means the physical level is operating properly, in order for the logical communications to take place.

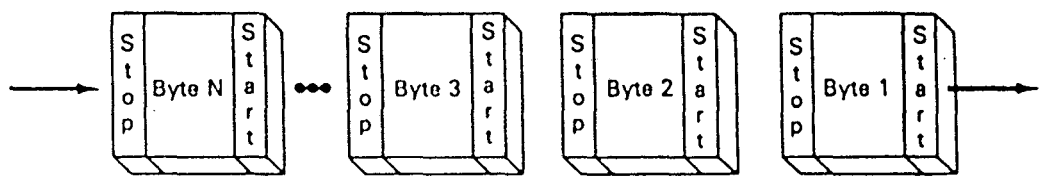
It is a good idea to pause here and consider what is meant by physical communications, because the topic is fundamental to the subject of data communications.

In order for computers and terminals to communicate, they must first notify each other that they are about to transmit data. Second, once they have begun the communication process, they must provide a method to keep both devices aware of the ongoing transmissions. Let us address the first point. A transmitter, such as a terminal or a computer, must transmit its signal so the receiving device knows when to search for and recognize the data as it arrives. In essence, the receiver must know the exact time each binary 1 and 0 is propagating through the communication channel. This requirement means that a mutual time base, or a common clock, is necessary between the receiving and transmitting devices.

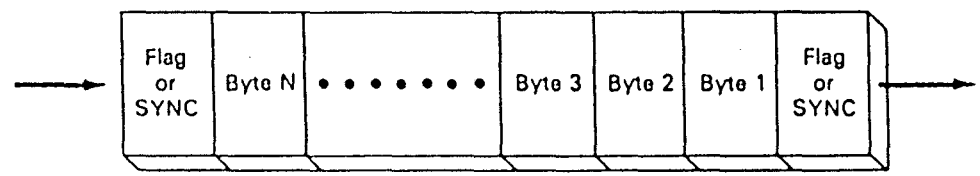
The transmitting machine first forwards to the receiving machine an indication that it is sending data – something like a human saying “hello.” If the transmitter sends the bits across the channel without prior notice, the receiver will likely not have sufficient time to adjust itself to the incoming bit stream. In such an event, the first few bits of the transmission would be lost, perhaps rendering the entire transmission useless. Moreover, the receiver may not be able to “train” itself onto the transmission if it does not detect the first part of the signal.

This process is part of a communications protocol and is generally referred to as synchronization. Connections of short distances between machines often use a separate channel to provide the synchronization. This line transmits a signal that is turned on and off or varied in accordance with pre-established conventions. As the clocking signal on this line changes, it notifies the receiving device that it is to examine the data line at a specific time. It may also adjust the receiver’s sampling clock to enable the receiver to stay accurately aligned on each incoming data bit. Thus, clocking signals perform two valuable functions: (1) they synchronize the receiver onto the transmission before the data actually arrives; and (2) they keep the receiver synchronized with the incoming data bits.

In summary, the clock provides a reference for the individual binary 1s and 0s. The idea is to use a code with frequent signal level transitions on the channel. The transitions delineate the binary data cells (1s and 0s) at the receiver, and sampling logic continuously examines the state of the transmissions in order to detect the bits.



(a) Asynchronous Format



(b) Synchronous Format

FIG 4: ASYNCHRONOUS AND SYNCHRONOUS TRANSMISSION

Receiver sampling usually occurs at a higher rate than the data rate in order to define the bit cells more precisely.

Asynchronous and Synchronous Transmission

We now know that clocking is a major consideration in data communications. Two data formatting conventions are used to help achieve synchronization. These two methods are illustrated in Figure 4. The first approach is called asynchronous formatting. With this approach, each data character has start and stop bits (i.e., synchronizing signals) placed around it. The purpose of these signals are to (a) alert the receiver that data is arriving and (b) give the receiver sufficient time to perform certain timing functions before the next character arrives. The start and stop bits are really nothing more than unique and specific signals which are recognized by the receiving device.

Asynchronous transmission is widely used because the interfaces in the DTEs and DCEs are relatively inexpensive. For example, most personal computers use asynchronous interfaces. Since the synchronization occurs between the transmitting and receiving devices on a character-by-character basis, some allowance can be made for inaccuracies, because the inaccuracy can be corrected with the next arriving character. In other words, a looser timing tolerance is allowed, which translates to lower component costs.

A more sophisticated process is synchronous transmission. It uses separate clocking channels or a self-clocking code. Synchronous formats eliminate the intermittent start/stop signals around each character and provide signals which precede and sometimes follow the user data stream. The preliminary signals are usually called synchronization (sync) bytes, flags or preambles. Their principal function is to alert the receiver of incoming user data. This process is called framing.

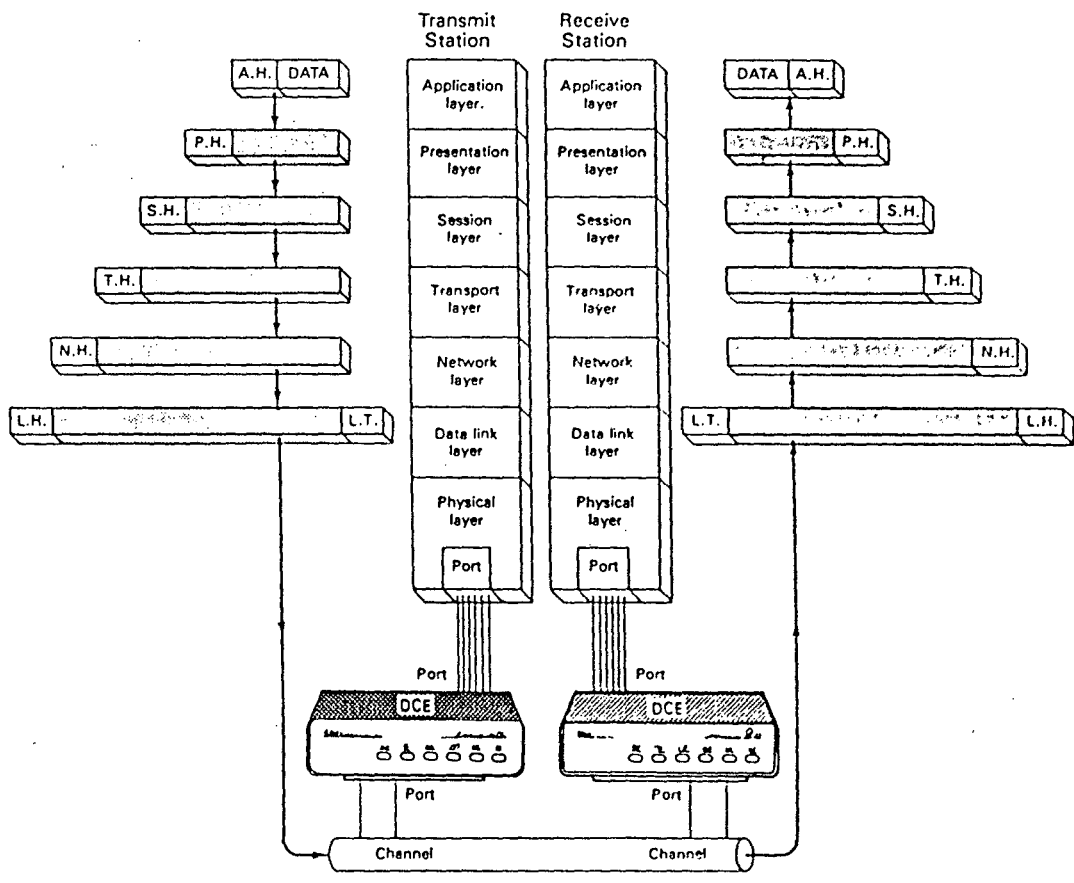
Network Architectures and Layered Protocols

Earlier in this chapter, we learned how machines communicate through established conventions called protocols. Since computer systems provide many functions to users, more than one protocol is required for this support. A convention is also needed to define how the different protocols of the systems interact with each other to support the end user. This convention is referred to by several names: network architecture, communications architecture, or computer-communications architecture.

Whatever terms we use, most systems are implemented with layered protocols, wherein each layer performs a specific function.

Layered protocols are used to meet the following objections:

- Provide a logical decomposition of a complex system into smaller, more understandable parts (layers)
- Provide for standard interfaces between systems; for example, provide standard interfaces between the software or hardware modules that comprise the layers.



AH: Application Header
 PH: Presentation Header
 SH: Session Header
 TH: Transport Header
 NH: Network Header
 LH: Link Header
 LT: Link Trailer

FIG 5: THE OSI LAYERS (SEVEN LAYER MODEL)

- Provide for symmetry in functions performed at each site in the network. Each layer performs the same functions as its counterpart in other machines in the network. This approach greatly simplifies the interfaces between the layers of the network
- Provide a means to predict and control the consequences of any changes made to the network system's logic (software or microcode); the logical decomposition aids in making these changes.
- Provide a standard language to clarify communications between and among network designers, managers, vendors, and users when discussing the logic of network systems.

The OSI Model

The Open Systems Interconnection (OSI) model was developed by several standards organizations and is now a widely used layered functional model. It warrants serious study, because it is becoming a pervasive approach to implementing layered data communications systems.

The stated purpose of the model is to:

- Establish a common basis for standards development
- Qualify products as open by their use of these standards
- Provide a common reference for standards
- Provides standards for communications between systems
- Removes any technical impediment to communicate between systems
- Eliminates the need to describe the internal operation of a single system
- Defines the points of interconnection for the exchange of information between systems
- Narrows the options in order to increase the ability to communicate between systems without expensive conversions and translations. This means different vendors' products can communicate with each other more easily.

In summary, OSI is intended to diminish the effects of the vendor specific mentality that has resulted in each vendor system operating with unique protocols. The OSI approach has relieved end users of the need to purchase expensive and complex protocol converters in order to interconnect and interface various systems.

The OSI layers

The seven OSI layers are depicted in Figure 5. The lowest layer in the model is called the physical layer. The functions within that layer are responsible for activating, maintaining, and deactivating a physical circuit between a DTE and a DCE and providing the clocking signals discussed earlier.

As discussed earlier, the data link layer is responsible for the transfer of data across the link. It delimits the flow of bits from the physical layer. It also provides for the identity of the bits. It usually ensures that the data arrives safely at the receiving DTE. It often provides for flow control to ensure that the DTE does not become overburdened with too much data at any one time. One of its most important functions is to provide for the detection of transmission errors and provide mechanisms to recover from lost, duplicated, or erroneous data.

The network layer specifies the interface of the user into a network, as well as the interface of two DTEs with each other through a network. It also defines network switching/routing and the communications between networks (internet working).

The transport layer provides the interface between the data communications network and the upper three layers (generally part of the user's system). It is the layer that gives the user options in obtaining certain levels of quality (and cost) from the network itself (i.e., the network layer). It is designed to keep the user isolated from some of the physical and functional aspects of the network. It also is the first layer to provide for end-to-end accountability across more than one link.

The session layer serves as a user interface into the transport layer. This layer provides for an organized means to exchange data between users, such as simultaneous transmission, alternate transmission, checkpoint procedures, and resynchronization of user data flow between user applications. The users can select the type of synchronization and control needed from the layer.

The presentation layer provides for the syntax of data; that is, the representation of data. It is not concerned with the meaning or semantics of the data. Its principle role, for example, is to accept data types (character, integer) from the application layer and then negotiate with its peer layer as to the syntax representation (such as ASCII). Thereafter, its functions are limited. The layer consists of many tables of syntax (teletype, ASCII, Videotex, etc).

The application layer is concerned with the support of the end-user application process. Unlike the presentation layer, this layer is concerned with the semantics of data. The layer contains service elements to support application process such as job management, financial data exchange, programming languages, electronic mail, and data base management.

Encapsulation and Decapsulation

Figure 5 depicts how the layers of a network communicate. The vast majority of networks use this approach, so, therefore, we will review this section carefully.

At a transmitting station, user data are presented by a user application to the upper layer (application). This layer adds its protocol control information (PCI) to the user data and usually performs some type of support service to the user. (A more common term for the PCI is header). It then passes its PCI and the user data to the next lower layer, which repeats the process. With the exception of the physical layer, each layer adds a PCI/header. A combination of the PCI and user data is called the protocol data unit (PDU). This concept is somewhat inaccurately called encapsulation

ANSI	American National Standards Institute 1430 Broadway, New York, NY 10018 Telephone: (212) 354-3300
EIA	Electronic Industries Association 2001 Eye Street, Washington DC 20006 Telephone: (202) 457-4966
FED-STD	General Services Administration Specification Distribution Branch Building 197, Washington Navy Yard Washington, DC 20407 Telephone: (202) 472-1082
FIPS	U.S. Department of Commerce National Technical Information Service 5285 Port Royal Road, Springfield, VA 22161 Telephone: (703) 487-4650
CCITT	<i>Outside the United States:</i> General Secretariat International Telecommunications Union Place des Nations, 1121 Geneva 20, Switzerland Telephone: +41 22 99-51-11 <i>In the United States:</i> U.S. Department of Commerce National Technical Information Service 5285 Port Royal Road, Springfield, VA 22161 Telephone: (703) 487-4650
ISO	<i>Outside the United States:</i> International Organization for Standardization Central Secretariat 1 rue de Varembe, CH-1211 Geneva, Switzerland Telephone: +41 22 34-12-40 <i>In the United States:</i> American National Standards Institute (address above)
ECMA	European Computer Manufacturers Association 114 rue du Rhone, CH-1204 Geneva, Switzerland Telephone: +41 22 35-36-34
IEEE	Institute of Electrical and Electronic Engineers 345 East 47 Street, New York, NY Telephone: (212) 705-7900
NBS	National Bureau of Standards Gaithersburg, Maryland Computer Sciences

TABLE 3: STANDARD ORGANISATIONS

(the data from the upper layers are only encapsulated at one end). The only layer that completely encapsulates the data is the data link layer, which adds both a header PCI and a trailer PCI.

The fully encapsulated data are transported across the communications circuit to a receiving station. Here the process is reversed: the data go from the lower layers to the upper layers, and the header created by transmitting peer layer is used by the receiving peer layer to invoke a service function for (a) the transmitting site and/or (b) the upper layers of the receiving site. As the data go up through the layers, the headers are stripped away after they have been used to invoke the services. This process is called decapsulation.

Insofar as possible, the internal operations of the layers are independent of each other. The idea is to reduce complexity and to allow changes to be made in one layer without affecting others. For example, a change to a routing algorithm in one layer should not affect the functions of, say, sequencing, that are located in another layer in the architecture.

Layered network protocols allow interaction between functionally paired layers in different locations. This concept aids in permitting the distribution of functions to remote sites. In the majority of layered protocols, the data unit passed from one layer to another is not altered. The data unit contents may be examined and used to append (i.e., encapsulate) additional PCIs (trailers/headers) to the existing unit. These concepts are key to the OSI model which will be covered later in little more detail.

Standards Organization

The growing acceptance of common conventions and protocols is the result of the efforts of several standards organizations. An overview of these groups is provided here (see Table 3) for their addresses.

The American National Standards Institute (ANSI)

The American National Standards Institute (ANSI) is a national clearing house and coordinating agency for standards implemented in the United States on a voluntary basis. It is a member of the International Organization for Standardization (ISO) and also develops and coordinates standards in data communications for the OSI. It develops standards for encryption activities and office systems. ANSI tries to adopt the ISO standards, but its specifications may differ due to unique aspects of North American systems.

In 1960, ANSI formed X3, a committee to establish standards for data communications, programming languages, magnetic storage media, and the OSI model. It parallels the work of the ISO technical committee (TC) 97.

The Electronic Industries Association (EIA)

The electronic Industries Association is a national trade association that has been active for many years in the development of standards. Its best known standard is EIA-232. The EIA publishes its own standards and also submits proposals to ANSI for accreditation.

The EIA's work is hardware-oriented. The TR-30 Technical Committee Data Transmission, is responsible for EIA-232 (first issued in 1962). TR-30 meets with ANSI X3S3 to ensure that work of the two groups is cohesive. TR-30 is responsible for the physical layer part of the OSI Reference Model; X3S# deals with the data link and network layers.

The European Computer Manufacturers Association (ECMA)

The European Computer Manufacturers Association is dedicated to the development of standards applicable to computer and communications technology. It is not a trade organization, as the name might imply, but a standards and technical review group. It was formed in 1961 as a non-commercial organization to promulgate standards for data processing and communications systems. The ECMA works in close coordination with many of the ISO and the CCITT technical committees and study groups. Initially organized by Compagnie des Machines Bull, the IBM World Trade Europe Corporation, and International Computers and Tabulators Limited, it now includes all European computer manufacturers.

The Institute of Electrical and Electronic Engineers (IEEE)

The Institute of Electrical and Electronic Engineers has been involved for many years in standards activities. It is a well-known professional society with chapters located throughout the world. Its recent efforts in local area networks have received much attention. The IEEE activity addresses local area networks and many other standards as well. The 802 structure; is as follows:

- IEEE 802.1 Higher Layer Interface Standard (HLI)
- IEEE 802.2 Logical Link Control Standard (LLC)
- IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
- IEEE 802.4 Token Bus
- IEEE 802.5 Token Ring
- IEEE 802.6 Metropolitan Area Network (MAN)
- IEEE 802.7 Broadband Technical Advisory Group
- IEEE 802.8 fiber-Optics Technical Advisory Group

<i>Number</i>	<i>Name</i>
I	Definition, operation, and quality of service aspects of telegraph, data transmission, and telematic services (facsimile, Teletext, Videotex, etc.)
II	Operation of telephone network and ISDN
III	General tariff principles, including accounting
IV	Transmission maintenance of international lines, circuits, and chains of circuits; maintenance of automatic and semi-automatic networks
V	Protection against dangers and disturbances of electromagnetic origin
VI	Outside plant
VII	Data communications networks
VIII	Terminal equipment for telematic services (facsimile, Teletext, Videotex, etc.)
IX	Telegraph networks and terminal equipment
X	Languages and methods for telecommunications applications
XI	ISDN and telephone network switching and signalling
XII	Transmission performance of telephone networks and terminals
XV	Transmission system
XVII	Data transmission over the telephone network
XVIII	Digital networks, including ISDN

TABLE 4: CCITT STUDY GROUPS

The International Organization for Standardization (ISO)

The International Organization for Standardization (ISO) is a voluntary body. It consists of national standardization organizations from each member country. The activities of ISO are principally from the user committees and manufacturers in contrast to the carriers that are represented in CCITT. The American National Standards Institute (ANSI) is the primary U.S representative to the ISO.

Technical committee 97 (TC97) deals with information technology. As such, its activities affect many of the products and systems that are used in the industry. The ISO documents are designated as IS (International Standard), DIS (Draft International Standard), and DP (Draft Proposal).

International Telegraph & Telephone Consultative Committee (CCITT)

The International Telegraph and Telephone Consultative Committee is a member of the International Telecommunications Union (ITU), a treaty organization formed in 1865. The ITU is now a specialized body within the United Nations. CCITT sponsors a number of recommendations dealing primarily with data communications networks, telephone switching standards, digital systems, and terminals. The State Department is the voting member on CCITT from the United States, although several levels of membership are permitted. For example, the recognized private operating agencies (RPOA) are allowed to participate at one level (such as the regional Bell Operation Companies).

The CCITT's recommendations (also known as Standards) are very widely used. Its specifications are republished every four years in a series of books that take more than two feet on a bookshelf. The books covering each four-year period can be identified by the colour of their covers. The 1960 books were red; 1964, blue; 1968, white; 1972, green; 1976, orange; 1980, yellow; and, in 1984, once again red. The 1988 blue books use about four feet of shelf space! The CCITT Study Groups are summarized in Table 4.

U.S.National Committee (to CCITT)

In the United States, the State Department is the principal member of CCITT, and a National Committee is the coordinating group for U.S participation in CCITT. Moreover, advisory committees coordinate contributions to CCITT Study Groups. These U.S. CCITT Study Groups prepare actions for CCITT consideration :

Study Group A:	U.S.Government of Regulatory Policies
Study Group B:	Telegraph Operations
Study Group C:	Worldwide Telephone Network
Study Group D:	Data Transmission

Several government organizations have important roles in developing international standards. As mentioned earlier, the State Department is the United States voting member of CCITT. The National Communications System (NCS) is a consortium of federal agencies that have large telecommunications capabilities. The

NCS works very closely with other organizations such as the EIA, ISO, and CCITT. One of its jobs is to develop federal input to the international standards organizations, and NCS is using the OSI architecture for its work. Indeed, the majority of government organizations are using OSI.

The National Bureau of Standards

The National Bureau of Standards (NBS) is also very active in international standards committees. Currently, it is working on the upper layers of the OSI standard. It is also responsible for the Federal Information Processing Standards (FIPS). A large number of ANSI documents are incorporated into the FIPS standards.

NBS is also responsible for the publication of GOSIP (U.S Government Open System Interconnection Protocols), a specification that defines the specific protocols to be used by U.S Government agencies.

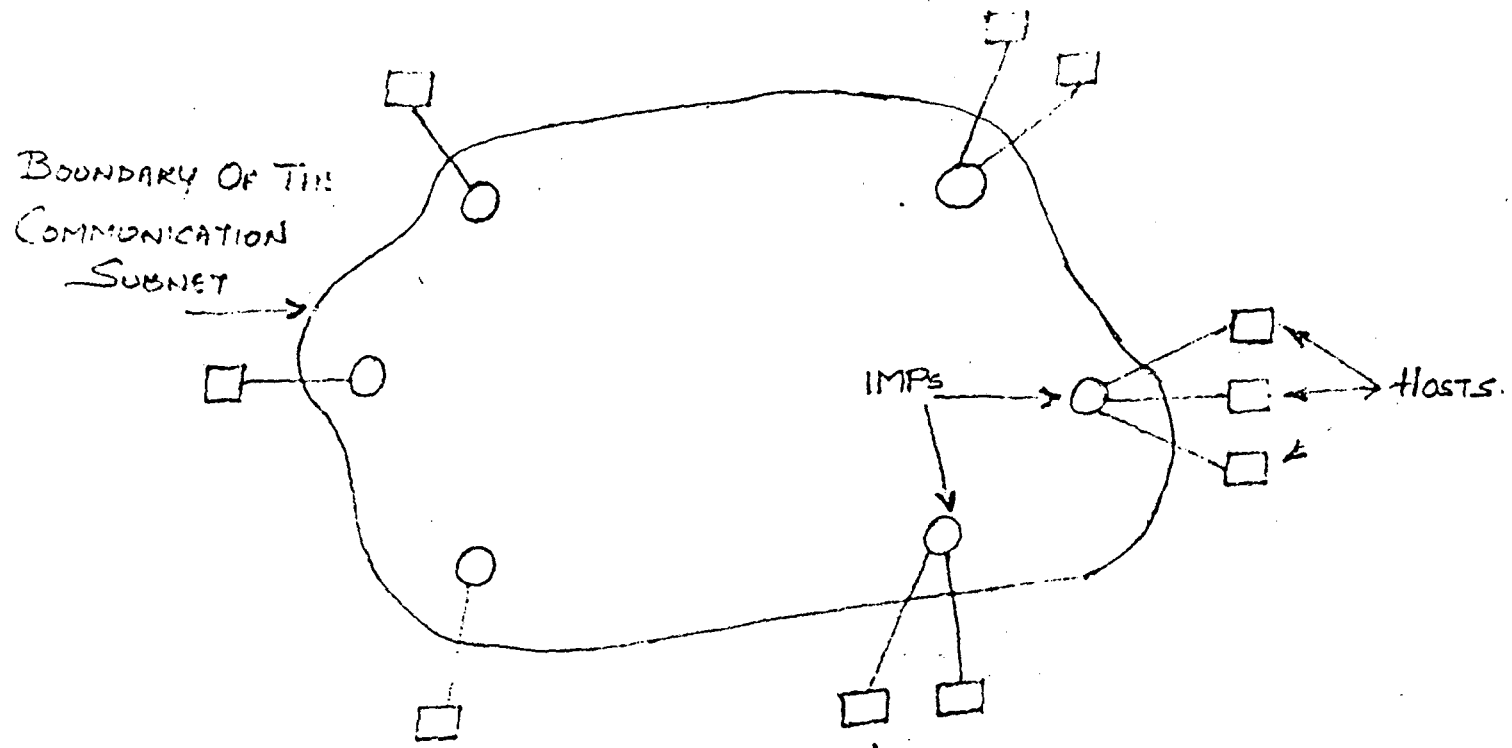


FIG. 6. RELATION BETWEEN HOSTS AND THE SUBNET

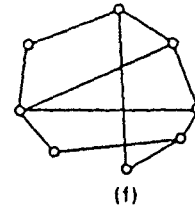
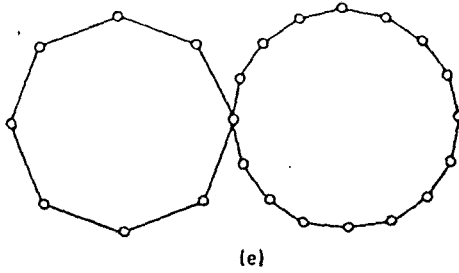
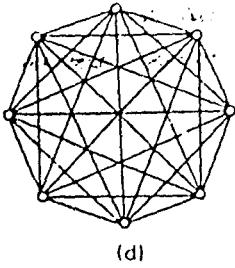
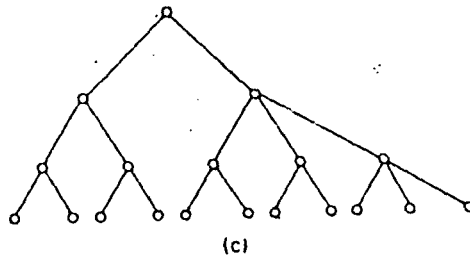
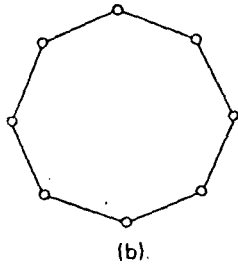
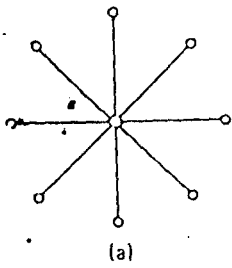


FIG 7 : SOME POSSIBLE TOPOLOGIES FOR

A POINT-TO-POINT SUBNET

(a) STAR (b) RING (c) TREE (d) COMPLETE

(e) INTERSECTING RINGS (f) IRREGULAR

NETWORK STRUCTURE AND ARCHITECTURE

In any network there exists a collection of machines intended for running user (i.e., application) programs. We will follow the terminology of one of the first major networks, the **ARPANET**, and call these machines **hosts**. The term **end system** is sometimes also used in the literature. The hosts are connected through the **communication subnet**, or just **subnet** for short. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener. By separating the pure communication aspects of the network (the subnet) from the application aspects (the hosts), the complete network design is simplified.

In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements. Transmission lines (also called **circuits**, **channels**, or **trunks**) move bits between machines.

The switching elements are specialized computers used to connect two or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line to forward them on. Again following the original **ARPANET** terminology, we will call the switching elements **IMP's (Interface Message Processors)**, although the term **packet switch node**, **intermediate system**, and **data switching exchange** are also commonly used. Unfortunately there is no consensus on terminology here; lot of books on this subject seem to be using different names. The term "**IMP**" is probably as good as any. In this model, shown in Fig. 6, each host is connected to one (or occasionally several) IMP's. All traffic to or from the host goes via its IMP.

Broadly speaking, there are two types of designs for communication subnet:

1. Point-to-point channels
2. Broadcast channels

In the first one, the network contains numerous cables or leased telephone lines, each one connecting a pair of IMP's. If two IMPs do not share cable nevertheless wish to communicate, they must do this indirectly, via other IMPs. When a message (in the context of the subnet often called **packet**), is sent from one **IMP** to another via one or more intermediate IMP's, the packet is received at each intermediate IMP in its entirety, stored there until the required output line is free, and then forwarded. A subnet using this principle is called **point-to-point, store and store-and-forward or packet-switched** subnet. Nearly all wide area networks have store-and-forward subnets.

When a point-to-point subnet is used, an important design issue is what the IMP interconnection topology should look like. Fig. 7 shows several possible topologies. Local network that are designed as such usually have a symmetric topology. In contrast, wide area networks typically have irregular topologies.

The second type of communication architecture uses broadcasting. Most local area networks and a small number of wide area networks are of this type. In a local area network, the IMP is reduced to single chip embedded inside the host, so there is always one host per IMP, whereas in a wide area network there may be many hosts per IMP.

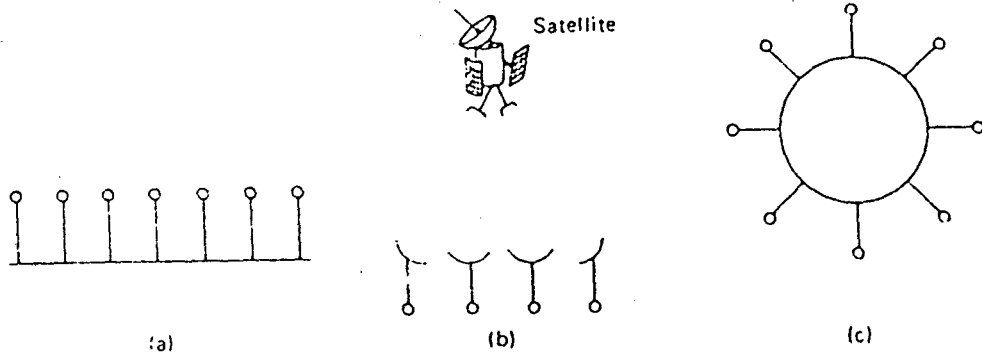


FIG 8: COMMUNICATION SUBNETS USING BROADCASTING

(a) BUS (b) SATELLITE OR RADIO (c) RING

Broadcast systems have a single communication channel that is shared by all machines of the network. Packets sent by any machine are received by all the others. An address field within the packet specifies for whom it is intended. Upon receiving a packet, a machine checks the address field. If the packet is intended for some other machine, it is just ignored.

As an analogy, consider someone standing at the end of a corridor with many rooms off it and shouting "Watson, come here. I want you." Although the packet may actually be received (heard) by many people, only Watson responds. The others just ignore it.

Broadcast systems generally also allow the possibility of addressing a packet to *all* destinations by using a special code in the address field. When a packet with this code is transmitted, it is received and processed by every machine on the network. Some broadcast systems also support transmission to a subnet of the machines, something known as **multicasting**. A common scheme is to have all addresses with a high order bit set to 1 be reversed to multicasting. The remaining $n-1$ addresses bits form a bit map corresponding to $n-1$ groups. Each machine can 'subscribe' to any or all of the $n-1$ groups. If a packet with, say, bits x , y and z set to 1 is transmitted, it is accepted by all machines subscribing to one or more of those three groups.

Figure 8 shows some of the possibilities for broadcast subnets. In a bus or cable network, at any instant one machine is the master and is allowed to transmit. All other machines are required to refrain from sending. An arbitration mechanism is needed to resolve conflicts when two or more machines may be centralized or disturbed.

A second possibility is a satellite or ground radio system. Each IMP has an antenna through which it can send and receive. All IMP's can hear the output from the satellite, and in some cases they can also hear to the upward transmissions of their fellow IMP's to the satellite as well.

A third broadcast system is the ring. In a ring, each ring propagates around on its own, not waiting, for the rest of the packet to which it belongs. Typically, each bit circumnavigates the entire ring in the time it takes to transmit a few bits, often before the complete packet has even been transmitted. Like all other broadcast systems, some rule is needed for arbitrating simultaneous access to the ring.

Broadcast subnets can be further divided into static and dynamic, depending on how the channel is allocated. A typical static allocation would be to divide up time into discrete intervals, and run a round robin, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to say during its allocation slot, so some systems attempt to allocate the channel dynamically (i.e., on demand)

Dynamic allocation method for a common channel are centralized or decentralized. In the centralized channel allocation method, there is a single entity, for example a bus arbitration unit, which determines who goes next. It might do this by accepting requests and making a decision according to some internal algorithm. In the

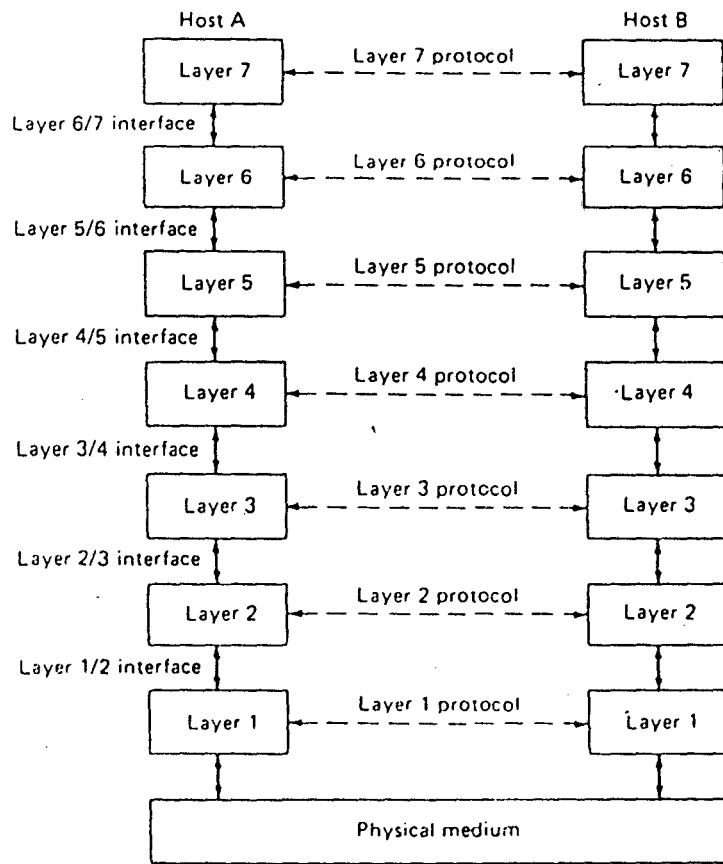


FIG 9: LAYERS , PROTOCOLS AND INTERFACES

decentralized channel allocation method, there is no central entity; each machine must decide for itself whether or not to transmit.

Network Architecture

Modern computer networks are designed in a high structured way. In the following sections we examine the structuring technique in some detail.

Protocol Hierarchies

To reduce the design complexity, most networks are organized as a series of **layers** or **levels**, each one built upon its predecessor. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. However, in all networks, the purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented.

Layer *n* on one machine carries conversation with layer *n* on another machine. The rules and conventions used in this conversation are collectively known as the layer *n* **protocol**, as illustrated in Fig. 9 for a seven layer network. The entities comprising the corresponding layers on different machines are called **peer processors**. In other words, it is the peer processors that communicate using the protocol.

In reality, no data are directly transferred from layer *n* on one machine to layer *n* on the another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. In fig. 9 virtual communication is shown by dotted lines and physical communication by solid lines.

Between each pair of adjacent layers, there is an interface. The interface defines which primitive operations and services the lower offers to the upper one. When networks designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well understood functions. In addition to minimizing the amount of information that must be passed between layers, clean-cut interfaces also make it simpler to replace the implementation of one layer with a completely different implementations (i.e., all the telephone lines are replaced by satellite channels), because all that is required of the new implementation is that it offers exactly the same set of services to its upstairs neighbor as the old implementation did.

The set of layers and protocols is called the **network architecture**. The specification of the architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation not the specification of the interfaces are part of the architecture because these are hidden away inside the machines and are not visible from the outside. It is not even necessary that the interfaces of all machines in a network be the same, provided that each machine can correctly use all the protocols.

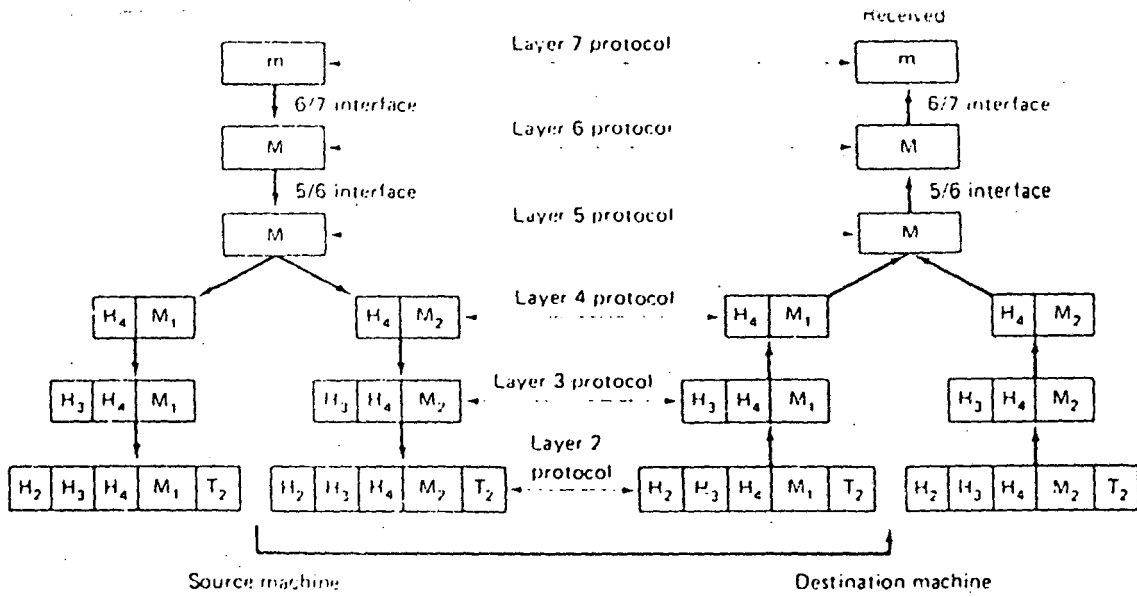


FIG 10: EXAMPLE : INFORMATION FLOW SUPPORTING VIRTUAL COMMUNICATION IN LAYER SEVEN

An analogy may help explain the idea of multilayer communication. Imagine two philosophers (peer processes in layer 3), one in Kenya and one in Indonesia, who want to communicate. Since they have no common language, they each engage a translator (peer processes in layer 2), each of whom in turn contacts an engineer (peer processes in layer 1). Philosopher 1 one wishes to convey his affection for *oryctolagus cuniculus* to his peer. To do so, he passes his message (in Swahili) across the 2/3 interface, to his translator, who might render it as “**I like rabbits**” or “**J’aime des lapins**” or “**Ik hou van konijnen**”, depending on the layer 2 protocol.

The translator then gives the message to his engineer for transmission, by telegram, telephone, computer network, or some other means, depending on what the two engineers have agreed on in advance (the layer 1 protocol). When the message arrives, it is translated into Indonesian and passed across the 2/3 interface to philosopher 2. Note that each protocol is completely independent of the other ones as long as the interfaces are not changed. The translators can switch from French to Dutch at will, provide that they both agree, and neither changes his interface with either layer 1 or layer 3.

Now consider a more technical example: how to provide communication to the top layer of the seven-layer network in Fig. 10. A message, m , is produced by a process running in layer 7. The message is passed from layer 7 to layer 6 according to the definition of the layer 6/7 interface. In this example, layer 6 transforms the message in certain ways (e.g., text compression), and then passes the new message, M , to layer 5 across the layer 5/6 interface. Layer 5, in the example does not modify message from being handled to layer 6 while layer 6 is busy handing a series of outgoing message to layer 5).

In many networks, there is no limit to size of messages accepted by layer 4, but there is a limit imposed by layer 3. Consequently, layer 4 must break up the incoming message s to smaller units, pretending a **header** to each unit. The header includes control information, such as sequence numbers, to allow layer 4 on the destination machine to get the pieces back together in the right order if the lower layers do not maintain sequence. In many layers, header also contain sizes, times and other control fields.

Layers 3 decides which of the outgoing lines to use, attaches its own headers, and passes the data to layer 2. Layer 2 adds not only a header to each piece, but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for the layers below n are passed up to layer n .

The important thing to understand about Fig.10. is the relation between the virtual and actual communication and the difference between protocols and interfaces. The peer progresses in layer 4, for example, conceptually think of their communication as being “horizontal”, using the layer 4 protocol. Each one is likely to have a procedure called *SendToOtherSide* and a procedure *GetFromOtherSide*, even though these procedures actually communicate with lower layers across the $\frac{3}{4}$ interface, not with the other side.

The peer process abstraction is crucial to all network design. Without this abstraction technique, it would be difficult, if not impossible, to partition the design of the complete network, an unmanageable problem, into several smaller, manageable, design problems, namely the design of the individual layers.

Design Issue for the Layers

Some of the key design issues that occur in computer networking are present in several layers. Below, we will briefly mention some of the more important ones.

Every layer must have a mechanism for connection establishment. Since a network normally has many computers, some of which have multiple processes, a means is needed for a process on one machine to specify to whom it wants to establish a connection. As a consequence of having multiple destinations, some form of addressing is needed in order to specify a specific destination.

Closely related to the mechanism for establishing connections across the network is the mechanism for terminating them once they are no longer needed.

Another set of design decisions concerns the rule for data transfer. In some systems, data only travel in one direction (simplex communication). In others they can travel in either direction, but not simultaneously (half-duplex communication). In still others they travel in both directions at once (full-duplex communication). The protocol must also determine how many logical channels the connection corresponds to, and what their priorities are. Many networks provide at least two logical channels per connection, one for normal data and one for urgent data.

Error control is an important issue because physical communication circuits are not perfect. Many errors-detecting and error-correcting codes are known, but both ends of the connection must agree on which one is being used. In addition, the receiver must have some way of telling the sender which messages have been correctly received and which have not.

Not all communication channels preserve the order of messages sent on them. To deal with a possible loss of sequencing, the protocol must make explicit provision for the receiver to allow the pieces, to be put back together properly. An obvious solution is to number the pieces, but this solution still leaves open the question of what should be done with pieces that arrive out of order.

Another problem that must be solved at several levels is the inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and then reassembling messages. A related issue is what to do when processes insist upon transmitting data in units that are so small that sending each one separately is inefficient. Here the solution is to gather together several messages heading toward a common destination into a single large message, and dismember the large message at the other side.

When it is inconvenient or expensive to set up a separate connection for each pair of communicating processes, the underlying layer may decide to use the same

connection for multiple, unrelated conversations. As long as this multiplexing and demultiplexing is done transparently, it can be used by any layer. Multiplexing is needed in the physical layer, for example, where all the traffic for all connections has to be sent over at most a few physical units.

When there are multiple paths between source and destination, a route must be chosen. Sometimes this decision must be split over two or more layers. For example, to send data from London to Rome, a high level decision might have to be made to go via, France or Germany based on their respective privacy laws, and a low level decision might have to be made to choose one of the many available circuits based on current traffic.

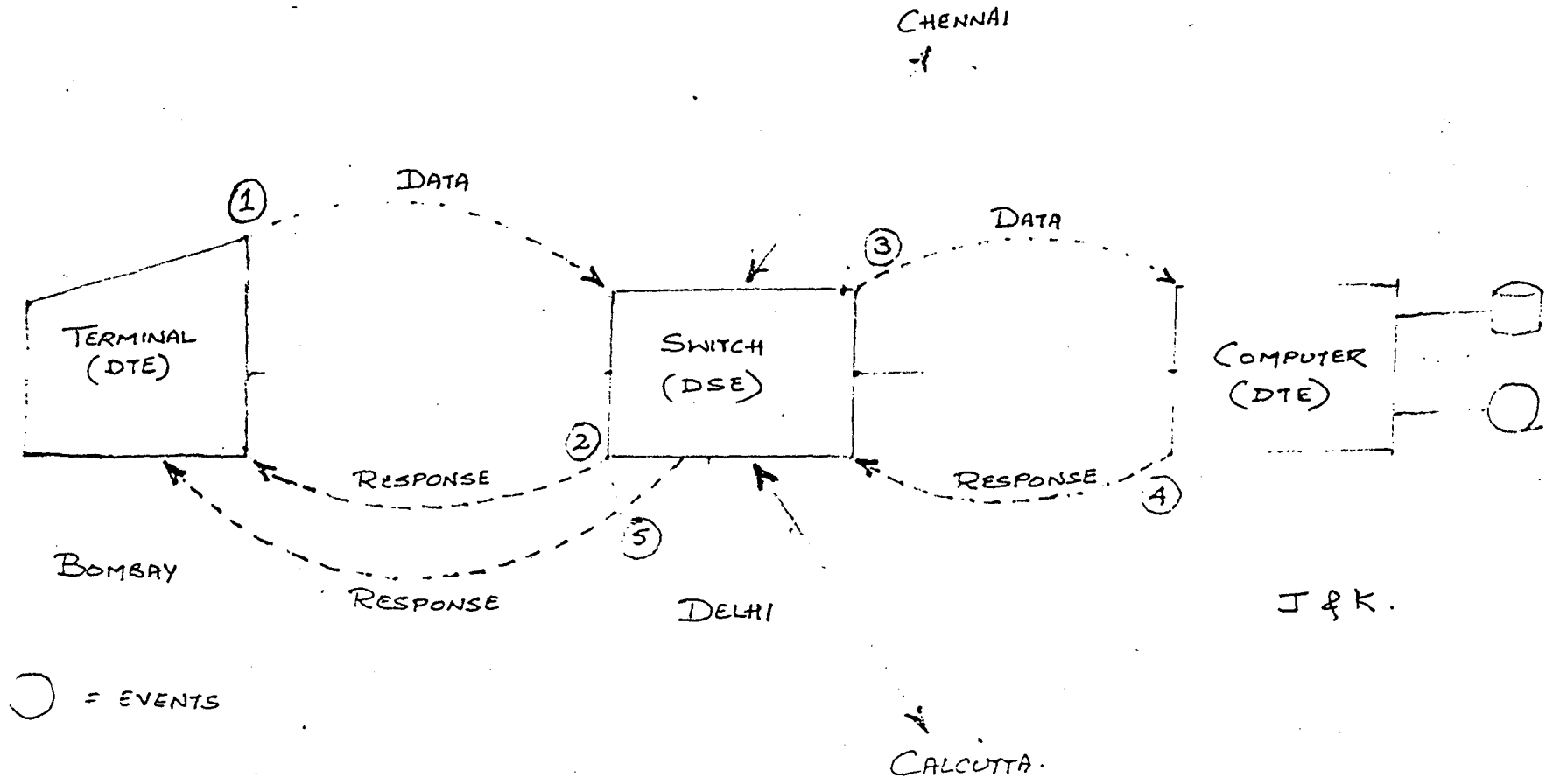


Fig. 10(a) TRAFFIC CONTROL AND ACCOUNTABILITY.

COMMUNICATIONS BETWEEN AND AMONG COMPUTERS AND TERMINALS

Introduction

This chapter provides a general description of how DTEs exchange data. The term protocol is used to describe the procedures and logic for this process.

Traffic Control and Accountability

Typically, several different protocols cooperate to manage the communications. For Example, one protocol is responsible for controlling the flow of the traffic on each channel; a second protocol usually selects the best channel (among several) for the first protocol to use. The first protocol is classified as a link or line protocol (also a data link control). The second protocol is called a switching or routing protocol. Additional protocols are also involved and are explained in later chapters.

Figure 10 illustrates several important points about network communications. A terminal (DTE) at Bombay is to transmit data to a remote computer (DTE) located in Jammu & Kashmir. The transmission goes through an intermediate point, a computer located in Delhi. The Delhi site performs routing and switching functions, since it also has lines to Chennai and Calcutta, and thus fits our definition of data switching equipment (DSE). The most common approach in network communications is to pass the data, like a baton in a relay race, from site to site until they finally reach the destination. One important aspect of the process is in event 2, where Delhi sends an acknowledgment of the data received to the Mumbai terminal. This acknowledgement means the Delhi site has checked for possible errors occurring during the transmission of the frame, and as best the Delhi site can determine, the data have been received without any errors. It so indicates by transmitting another frame back on the return path indicating acceptance.

The data communications industry uses two terms to describe the event 2 response. The terms ACK denotes a positive acknowledgement; the term NAK represents a negative acknowledgement. A NAK usually occurs because the transmission (ie., the signal representing the data) is distorted due to faulty conditions of the channel (lighting storm, etc.). The frame in event 2 to Mumbai will either be an ACK or a NAK. In the event of an error in the transmission, the terminal in Mumbai must receive a negative acknowledgement (NAK) so it can retransmit the data. It is also essential that the process shown in events 1 and 2 are completed before event 3 occurs. If Delhi immediately transmitted the data to Jammu and Kashmir before performing the error check, Jammu & Kashmir could possibly receive erroneous data. (ACKs and NAKs are represented by the codes discussed in Appendix A).

If the Mumbai site receives an ACK in event 2, it assumes the data have been received correctly in Delhi, and the communications system in Mumbai can purge this message from its queue. (The application process often saves a copy on disk or tape for accounting, audit, or security reasons).

Continuing the process in events 3 and 4, assume that an ACK is returned from J&K to Delhi. The end user in Mumbai may assume through event 2 that the data arrived in J&K. A false sense of security could result, because event 2 indicates only that the data arrived safely in J&K. If the data are lost between the Delhi and J&K sites (it can happen), the Mumbai terminal assumes no problem exists. This scenario provides no provision for an end-to-end acknowledgment. If an end user wishes to have absolute assurance that the data arrived at the remote site, event 5 is required. Upon receiving event 4 at the Delhi site, Delhi sends another acceptance (ACK) to Mumbai. In other words, event 5 says that J&K also accepts the data.

End-to-end protocols add overhead and costs. Consequently, end users may not choose to have end-to-end acknowledgment with low-priority, unimportant traffic. However, if the data are important – for instance, a transfer of \$20 million to a J&K bank over a funds-transfer network – a prudent user would want to have absolute assurance that the funds arrived and were posted to an account. In this case, the user would want event 5 to occur.

The preceding statements point out another aspect of a data communications system. Even though it usually provides for all five of the transactions, the actual posting of the funds transfer to a bank account ordinarily is not performed by the communications software. The applications process is responsible for the posting and data base update. Therefore, be aware that event 5 means the communication system in J&K received the data correctly. In turn, it passes the data to an application process for the data base update. If the data base problem or an applications software failure prevents the funds transfer from being posted, it is the responsibility of the applications process to send an indicator back to the terminal user in Mumbai. It is rarely the responsibility of the communications system to perform the application-to-application accountability of traffic.

7H-8872

Checking for Errors

The most common method used today for error checking is cyclic redundancy checking (CRC). The technique uses a constant derived from a CRC polynomial [An algebraic expression consisting of two or more terms; $(x-1) \times (x^{15} - x - 1)$] to divide the constant into a binary representation of a data field (such as the contents of a frame). The quotient of the division is discarded, but the remainder is retained and used at the receiver to check for transmission errors.

At the receiving end, the transmitted CRC field (usually 16 bits) is compared to the answer of an identical CRC calculation. If they are consistent, the frame is considered to be error-free.

Wide Area and Local Networks

The previous discussion explains in general terms how the DTEs communicate directly. The concern with errors is evident. The use of error-checking techniques and ACKs/NAKs are necessary to ensure the integrity of user data.

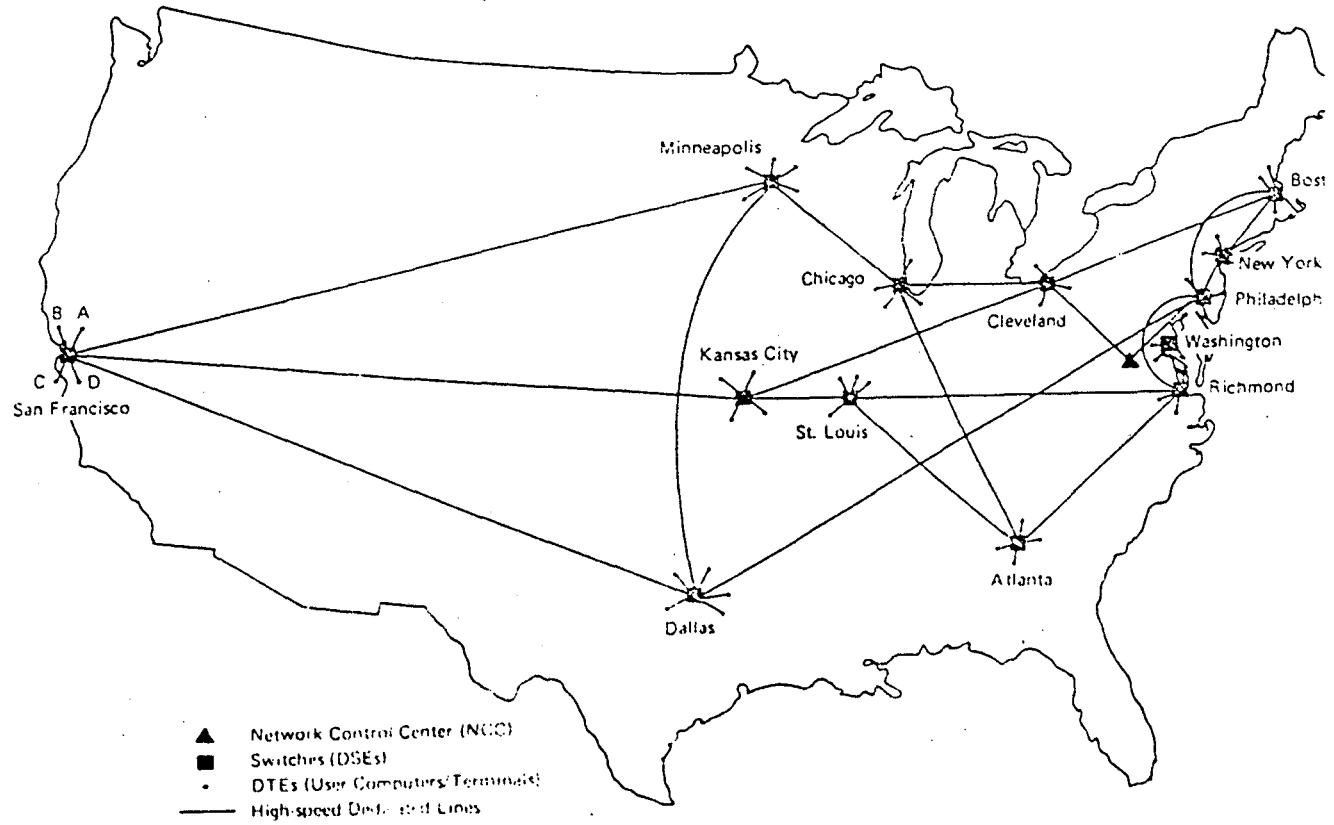


FIG 11: A TYPICAL WIDE AREA NETWORK

Practically speaking, the user data may not warrant such careful attention; each data character may not have to arrive error-free. A one-bit error in the transmission of a business letter (electronic mail) would distort only one character of the entire letter – better than the performance of most typists. Moreover, the scenario depicted in Figure 10 assumes the communications channel is unreliable, experiencing frequent errors. Such is the case with a conventional voice-oriented telephone line, but other communications channels are of better quality and are more reliable. For example, optic fiber channels are of significantly higher quality than metallic circuits.

If the user does not require each character to be received correctly and/or the communications channel is reliable, the expense to perform the elaborate functions depicted in Figure 10 may not be warranted. The issue is important under the simple arrangement in the figure. It is equally important when the user ties into a more complex network with several layers of protocols such as the wide area network (WAN) in Figure 11.

This network consists of DSEs (switching computers) connected together by high-speed, leased channels (for example, 56 kbit/s lines). Each DSE uses a protocol responsible for routing data and providing support to the end-user computers and terminals attached to it. The DTE support function is often called a PAD (packet assembly/disassembly). The DSE acts as the PAD into and out of the network for the DTEs. The network control center (NCC) is responsible for the efficient, reliable operations of the network.

A portion of Figure 11 is expanded in Figure 12. Notice the DTEs' varied connections into the PAD/switch:

- A. A user-site computer is connected to the DSE through an asynchronous protocol, with dial-up analog lines into a DSE-dedicated port (a port reserved exclusively for the user).
- B. A user-site front-end processor is connected to the DSE through a synchronous protocol, with dedicated 56 kbit/s digital lines using data service units (DSUs).
- C. A user-site asynchronous terminal (or personal computer) is connected to the DSE, with dial-up analog lines into a DSE non dedicated port.
- D. A user site has a dedicated DSE on premises connected into the network using private network 56 kbit/s digital lines with data service units (DSUs).

Figure 11 and 12 depict a topology for a wide area network (WAN). This type of network is noted for the following characteristics:

- Wide area links are usually provided by an interexchange carrier (such as MCI or AT&T), at a monthly cost for leased lines and usage cost for dial-up lines.

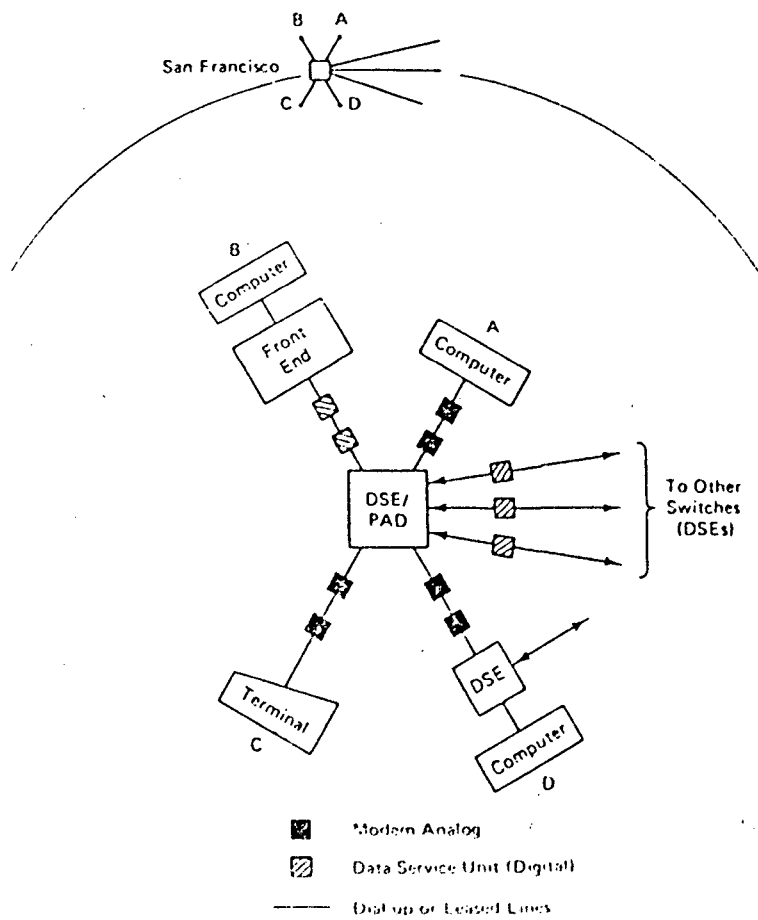


FIG 12: A DSE/PAD

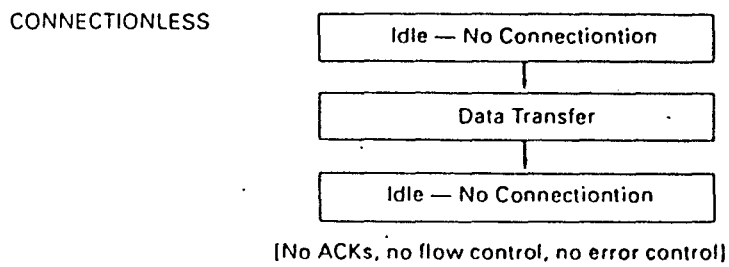
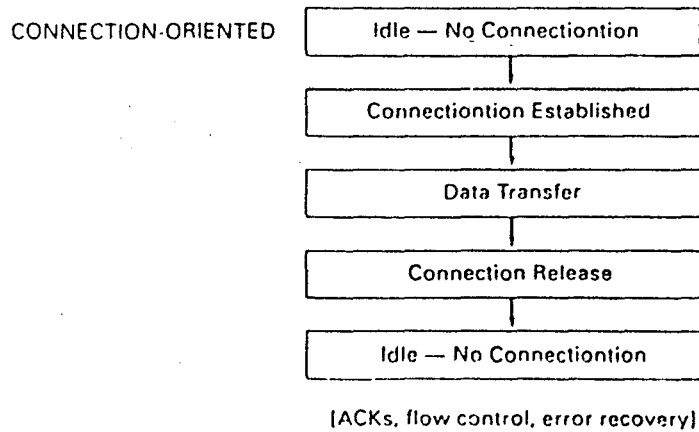


FIG 13: A CONNECTION - ORIENTED
AND CONNECTIONLESS NETWORK

- The links are relatively slow (1200 kbit/s to 1.544 Mbit/s). DTE connections into DSE are usually slower (150 bit/s to 19.2 kbit/s).
- DTEs and DSEs are located several miles to several hundred miles apart.
- Links are relatively error-prone (if using conventional telephone circuits).

The local area network (LAN) is significantly different from a wide area network. The LAN is one of the fastest growing sectors in the communications industry. The LAN is noted for the following characteristics:

- Links are usually owned by the user organization.
- Links operate on very high-speed lines (1 Mbit/s to 400 Mbit/s). DTEs are attached to network with lower speed channels (600 bit/s to 56 kbit/s).
- DTEs are located closely together, usually within a building or plant. A DSE is used for switching in some configurations, but not as frequently as in a WAN.
- Links are of better quality than WAN channels.

Because of these major differences between wide area and local area networks, their topologies often take different shapes. A WAN structure tends to be more irregular due to the need to multidrop and/or multiplex terminals, computers, and switches onto the lines. Since the channels are leased on a monthly basis (at considerable expense), a user organization strives to keep the lines fully used. This requirement often creates the need to “snake” the channel through a geographical area, connecting the various DTEs, wherever they may be located, to one channel. Consequently, a WAN topology often has an irregular shape.

The LAN owner is not as concerned with maximum utilization of the channels, which are inexpensive in comparison to their bit-rate capacity (and LAN bottlenecks usually occur in the software, anyway). Consequently, clever multidropping and multiplexing schemes are not as critical in a local environment as they are on wide area network. Moreover, since local networks usually reside within a building, the topology inherently tends to be more ordered and structured, taking such shapes as the bus, ring, or star configurations.

CONNECTION-ORIENTED AND CONNECTIONLESS NETWORKS

The DTEs in Figures 11 and 12 communicate through the network DSE/PAD by one of two techniques. One technique is connection-oriented; another technique is connectionless. As illustrated in Figure 13, a connection-oriented network is one in which no logical connection initially exists between the DTEs and the network. The network connection between the two DTEs is in an idle state. In order for computers or terminals to communicate through a connection-oriented network, they must go through connection establishment, which is called a “handshake”. Once a connection is established, the data-transfer state is entered; the user data are exchanged through a pre-established protocol. The DTEs subsequently perform a connection release, after which they return to the idle condition.

The connection oriented network provides a substantial amount of care for the user data. The procedure requires a specific acknowledgement that the connection is established, or the network informs the requesting DTE if the connection is not established. Flow control (i.e., making certain that all the data arrive correctly, in order, and do not saturate the DSEs and DTEs in the various parts of the network) is also required for the network. Error checking is performed, as well as error recovery. Connection-oriented networks maintain a continuous awareness of all DTE-to-DTE sessions and attempt to assure that user data are not lost in the network. The care provided by this type of network requires considerable overhead because of the many support functions.

The connectionless (also called datagram) network goes directly from an idle condition (the two DTEs are not logically connected to each other) into a data transfer mode, followed directly by the idly condition. The major difference is the absence of a connection-establishment phase and a connection-release phase. Moreover, a connectionless network has no network-wide acknowledgements, flow control or error recovery, although these services may be provided on a link-by-link basis. Obviously, the connectionless network involves less overhead.

Connection-oriented networks are often compared conceptually to the telephone system (either dial-up or leased lines). The caller knows when a connection is made because is talking to someone at the other end of the line. The connectionless network is comparable to mailing a letter. A letter is placed into the postal system with the assumption it will arrive at its destination. The letter usually arrives safely, but the letter writer never knows it. The post office sends nothing back to tell the letter writer that the letter arrived. The end recipient of the letter must initiate a response indicating acceptance, usually in the form of another letter, which in communications parlance is called a higher-level protocol.

The tradeoffs between connection-oriented and connectionless network center around overhead required versus functions provided. A connection-oriented network is rich in functions, yet these functions add to the costs of the system. In contrast, the connectionless network requires less overhead because it is limited in the support it provides the user application process. The issue is really one of deciding where transmission and reception integrity are assured – within or outside of the network.

Connection-oriented networks have dominated computer wide area networks (WAN) because of the inherent error-prone nature of the telephone system. Consequently, systems using the telephone channel perform many functions to ensure data integrity is maintained between the communicating devices. A connectionless network makes more sense with a local area network (LAN). A LAN channel is usually within one building and privately owned. Based on its technology, a LAN is much less error-prone. It is relatively unusual for data to be distorted on a LAN channel. A typical telephone channel connected with a WAN experiences an error rate in the approximate range of $1:10^3$ to $1:10^5$ – one bit in error to every 1,000 to 100,000 bits transmitted. A LAN typically experiences an error rate of approximately $1:10^8$. The error performance between a WAN and a LAN differs by several orders of

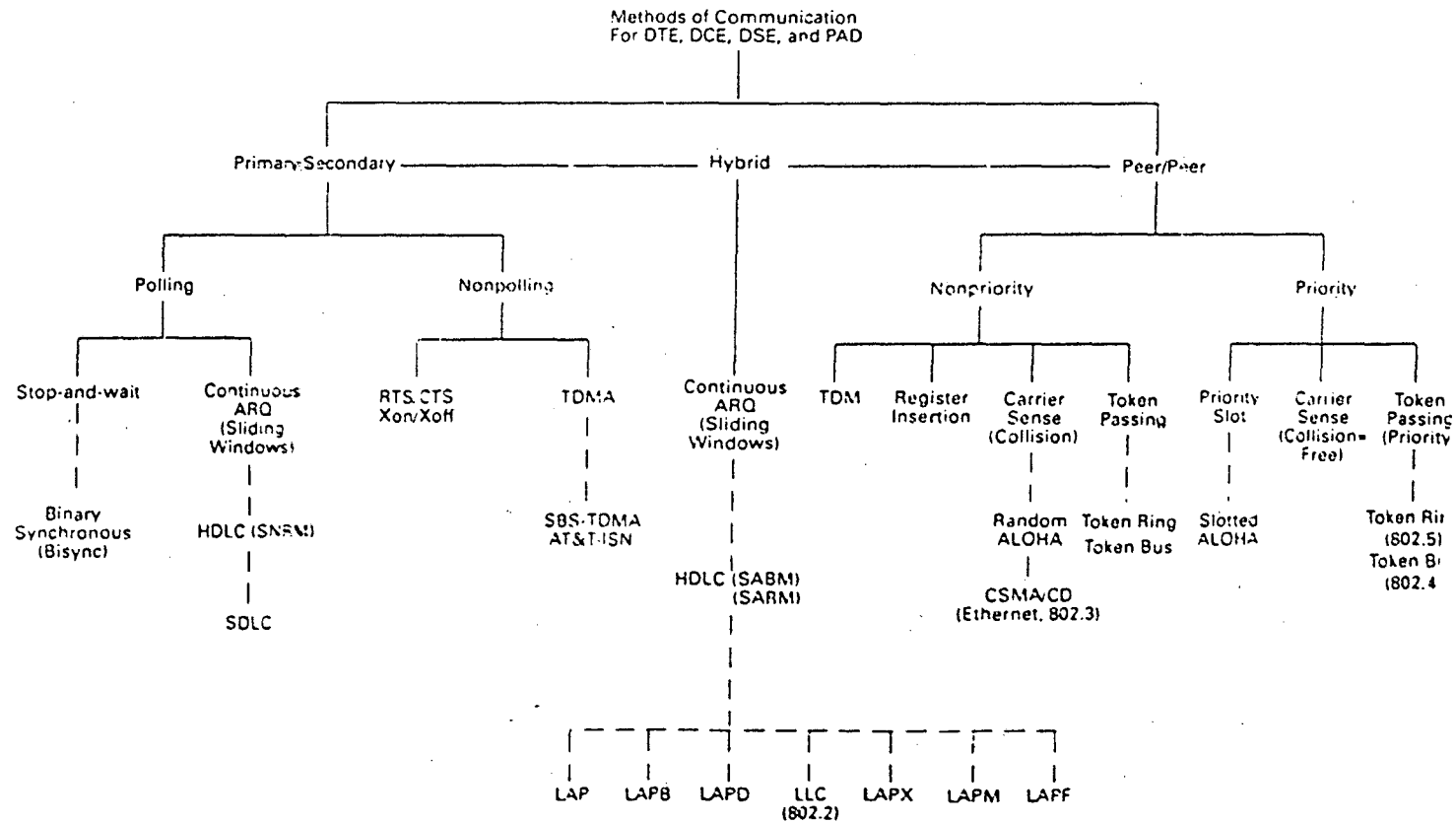


FIG 14: PROTOCOL CLASSIFICATION

magnitude. Consequently, it may make little sense in a connectionless network (especially if it is a LAN) to perform the expensive overhead options of flow control, error control, and recovery, because the rare occurrence of an error is not worth all the expense of avoiding it.

Of course, a valid response to this rationale could be, “yes, that may be true, but on the rare occasion that the error does occur, it may be catastrophic to the organization.” To allow for this contingency, one practical alternative is to “push” error control up into the application process (or a higher-level protocol), instead of having the lower-level communications protocols deal with it. An error-prone network should not present erroneous data to the application, because the application frequently is forced to devote resources to the error-correction capabilities in the application obviates having the functions in the communication system, which translates to a simpler, less expensive network.

Many networks today are designed to use connectionless protocols within the network and in relay systems between networks. End-to-end integrity is provided by logic (an upper layer) outside the network. This layer is called the transport layer.

Classification of Communications Protocols

DTEs communicate with each other by the techniques depicted in Figure 14. The DCE, PAD and DSE also use these methods to communicate with each other and the DTEs. The classification tree is not meant to be all-encompassing, but is used to provide a structure from which to gain an understanding of communications techniques.

The majority of the protocols depicted in the figure are called line (like or channel) protocols or data link controls (DLS). They are so named because they control the traffic flow between stations on one physical communications channel.

Data-link protocols manage all communications traffic on a channel. For example, if a communications port had several users accessing it, the DLC would be responsible for ensuring all users had their data transported error-free to the receiving node on the channel. The DLC is generally unaware that the data on the channel are from multiple users.

Data-link controls follow well-ordered steps in managing a communication channel:

- Link establishment. Once the DCE has a physical connection to the remote DCE, the DLC “handshakes” with the remote DLC logic to ensure both systems are ready to exchange user data.
- Information transfer. User data are exchanged across the link between the two machines. The DLC checks all data for possible transmission errors and sends acknowledgements back to the transmitting machine.

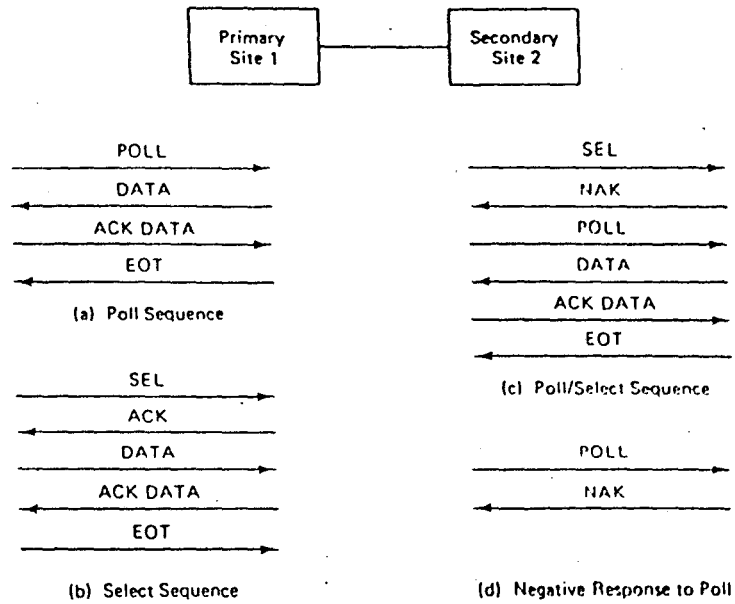


FIG 15: POLLING/SELECTION SYSTEMS

Link termination. The DLC relinquishes control of the link (channel), which means no data can be transferred until the link is reestablished. Typically, a DLC keeps a link active as long as the user community wishes to send data across it.

A widely used approach to managing the communications channel is through a primary/secondary (sometimes called master/slave) protocol. This technique designates one DTE, DCE, or DSE as the primary station on the channel. The primary station (usually a computer) controls all the other stations and dictates when and if the devices can communicate. Primary/secondary systems are implemented with several specific technologies depicted in Figure 14.

The second major approach is through a peer-to-peer protocol. This technique has no primary station and typically provides for equal status to all stations on the channel. However, nodes may not have equal access to the network, since they can have pre-established priority over others. Nonetheless, the absence of a primary site usually provides for an equal opportunity to use network resources. Peer/peer systems are often found in local area networks (LANs) with ring, bus, and mesh topologies, and in certain hybrid systems as depicted in the figure.

Polling/Selection systems

The first example of a primary/secondary system is polling/selection, usually shortened to polling. The configuration in Figure 15 shows a host computer at site 1 and a terminal at site 2. There could be many other configurations (for example, a multidrop line or a ring topology). Polling/selection works the same conceptually with computers linked to other computers; it is possible to have primary/secondary computers, as well as terminals.

Polling/selection systems revolve around two commands, Poll and Select. The purpose of the Poll command is to transmit data to the primary site. The purpose of the Select command is just the opposite: to transmit data from the primary site to the secondary site. Select commands are no longer needed on the newer protocols, because the master site reserves resources and buffers at the receiver during link establishment, thereby sending data at the discretion of the master node.

A hierarchical network typically exists as an ordered form of a primary/secondary relationship. Poll and Select are the principal commands needed to move data to any site on a channel or in the network. Let us examine how this is accomplished, referring to Figure 15(a). First, a Poll command is sent from the master site to secondary site 2. The poll says, in effect: "Secondary site 2, have you data for me?" The poll is sent to secondary site 2 and if data are waiting to be transmitted, they are sent back to the polling site. The primary site checks for errors and sends an ACK if the data are correct or a NAK if they are incorrect. These two events of data and ACK/NAK may occur many times until the secondary site has no more data to send. The secondary station must then send an indicator that it has completed its transmission, such as the end-of-transmission code (EOT), or a bit in a control field.

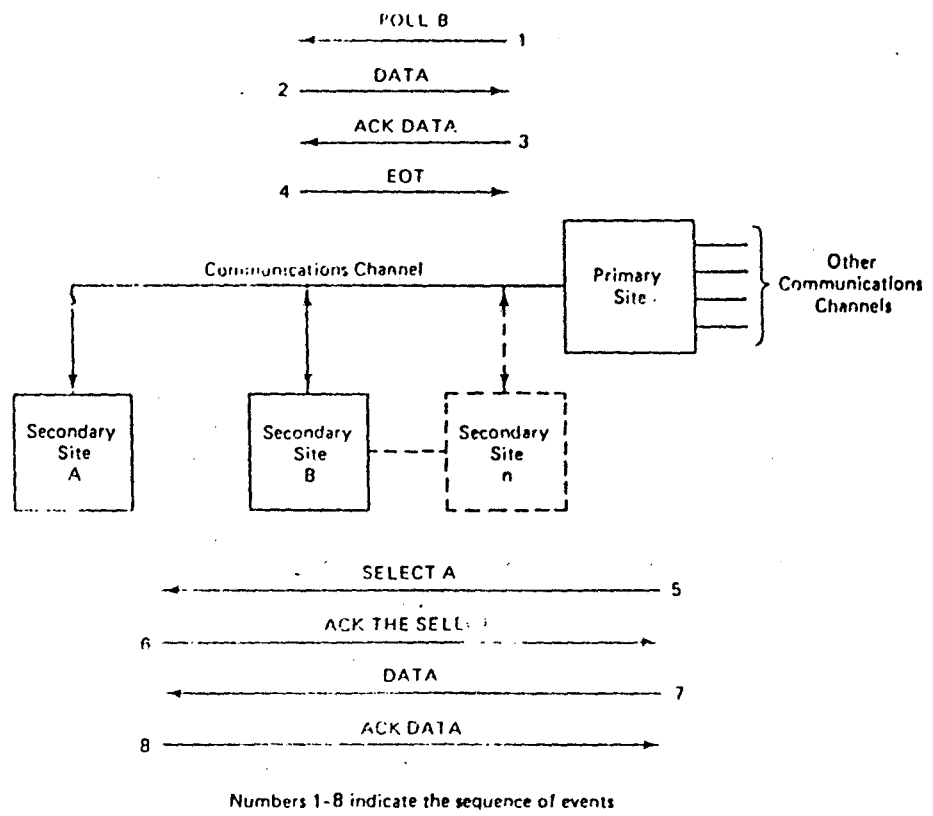


FIG 16: POLLING SELECTION SYSTEM

The Select command is illustrated in Figure 15(b). Select means: "Secondary site 2, I am selecting you because I have data for you. Can you receive?" The ACK to the select means: "Yes, I am available and ready to receive your data." The data are transmitted, checked for errors, and acknowledged. (As stated earlier, newer systems reserve resources at links establishment and assume the receiver can indeed receive the data. Therefore, no selects are required with this approach.) The process can repeat itself. Eventually, an EOT control indicator is transmitted, meaning: "I have no more traffic to send."

Figure 15(c) shows the complexities of polling/selection. It is called the select/poll sequence. Notice the select is transmitted to secondary site 2, but the site responds with a negative response (NAK) to the select. This dialogue means: "Secondary site 2, I have data for you, can you receive?" The response is: "No, I cannot." There are a number of reasons the site cannot receive. It may be busy doing other things or it may have no memory (buffer space) available to receive data. As another example, it may have data to send to the primary site. The poll/selection systems handles the problem by the primary site initiating a poll, which allows the secondary site to send data and clear its buffers.

The last sequence of operations (Figure 15(d) shows what happens in the polling/selection network when a poll is issued to the secondary site and it responds negatively. In this case, the system uses a NAK to indicate a negative response to a poll. Simply stated, it means: "Secondary site 2, have you data for me?" The NAK means: "No, I do not." In newer systems, the indication of a willingness to receive or transmit is called a Receive Ready; unwillingness is called a Receive Not Ready.

A disadvantage of a polling/selection system is the number of negative responses to polls, which can consume precious resources on the channel. This overhead is especially evident in systems without multiplexers or terminal cluster controllers. These devices can accept a general poll to any device, scan their attached devices for an active request, and transmit to the primary.

Another approach to decreasing the effect of polling overhead is to use dynamic polling/selection tables. If a device continues to be polled and does not respond after a certain number of attempts, its priority is moved down within a polling table. Therefore, it is serviced less and polled fewer times. The nonresponding station is dropped to a lower priority, and those devices which have been responding positively to the poll are moved up in the priority table. It is also conceivable to design the table to provide multiple entries in the table by the same device. Station A might be polled, then station C, then A again, because A has been busy and responded positively to polls. Dynamic polling/selection eliminates some of the overhead found in the conventional static polling/selection systems.

Figure 16 shows the polling/selection systems used to manage traffic between two DTEs on the same channel. DTE B wishes to communicate with DTE A. In order for this transmission to take place, event 1 requires that the primary site poll DTE B. The data are not sent to A directly, but to the primary site. The data are checked for errors, an acknowledgement is sent in event 3, and an EOT is sent in event 4. When

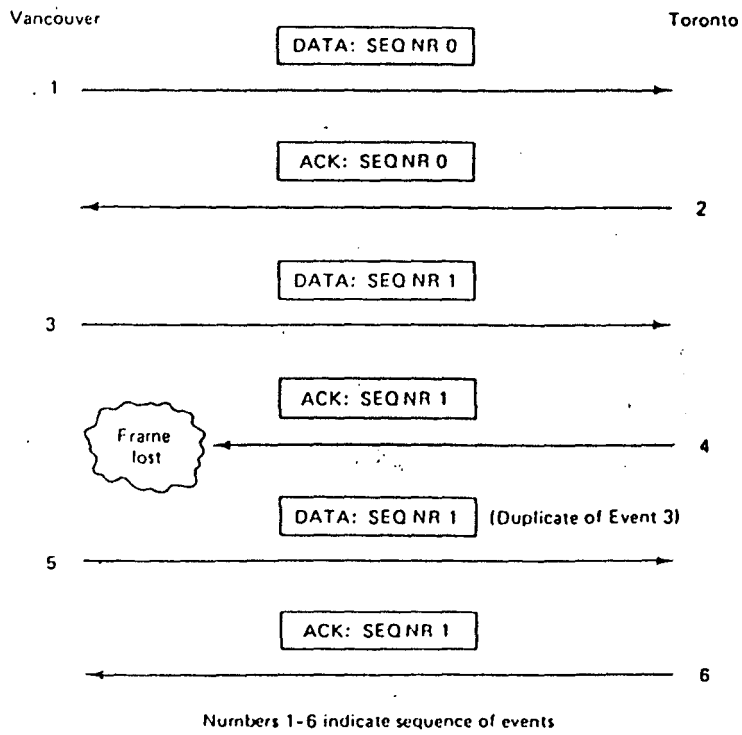


FIG 17: STOP AND WAIT SEQUENCING

the data arrive at the master site, they can then be relayed (onto the same channel) to DTE A. This is accomplished with (as you can probably guess) the select. (Remember, the poll moves data into the host, the select moves data out of the host.) Event 6 shows the select is ACK'd, the data are sent across the link in event 7, and they are acknowledged in event 8, completing the accountability. The user data in event 2 are a copy of the user data shown in event 7.

These examples illustrate once again the hierarchical aspect of the primary/secondary system. All traffic comes into and goes out of the primary host. The hierarchical topology presents some potential bottleneck problems, since all traffic is managed by one device. The configuration also has some reliability problems – if the primary site goes down, the entire network is lost. Hierarchical systems should provide for some form of backup in the event the primary site is lost.

Selective and Group Polling

Selective polling is the technique we have just examined and is a common mechanism for multidrop communications links. Group polling is more common on a ring or loop topology, or on a line with cluster controllers. Both techniques use a primary node to issue the poll command. The multidrop topology has each poll addressing a specific station on the channel. The station responds with data or a negative response to the poll. The ring configuration uses a group or broadcast poll to all stations on the channel. Each station can use the poll and respond accordingly, pass the poll (and perhaps data) to the next station on the loop. A station may “piggyback” its transmission onto the data passing around the ring.

Stop-and-wait Polling/Selection

One of the simplest and oldest forms of polling/selection is the stop-and-wait technique. It is so named because a DTE transmits a frame and waits for a reply. It is inherently half-duplex (two-way alternate) because the transmissions are in both directions, but only in one direction at a time. Stop-and-wait is a widely used approach because it is relatively inexpensive; the software is simple, with little logic involved. Most stop-and-wait systems use sequencing, in which stations use sequence numbers to maintain accountability and to control the flow of traffic.

Figure 17 shows a situation in which data are transmitted with a sequence number of 0 from station A (in Vancouver) to station B (in Toronto). Sequence numbers are added to each transmission. As illustrated in the figure, the data are checked in Toronto; the computer responds with an ACK (event 2). The ACK uses a 0 in the header to account for the data sent to it. Upon receipt of the ACK, station A in Vancouver then transmits another data frame, and this time it changes the sequence number to a 1 (event 3). The data are checked for errors at Toronto and an ACK of 1 is sent.

Some protocols do not actually require the sending stations to insert a sequence number. Rather, the sequence number is inferred, alternately changing from

a 1 to a 0. The transmitting station simply “flips” a counter from 1 to 0 as it sends a frame, and then looks for the corresponding ACK of 1 or 0.

The reason for the sequence number can be seen in the next data flow. Traffic can be lost in the network because of complexities of the traffic pattern, logic problems (“bugs”), or failed components. The data may also be lost because the frame is damaged en route, for example, by being routed over a microwave communications link through a rainstorm. The frame with ACK 1 can be distorted so severely that the Vancouver site receives “noise” on the line and the traffic is indecipherable.

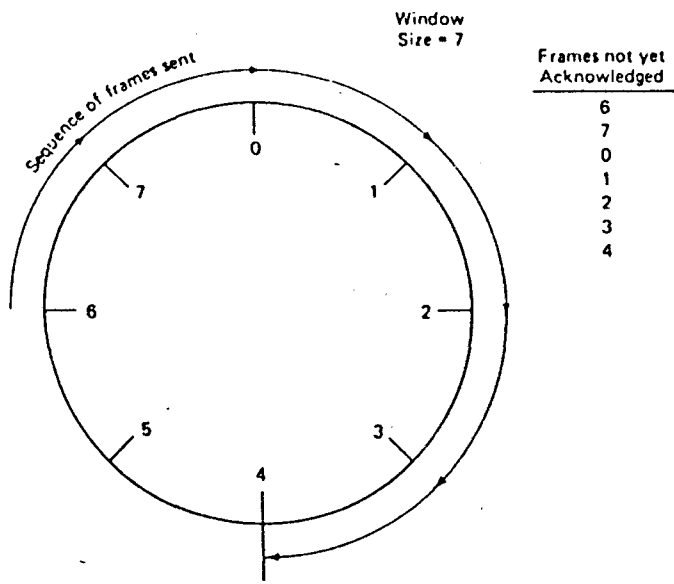
In such an event, the Vancouver site performs a timeout. A timeout means that Vancouver, after not receiving a reply to its transmission within a given period, retransmits the data. The data transmitted in event 5 are exactly the same as in event 3. If a sequence number did not exist to identify the duplicate traffic, Toronto may not detect the duplicate frame. Indeed, the Toronto site might send the duplicate transmission to a data base, in which case a redundant update could be applied to the data. However, Toronto is expecting a different sequence number, a 0. Therefore, Toronto discards the duplicate data and retransmits the ACK 1 to complete the accountability (event 6).

Continuous ARQ (Sliding Windows)

Another example of polling systems is the Continuous ARQ (automatic request for repeat) technique. Continuous ARQ is so named because a station is allowed to request automatically a retransmission from another station. This approach can utilize full-duplex (two-way simultaneous) transmission, which allows transmission in both directions between the communicating devices. Because Continuous ARQ has several advantages over the stop-and-wait, half-duplex system, it has seen increasing use in the industry during the past several years. We introduce the topic of Continuous ARQ here.

Continuous ARQ devices use the concept of transmitting and receiving windows. A window is established on each link to provide a reservation of resources at both DTEs. These resources may be the allocation of specific computer resources or the reservation of buffer space for the transmitting DTE. In most systems, the window provides both buffer space and sequencing rules. During the initiation of a link session (handshake) between the DTEs, a window is established. If DTE A and DTE B are to communicate with each other, DTE A reserves a window for B, and B reserves a window for A. The windowing concept is necessary to full-duplex protocols because they entail a continuous flow of frames into the receiving site without the intermittent stop-and-wait acknowledgements. Consequently, the receiver must have a sufficient allocation of space to handle the continuous incoming traffic.

The windows at the transmitting and receiving site are controlled by state variables, which is another name for a counter. The transmitting site maintains a send state variable $[V(S)]$. It is the sequence number of the next frame to be transmitted. The receiving site maintains a receive state variable $[V(R)]$, which contains the number that is expected to be in the sequence number of the next frame. The $V(S)$ is



OUTSTANDING: Frames 6 through 4

FIG 18: WINDOW MANAGEMENT

incremented with each frame transmitted and placed in the send sequence field in the frame.

Upon receiving the frame, the receiving site checks for a transmission error and the send sequence number with its $V(R)$. If the frame is acceptable, it increments $V(R)$ by one, places it into a receive sequence number field in an acknowledgement (ACK) frame and sends it to the original transmitting site to complete the accountability for the transmission.

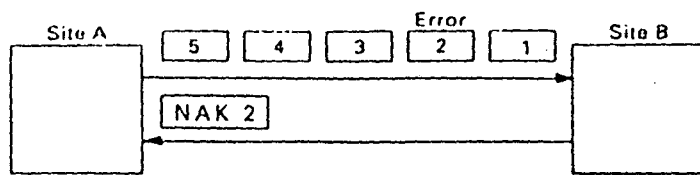
If the $V(R)$ does not match the send sequence number in the frame, or an error is detected, something has gone awry. After a timeout occurs, a NAK (with the receiving sequence number containing the value of $V(R)$) is sent to the original transmitting site. Most protocols call this NAK a Reject as a Selective Reject. The $V(R)$ value informs the transmitting DTE of the next frame that it is expected to send. Since the transmitter has sent a frame with this value, it knows something is wrong and must then reset its $V(S)$ and transmit the frame whose sequence number matches the value of $V(R)$.

Many systems use the numbers of 0 through 7 for $V(S)$, $V(R)$, and the sequence numbers in the frame. Once the state variables are incremented through 7, the numbers are reused beginning with 0. Because the numbers are reused, the DTEs must not be allowed to send a frame with a sequence number that has not yet been acknowledged. For example, the protocol must wait for frame number 6 to be ACK'd before it uses a $V(S)$ of 6 again. The process is shown in Figure 18. Frames 6 through 4 are not yet acknowledged. If another frame was sent with a sequence number of 6, the corresponding ACK of 6 would not indicate which frame 6 was acknowledged.

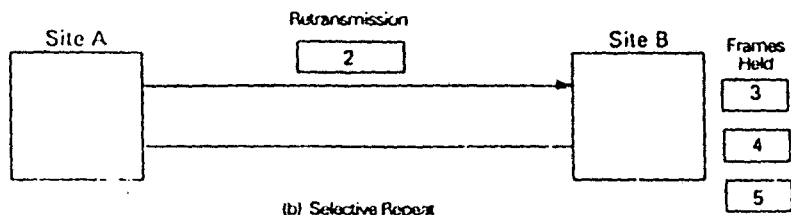
The use of numbers 0-7 permits seven frames to be outstanding before the window is "closed" Even though 0-7 gives eight sequence numbers, the $V(R)$ contains the value of the next expected frame, which limits the actual outstanding frames to 7.

Window size is an important design consideration. The larger the window, the more frames that can be transmitted without a response from the receiver. Yet the larger window size also means that the receiver must allocate more resources and larger buffers to handle the incoming transmissions. The line protocols used in the industry today typically allocate a window of seven at session-initiation time, which means that a transmitting DTE is allowed to send seven frames without receiving an acknowledgment back to it. However, in the event the seven frames are sent without any acknowledgment, the transmitting station's window is closed to its session partner, the receiver DTE. Window closing is necessary to prevent the transmitting station from saturating the receiver, which could overflow buffers and result in lost data. Window closing also provides the master a means to service the other sessions on the channel.

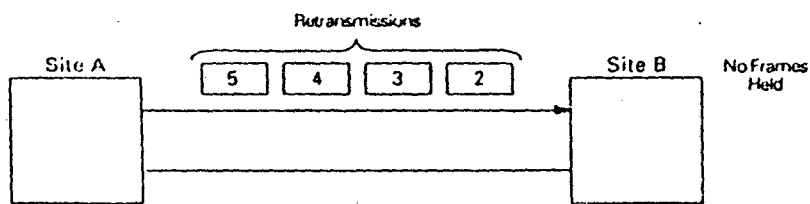
When the receiving station transmits a positive acknowledgement (ACK) to the transmitter, the transmitter's window is opened. For example, if the receiver transmits four ACKs back to the transmitter, then the transmitter's window is opened by four frames.



(a) Frames 1 Through 5 Transmitted with an Error in Frame 2



(b) Selective Repeat



(c) Go back N

FIG 19: RETRANSMISSION OF ERRORS

The goal of line protocols is to keep the window open for all user sessions on the line. In so doing, the transmitting application and receiving applications are more likely to experience fast response time. The Continuous ARQ protocols also are designed to keep the expensive communications channel as busy as possible.

The concepts of sliding windows are relatively simple, yet it should be realized that under a primary/secondary system, the primary DTE is tasked with efficient transmission, data flow, and response time between itself and all the secondary sites attached to it. The primary host must maintain a window for every station with which it has a connection. It must ensure that the windows stay open and manage traffic in a manner to keep the stations as busy as possible. This is no small feat, considering that polling and selection systems may have hundreds of terminals or computers attached to a master computer.

Continuous ARQ protocols, if using a window of seven, require at least three bits to provide the windowing and sequencing operations. (For example, the binary number 111 equals 7 in base 10). Sequencing is required for these systems because more than one frame may be outstanding on the channel at any one time. Therefore, the receiver must indicate to the transmitter the positive acknowledgement (ACK) or the negative acknowledgment (NAK) of each specific frame. As noted earlier, the acknowledgement is accomplished through the use of sequence numbers. For example, if the transmitting site sends frames 1,2,3 and 4 to the receiver, the receiver is required to indicate through ACKs and NAKs the specific frames that were received correctly or incorrectly.

In this regard, continuous ARQs provide several notable advantages over the stop-and-wait systems. One advantage is called inclusive acknowledgement. Using the above example, the receiver could send an ACK of 5. ACK of 1,2,3 and 4 are not transmitted. The ACK of 5 means "I have received and acknowledge everything up to and including 4; the next frame expected should have a 5 in its send sequence field." It is evident from this simple example that continuous ARQ protocols with inclusive acknowledgement can reduce considerably the overhead involved in the ACKs. In this example, one ACK acknowledges 4 frames, considerably better than the stop-and-wait systems, in which an ACK is required for every transmission.

Polling Continuous ARQ protocols are used extensively with wide area networks (WANs). Consequently, error control is an important feature in the systems. A considerable amount of the logic found in polling Continuous ARQs is devoted to error detection and resolution. Continuous ARQ uses one of two methods to detect and retransmit erroneous data. The first, Selective Repeat, requires that only the erroneous transmission be retransmitted. The second approach, Reject, requires that not only the erroneous transmission be repeated, but all frames that were transmitted behind it as well. Selective Repeat and Reject are illustrated in figure 19.

Both techniques have advantages and disadvantages. Selective Repeat provides better line utilization, since the erroneous frame is the only retransmission. However, as shown in Figure 19(b), site B must hold frames 3,4 and 5 to await the

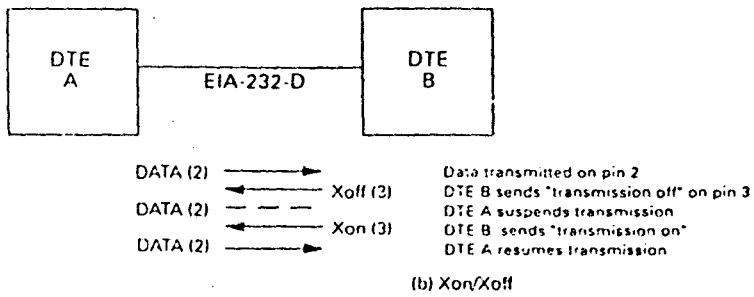
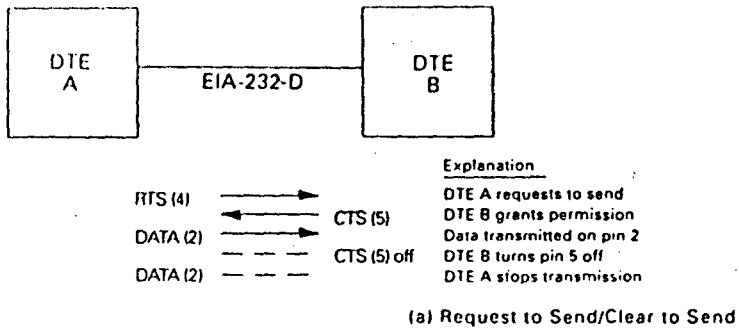


FIG 20: SIMPLE NONPOLLING SYSTEMS

retransmission of frame 2. Upon its arrival, frame 2 must be inserted into the proper sequence before the data are passed to the end-user application. The holding of frames can consume precious buffer space, especially if the DTE has limited memory available and several active links.

Reject is a simpler technique. Once an erroneous frame is detected, the receiving station discards all subsequent frames in the session until it receives the correct retransmission. Reject requires no frame queuing and frame resequencing at the receiver. However, its throughput is not as high as Selective Repeat, since it requires the retransmission of frames that may not be in error.

Nonpolling Systems

The classification of network communications protocols can be done by branching our classification tree (in Figure 14) to the primary/secondary nonpolling systems. As depicted in the figure, the following nonpolling systems:

1. request to send/clear to send (RTS/CTS);
2. Xon/Xoff;
3. Time division multiple access (TDMA)

The first two approaches, RTS/CTS and Xon/Xoff, are rather simple; the third approach, TDMA, is more sophisticated and is used in several satellite systems.

Request to Send/Clear to Send

Request to send/clear to send (RTS/CTS) is considered a rather low-level approach to protocols and data communications. Nonetheless, it is widely used because of its relationship and dependence upon the frequently used physical interface EIA-232-D.

The use of EIA-232-D to effect communications between DTEs is most common in a local environment, because EIA-232-D is inherently a short-distance interface, typically constraining the channel to no greater than a few hundred feet. As shown in Figure 20(a), devices can control the communications between each other by raising and lowering the RTS/CTS signal on the EIA-232-D channel (pins 4 and 5, respectively). A common implementation of this technique is found in the attachment of a terminal to a simple multiplexer. The terminal requests use of the channel by raising its RTS line (4). The multiplexer responds to the request by raising the CTS line (5). The terminal then sends its data to the multiplexer through the transmitted data line (2).

Xon/Xoff

Another widely used primary/secondary nonpolling technique is Xon/Xoff (see Figure 20(b)). Xon is an ANSI/IA5 transmission character. The Xon character is usually implemented by DC1. The Xoff character, also an ANSI/IA5 character, is represented by DC3. Peripheral devices such as printers, graphics terminals, or

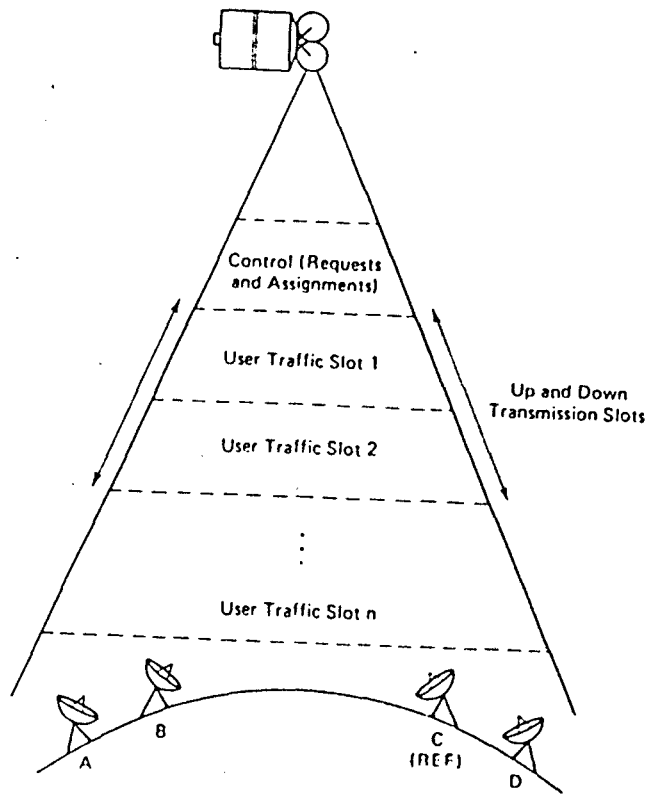


FIG 21: TIME DIVISION MULTIPLE ACCESS (TDMA)

plotters can use the Xon/Xoff approach to control traffic coming into them. The master or primary station, typically a computer, sends data to the remote peripheral site, which prints or graphs the data onto an output media. Since the plotter or printer is slow relative to the transmission speed of the channel and the transmission speed of the transmitting computer, its buffers may become full. Consequently, to prevent overflow it transmits back to the computer an Xoff signal, which means stop transmitting or “transmit off.”

Upon receiving the Xoff, the computer ceases transmission. It holds any data until it receives an Xon signal. This indicates that the peripheral device is now free (for instance, its buffers now have been cleared) and is ready to receive more data.

As you can see, the Xon/Xoff approach is quite simple; it is of a fairly low level, generally using the EIA-232-D pin connections, a V.24, or some other interface. For example, pins 2 and 3 can be used to support this protocol. The data are transmitted across pin 2 from the computer to the peripheral device and the Xon/Xoff signals are transmitted back to the computer through pin 3.

An inquiring person might question why RTS/CTS and Xon/Xoff are included in a classification of network protocols. After all, one might say, these systems are too basic to be considered a protocol. The answer to this question is simply that these systems are used extensively in DTE and DCE communications, especially with multiplexers, modems, printers, and plotters, so you should be aware of them. It is likely that your installation uses these approaches for some of your interfaces. While they may not be as complex as a Continuous ARQ, they are quite useful and inexpensive.

Time Division Multiple Access (TDMA)

A more elaborate approach to primary/secondary nonpolling system is time division multiple access (TDMA). This technique is a sophisticated form of time division multiplexing (TDM) introduced in the last section of Chapter One and discussed further in the following section. Figure 21 provides an illustration of a TDMA satellite network. Since C is designated as a master station (often called the reference station [REF]). The responsibility of the reference station is to accept requests from the secondary stations which are an indication that the secondary station wishes to use the channel. The requests are sent as part of the ongoing transmissions in a special control field. Periodically, the reference station transmits a control frame indicating which station can use the channel during a given period. Upon receiving the permission frame, the secondary stations adjust their timing to transmit within the predesignated slot.

TDMA does not use a polling selection system. Nonetheless, it does fit into our classification of primary/secondary networks, because the TDMA reference station has the option of assigning or not assigning stations to a slot. The assignments, made in response to requests, are based on the relative priority of the station or the type of traffic from the station.

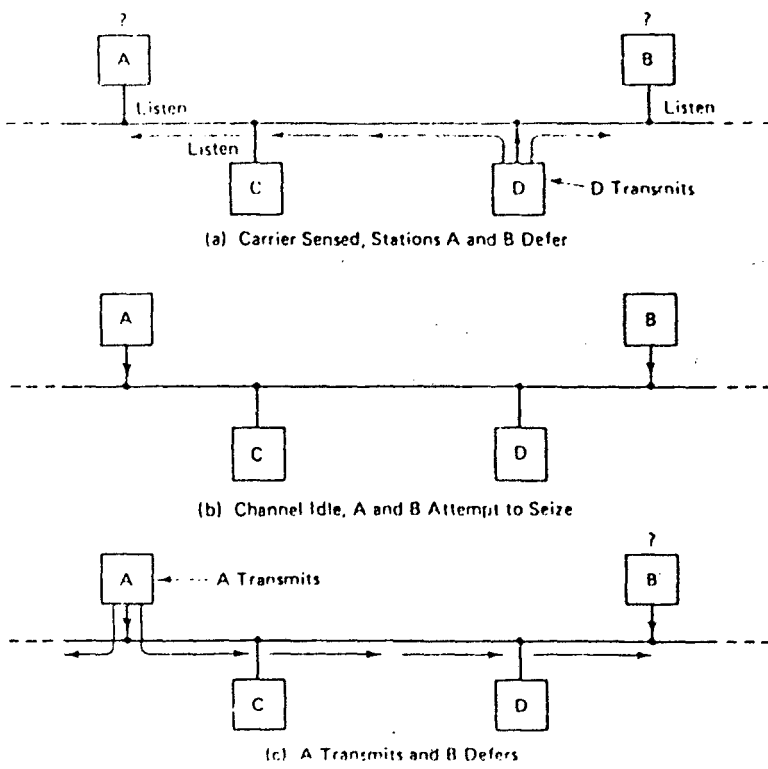


FIG 22: CARRIER SENSE (COLLISION) SYSTEMS

PEER-TO-PEER NONPRIORITY SYSTEMS

Time Division Multiplexing (TDM) or Slot

We now turn our attention to the second major classification of network protocols – the peer-to-peer technique. First, let us examine peer-to-peer nonpriority systems. Time division multiplexing (TDM) is probably the simplest example of peer-to-peer nonpriority systems. Under the TDM system, each station is given a slot of time on the communications channel and the slots are divided equally among the users. Each user has the full use of the channel during that slot of time. TDM is actually a simple form of TDMA discussed in the previous section. The TDM approach is found in both local area networks and wide area systems. Some vendors might not classify a TDM as a protocol; nonetheless, the approach is used in the networking of computer and terminals on both bus and ring topologies.

Register Insertion

A number of ring-based networks use the register-insertion technique to control traffic. Any station can transmit whenever an idle state exists on the link. If a frame is received while the station is transmitting, the frame is held in a register and transmitted behind the station's frame. This approach permits the "piggybacking" of multiple frames on the ring. Register insertion is a sophisticated form of a slotted ring.

Carrier Sense (Collision) Systems

Carrier sense (Collision) networks are another example of peer-to-peer nonpriority systems. This approach is also widely used in local networks. Several implementations use this technique with the Ethernet specification and IEEE 802.3 standard. A carrier sense network considers all stations equal, so the stations contend for the use of the channel on an equal basis. Before transmitting, the stations are required to monitor the channel to determine if the channel is active (that is, if another station is sending data on the channel). If the channel is idle, any station with data to transmit can send its frame onto the channel. If the channel is occupied, the stations must defer to the passing signal.

Figure 22 is an illustration of a carrier sense collision network. Stations A, B, C and D are attached to a bus or channel (providing a horizontal topology) by bus interface units (BIUs). Let us assume stations A and B wish to transmit; however, station D is currently using the channel, so the BIUs at stations A and B "listen" and defer to the passing frame being transmitted from station D. Upon the line going idle (figure 22 (b)), stations A and B attempt to seize the channel.

Table 5 CARRIER SENSE NETWORKS

Condition	Nonpersistent	p-persistent	1-persistent
Channel idle	transmit immediately	transmit with p; Defer with $1 - p$	transmit immediately
Channel busy	randomized wait &	transmit with p;	continually

	sense	defer with $1 - p$	sense
Collision	randomized	randomized	randomized
	retransmission	retransmission	retransmission

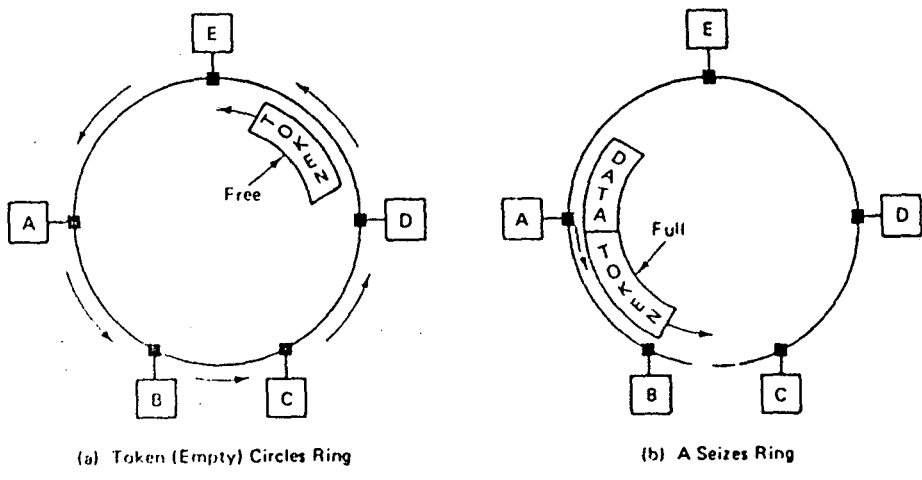
Carrier sense networks provide several methods for channel seizing (see Table 5). One technique, the nonpersistent carrier sense technique, provides the facility for all stations to transmit immediately upon sensing the idle channel, with no arbitration before the transmission. In the event the channel is busy, the stations wait a random period of time before sensing the channel again. Another technique used on slotted systems, the p-persistent carrier sense, provides a waiting algorithm at each station (P stands for probability). For example, stations A and B do not transmit immediately upon sensing a line going idle; rather, each station invokes a routine to generate a randomized wait, typically a few microseconds. If a station senses a busy channel, it waits a slot (time period) and tries again. It transmits to an idle channel with a probability p and it defers to the next slot with a probability of $1 - p$. Yet another technique, I-persistent carrier, has a station transmitting immediately upon sensing an idle channel. When a collision occurs, the stations wait a random period before sensing the channel. The method is called 1-persistent because the station transmits with a probability of 1 when a channel is sensed as idle.

The p-persistent technique is designed to meet a 1-persistent goal of reduced idle channel time and a non-persistent goal of reduced collisions. However, p must be set to a low value to achieve proper performance. Perhaps surprisingly, 1-persistent is favoured by many vendors and standards groups.

To continue the discussion, we assume that station A in Figure 22(C) seizes the channel before station B has an opportunity to finish its randomized wait. A short time later, after B's randomized threshold has expired, it listens and determines that A has transmitted and seized the channel. Consequently, it must continue to adhere to one of the three techniques for busy conditions until the channel goes idle again.

Since A's transmission requires time to propagate to station B, station B may be unaware that a signal is on the channel. In this situation, channel B may transmit its frame even though channel A has supposedly seized the channel. This problem called the collision window. The collision window is a factor of the propagation delay of the signal and the distance between the two competing stations. For instance, if A and B are one kilometer apart (.6 of a mile), it takes approximately 4.2 microseconds for station A's signal to reach station B. During this period, B has an opportunity to transmit, which results in a collision with station A.

Carrier sense networks are usually implemented in local area networks because the collision window lengthens with a longer, wide area channel. The long channel gives rise to more collisions and reduces throughput in the network. Generally, a long propagation delay (along delay before one station knows the other



■ Ring Interface Unit (RIU)

FIG 23: TOKEN RING

is transmitting) gives rise to a greater incidence of collisions. Longer frames can mitigate the effect of long delay.

In the event of a collision, the stations have a facility to detect the distorted data. Each station is capable of transmitting and listening to the channel simultaneously. As the two signals collide, they create voltage irregularities on the channel which are sensed by the colliding stations. Both stations turn off the transmission and, after a randomized wait period, attempt to seize the channel again. The randomized wait prevents the collision from recurring, since it is unlikely that the competing stations will generate the same randomized wait time.

Token Passing

Token passing is another widely used method of implementing both peer-to-peer nonpriority and priority systems. The priority systems are discussed later. The technique is found in many local area networks. Some token-passing systems are implemented with a horizontal bus topology; others are implemented with a ring topology.

Token Ring : The ring topology is illustrated in Figure 23. The stations are connected to a concentric ring through a ring interface unit (RIU). Each RIU is responsible for monitoring the data passing through it, as well as for regenerating the transmission and passing it to the next station. If the address in the header of the transmission indicates that the data are destined for a station, the interface unit copies the data and passes the information to the user DTE or DTEs attached to it.

If the ring is idle (that is, no user data is occupying the ring), a “free” token is passed around the ring from node to node. The token is used to control the use of the ring by a free or busy indication. A busy token is an indication that a station has seized the ring and is transmitting data. A free token indicates that the ring is free and any station that has data to transmit can use the token to transmit data. The control of the ring is passed sequentially from node to node around the ring. This technique is called an implicit token system because any station is allowed to transmit data when it receives a free token.

During the period that the station has seized the token, it has control of the ring. Upon seizing the token (i.e., marking the token busy), the transmitting station (station A in figure 23) inserts data behind the token and passes the data through the ring. As each RIU monitors the data, it regenerates the transmission, checks the address in the header of the data, and passes the data on to the next station. Eventually the data are received at the original transmitting station. This station is required to mark the token free, absorb the data, and pass the token on to the next station on the ring. This requirement prevents one station from monopolizing the entire ring. If the token passes around the ring without being used, then the station can once again seize the token and transmit data.

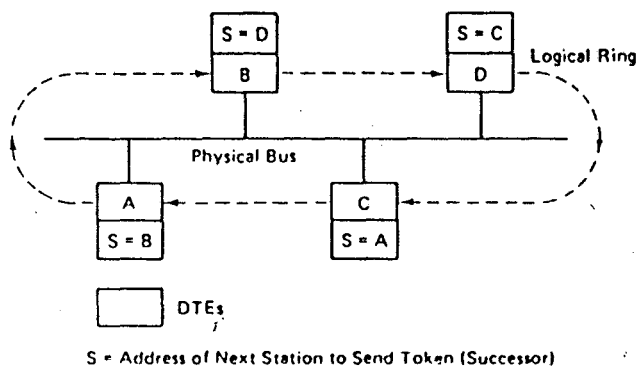


FIG 24: TOKEN BUS

Some systems provide for the token to be removed from the ring, another user frame placed behind the first data element, and the token placed behind the last data transmission. This allows a “piggybacking” effect (similar to register insertion) on the network with multiple user frames circling the ring. Piggybacking is especially useful for large circumferential rings that experience a long delay in the transmission around the ring.

Token Bus : Token-bus system provides a horizontal channel (bus), yet provide access to the channel as if it were a ring. The protocol eliminates the collisions found in the carrier sense (collision) systems and allow the use of a nonring (bus) channel. Figure 24 gives a simple illustration. Keep in mind that the token bus requires no physical ordering on the bus. The station can be logically configured to pass the token in any order.

The protocol uses a control frame called an access token or access right. This token gives a station the exclusive use of the bus. The token-holding station uses the bus for a period of time to send and received data (or even to poll other stations), then passes the token to a designated station. In the bus topology, all stations listen and receive the access token, but the only station allowed to seize the channel is the station so designated in the access token. All other stations must wait their turn to receive the token.

The stations receive the token through a cyclic sequence, which forms a logical ring on the physical bus. This form a token passing is called an explicit token system, because the bus topology causes an ordering of the nodes’ use of the channel.

PEER-TO-PEER PRIORITY SYSTEMS

The last major classification of network communications systems is the peer-to-peer priority technique. As indicated in Figure 14, the technique is illustrated with three approaches; priority slot, carrier sense (collision-free), and token passing (priority). The systems introduced here are as follows:

Priority Slot

The priority-slot system is similar to the conventional time division multiplexing approach discussed earlier. However, the use of the channel is determined on a priority basis. For instance, the following criteria can be used to establish the priority for use of the channel:

- Prior ownership of the slot
- Response time needs for a station
- Amount of data to be transmitted
- Time-of-day transmission requirements.

Priority slot can be established without a master station. The loading of priority parameters into logic at each site provides the control of the use of the slots.

Carrier Sense (Collision-Free) Systems

Carrier sense (collision-free) systems have many similarities to the carrier sense (collision) networks. The major difference is the use of logic to prevent collisions from occurring. Collision-free systems may be implemented with techniques resembling the priority slot network. Another approach is to provide an additional facility in the network called a timer or arbiter. This device determines when a station can transmit without danger of collisions. The timing is determined at each station, with no master site to supervise the use of the channel.

Each port has a predetermined timing threshold. When the timing threshold expires, the port uses a timing parameter to determine when to transmit (similar in concept to "seizing" a token). The timing can be established on a priority basis, with the highest priority port having its timer expire first. If this port chooses not to transmit, the channel remains idle. The next highest priority station senses the channel is idle. Its timer indicates it is within a time threshold to transmit, so it may then seize the channel.

The higher priority stations, if they do not transmit, create an idle condition on the channel, which allows the lower priority stations to use the channel. In conventional slot networks, the idle time translates into wasted transmission opportunities. However, the collision-free network uses the arbiter to allow the next

highest priority station on the link to seize the idle time if it has data to transmit. This approach reduces considerably the idle time on the channel.

Token-passing (Priority) Systems

The last example of peer-to-peer priority systems is an enhanced token passing scheme, in which priorities are added to a token-passing system, usually a token ring. Each system attached to a token network has a priority assigned to it. Typically, eight possible priorities are available. The object of the token-passing priority scheme is to give each station an opportunity to reserve the use of the ring for the next transmission around the ring. As the token and data circle the ring, each node examines the token, which contains a reservation field. If the individual node's priority is higher than the priority number in the reservation field, it raises the reservation field number to its level, thus reserving the token on the next round. If another node does not make the reservation field higher, then the station is allowed to use the token and channel on the next pass around the ring.

The station seizing the token is required to store the previous reservation value in a temporary storage area at its location. Upon releasing the token when it finishes a complete loop around the ring, the station restores the network to its previous lowest priority request. In this manner, once the token is made free for the next round, the station with the highest reservation is allowed to seize the token. Token-passing priority systems are widely used in local area networks (LANs).

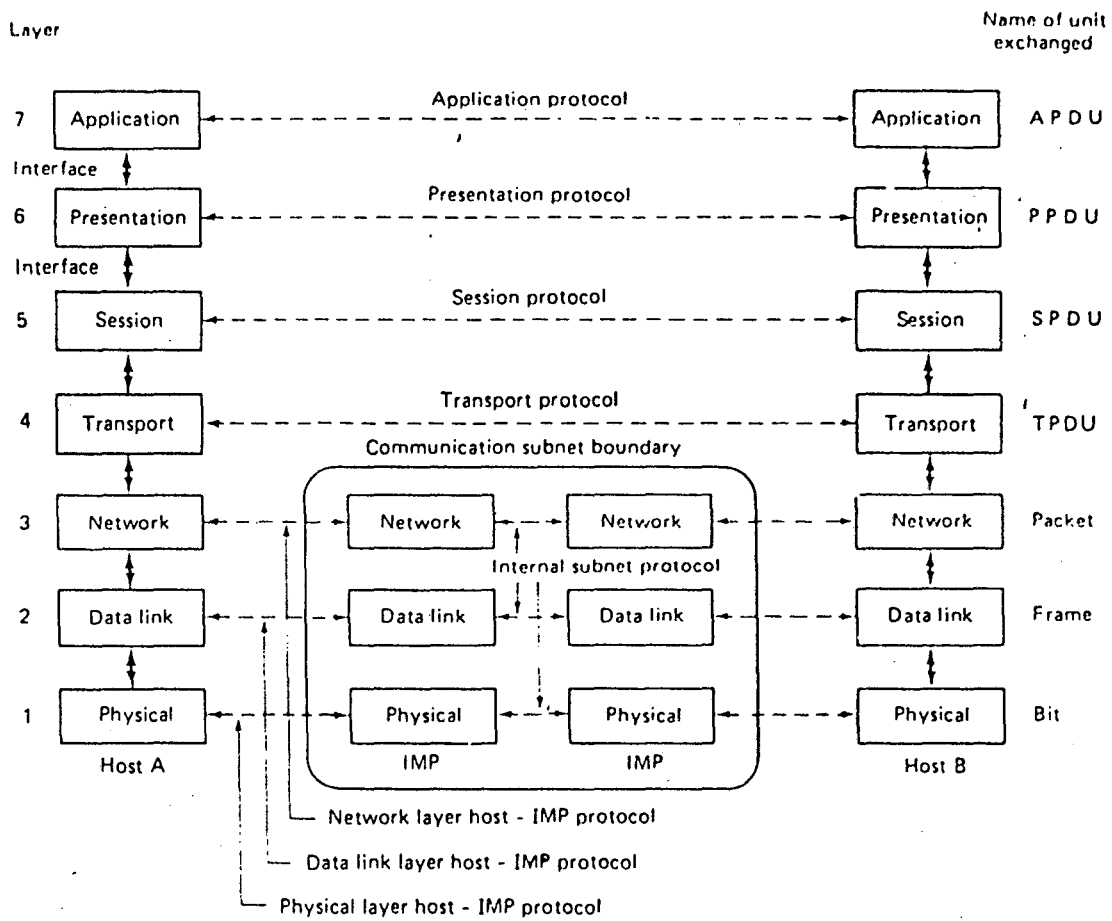


FIG 25: NETWORK ARCHITECTURE BASED

ON OSI MODEL

THE OSI REFERENCE MODEL

The model is shown in Fig 25. This model is based on the proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the various protocols (Day and Zimmermann, 1983). The model is called the **ISO OSI (Open System interconnection) Reference Model** because it deals with connecting open systems – that is, systems that are open for communication with other systems.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers are as follows: -

1. A layer should be created where a different level of abstraction is needed.
2. Each layer should perform a well defined function
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols
4. The layer boundaries should be chosen to minimize the information flow across the interfaces
5. The number of layers should not be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy

The OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do. However, ISO has also produced standards for all the layers, although these are not strictly speaking part of the model. Each one has published as a separate international standard.

The Physical Layer

The **physical layer** is considered with transmitting raw bits over a communication channel. The design issues have to do so with making sure that when one side sends 1 bit, it is received by the other side as 1 bit, not as a 0 bit. Typical questions here are how many volts should be used to represent a 1 and how many for a 0, how many microseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established and how it is torn down when both sides are finished, and how many pins the network connector has and what each pin is used for. The design issues here largely deal with mechanical, electrical, and procedural interfaces, and the physical transmission medium, which lies below the physical layer. Physical layer design can be properly considered to be the domain of the electrical engineer.

The Data Link Layer

The main task of the **data link layer** is to take a raw transmission facility and transform it into a line that appears free of transmission errors to the network layer. It accomplishes this task by having the sender break the input data up into **data frames** (typically a few hundred bytes), transmit the frame sequentially, and process the **acknowledge frames** sent back by the receiver. Since the physical merely accepts and transmits a stream of bits without any regards to meaning or structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by

attaching special bit patterns to the beginning and end of the frame. If these bit patterns can accidentally occur in the data, special care must be taken to avoid confusion.

A noise burst on the line can destroy a frame completely. In this case, the data link layer software on the source machine must retransmit the frame. However, multiple transmissions of the same frame introduce the possibility of duplicate frames. A duplicate frame could be sent, for example, if the acknowledge from the receiver back to the sender was destroyed. It is up to this layer to solve the problems caused by damage, loss, and duplicate frames. The data link layer may offer several different services classes to the network layer, each of a different quality and with a different price.

Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism must be employed to let the transmitter know how much buffer space the receiver has at the moment. Frequently, this flow regulation and the error handling are integrated, for convenience.

If the line can be used to transmit data in both directions, that introduces a new complication that the data link layer software must deal with. The problem is that the acknowledge frames for *A* and *B* traffic complete for the use of the line with data frames for *B* and *A* traffic. A clever solution (piggybacking) has been devised; we will discuss it in detail later.

Network layer

The **Network layer** is concerned with controlling the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes could be based on static tables that are “wired into” the network and rarely changed. They could also be determined at the start of each conversation, for example a terminal session. Finally, they could be highly dynamic, being determined anew for each packet, to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in each other’s way, forming bottlenecks. The control of such congestion also belongs to the network layer.

Since the operations of the subnet may well except remuneration for their efforts, there is often some accounting function built into the network layer. At the very least, the software must count how many packets or characters or bits are sent by each customer, to produce billing information. When a packet crosses a national border, with different rates on each side, the accounting can become complicated.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network

layer to overcome all these problems to allow heterogeneous networks to be interconnected.

In broadcast networks, the routing problem is simple, so that network layer is often thin or even non-existent.

The Transport layer

The basic function of the **transport layer**, is to accept data from the session layer, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently, and in a way that isolates the session layer from the inevitable changes in the hardware technology.

Under normal conditions, the transport layer creates a distinct network connection from each transport connection required by the session layer. If the transport connection requires a high throughput, however the transport layer might create multiple network connections, dividing the data among the network connections to improve throughput. On the other hand, if creating or maintaining a network connection is expensive, the transport layer might multiplex several transport connections onto the same connection to reduce the cost. In all cases, the transport layer is required to make the multiplexing transparent to the session layer.

The transport layer also determines what type of service to provide the session layer, and ultimately, the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages in the order in which they are sent. However, other possible kinds of transport service are transport of isolated messages with no guarantee about the other order of delivery, and broadcasting of messages to multiple to multiple destinations. The type of service is determined when the connection is established.

The transport layer is a true source-to-destination or **end-to-end** layer. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message header and control messages. In the lower layers, the protocols are between each machine and its immediate neighbors, and not by the ultimate source and destination machines, which may be separated by many IMP's. The difference between layers 1 through 3, which are chained, and layers 4 to 7, which are end-to-end, is illustrated in Fig 25.

Many hosts are multiprogrammed, which implies that multiple connections will be entering and leaving each post. There is to be some way to tell which message belongs to which connection. This transport header* is one place this information could be put. In addition to multiplexing several message streams onto one channel, the transport layer must take care of establishing and deleting connections across the network. This requires some kind of naming mechanism, so that a process on one machine has a way of describing with whom it wishes to converse. There must also be a mechanism to regulate the flow of information, so that a fast host cannot overrun a slow one. Flow control between host is distinct from flow control between IMP's, although we will later see that similar principles apply to both.

The Session Layer

The session layer allows users on different machines to establish the sessions between them. A session allows ordinary data transport, as does the transport layer, but it also provides some enhanced services useful in some applications. A session might be used to allow a user to log into a remote time sharing system or to transfer a file between two machines.

One of the services of the session layer is to manage dialogue control. Sessions can allow traffic to go in both directions at the same time, or in only one direction at a time. If traffic can only go one way at a time (analogous to a single rail road track), the session layer can help keep track of those turn it is.

A related session service is token management. For some protocols it is essential that both sides do not attempt the same operation at the same time. To manage these activities the session layer provides tokens that can be exchanged. Only the side holding the token may perform the critical operation.

Another session service is synchronization. Consider the problems that might occur when trying to do a two-hour file transfer between two machines on a network with a one hr mean time between crashes. After each transfer was aborted, the whole transfer would have to start over again, and would probably fail again when the network next crashed. To eliminate this problem, the session layer provides a way to insert checkpoints into the data stream, so that after a crash, only the data after the last checkpoint have to be repeated.

The Presentation Layer

The presentation layer performs certain functions that are requested sufficiently to warrant finding a general solution for them, rather than letting each user solve the problems. In particular, unlike all the lower layers, which are just interested in moving bits reliably from here to there, the presentation layer is concerned with the syntax and semantics of the information transmitted.

A typical example of a presentation survey is encoding data in a standard agreed upon way. Most user programs do not exchange random binary bit strings. They exchange things such as people's names, dates, amounts of money, and invoices. These items are represented as character strings, integers, floating points numbers, and data structures composed of several simpler items. Different computers have different codes for representing character strings (e.g., ASCII and EBCDIC), integers (e.g., one's complement and two's complement) and so on. In order to make it possible for computers with different representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire". The job of managing these abstract data structures and converting from the representation used into the computer to the network standard representation is handles by the presentation layer.

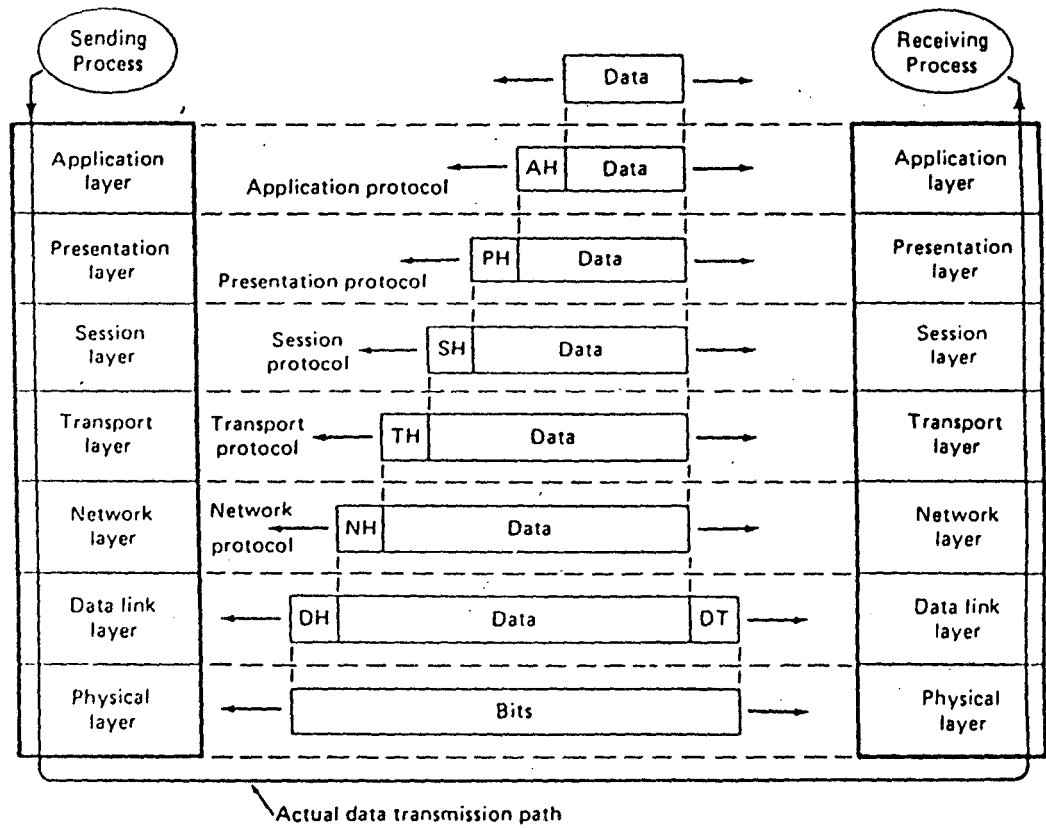


FIG 26: DATA TRANSMISSION IN OSI MODEL

The presentation layer is also concerned with other aspects of information representation. For example, data compression can be used here to reduce the number of bits that have to be transmitted and cryptography is frequently required for privacy and authentication.

The Application Layer

The application Layer contains a variety of protocols that are commonly needed. For example, there are hundreds of incompatible terminal types in the world. Consider the plight of a full screen editor that is supposed to work over a network with many different terminal types, each with different screen layouts, escape sequences for inserting and deleting text, moving the cursor, etc.

One way to solve this problem is to define an abstract network virtual terminal that editors and other programs can be written to deal with. To handle each terminal type, a piece of software must be written to map the functions of the network virtual terminal on to the real terminal. For example, when the editor moves the virtual terminal's cursor to the upper left-hand corner of the screen, this software must issue the proper command sequence to the real terminal to get its cursor there too. All the virtual terminal software is in the application layer.

Another application layer function is file transfer. Different file systems have different naming conventions, different ways of representing text lines and so on. Transferring a file between two different systems requires handling these and other incompatibilities. This work, too, belongs to the application layer, as do electronic mail, remote job entry, directory lookup, and various other general-purpose and special-purpose facilities.

Data transmission in the OSI Model

Figure 26 shows the example of how data can be transmitted using the OSI model. The sending process has some data it wants to send to the receiving process. It gives the data to the application layer, which then attaches the application header, AH (which may be null), to the front of it and the resulting item to the presentation layer.

The presentation layer may transform this item in various ways, and possibly add a header to the front, giving the result to the session layer. It is important to realize that the presentation layer is not aware of which portion of the data given to it by the application layer is AH, if any, and which is true used data. Nor shall it be aware.

This process is repeated until the data reach the physical layer, where they are actually transmitted to the receiving machine. On that machine the various headers are stripped off one by one as the message propagates up the layers until it finally arrives at the receiving process.

The key idea throughout is that although actual data transmission is vertical in Fig 26, each layer is programmed as though it were really horizontal. When the

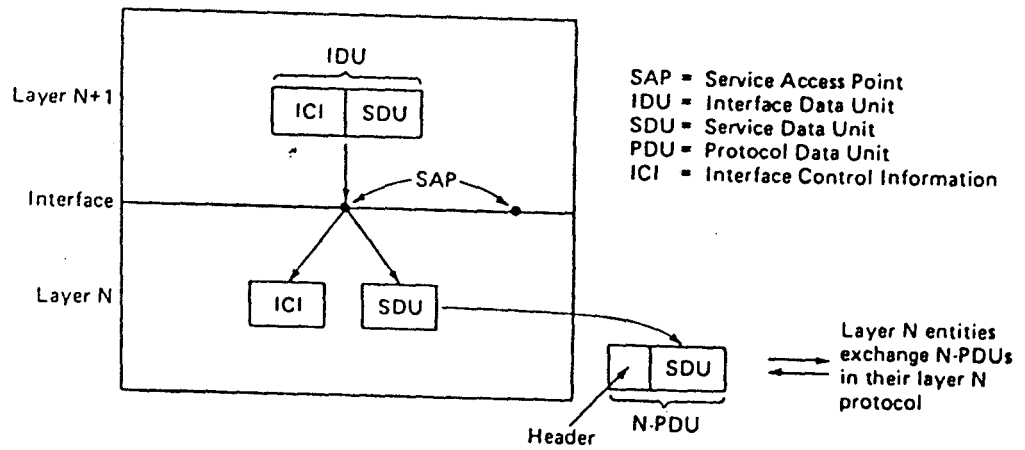


FIG 27: RELATION BETWEEN LAYERS

AT AN INTERFACE

sending transport layer, for example, gets a message from the session layer, it attaches a transport header and sends it to the receiving transport layer. From its point of view, the fact that it must actually hand the message to the network layer on its own machine is an unimportant technicality. As an analogy, when an Uighur-speaking diplomat is addressing the United Nations, he thinks of himself as addressing the other assembled diplomats. That, in fact, he is really only speaking to his translator is seen as a technical detail.

Services

The real elements in each layer are called entities. An entity can be a software entity (such as a process), or a hardware entity (such as an intelligent I/O chip). Entities in the same layer on different machines are called **peer entities**. The layer 7 entities are called **application entities**; the layer 6 entities are called the **presentation entities**, and so on.

The entities in layer N implement a service used by layer N+1. In this case layer N is called the **service provider** and layer N+1 is called the **service user**. Layer N may use the services of layer N-1 in order to provide service. It may offer several classes of service, for example, fast, expensive communication and slow, cheap communication.

Services are available at **SAP's (service access point)**, the layer N SAPs are the places where layer N+1 can access the services offered. Each SAP has an address that uniquely identifies it. To make this point clearer, the SAP's in the telephone system are the sockets into which modular phones can be plugged, and the SAP addresses are the telephone numbers of these sockets. To call someone you must know his SAP address. Similarly, in the postal system, the Sap addresses are the street addresses and the post office boxes. To send a letter, you must know the addressee's SAP address.

In order for two layers to exchange information, there has to be an agreed upon set of rules about the **interface**. At a typical interface, the layer N+1 entity passes an **IDU (Interface Data Unit)** to the layer N entity through the SAP as shown in Fig 27. The **IDU** consists of an **SFU (Service Data Unit)** and some control information. The SDU is the information passed across the network to the peer entity and then up the layer N+1. The control information is needed to help the lower layer do its job (e.g., the number of bytes in the SDU), but is not the part of the data.

In order to transfer the SDU, the layer N entity may have to fragment it into several piece, each of which is given a header and sent as a separate **PDU (Protocol Data Unit)** such as a packet. The PDU headers are used by the peer entities to carry out their peer protocol. They identify which PDU's contain data and which contain control information, provide sequence numbers and counts, and so on. The transport, session and application PDU's are often referred to as TPDUs ,SPDUs and APDUs respectively. No one talks much about the other PDUs.

Connection-oriented and Connectionless Services

Layers can offer two different types of services to the layers above them: connection-oriented and connectionless. In this section we will look at these two types, and examine the differences between them.

Connection-oriented service is modeled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk, and then hang up. Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then terminates the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects in at one end, and the receiver takes them out in the same order at the other end.

In contrast, connectionless service is modeled after the postal system. Each message (letter) carries the full destination address, and each one is routed through the system independent of all the others. Normally, when two messages are sent to the same destination the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first. With a connection-oriented service this is impossible.

Each service can be characterized by a quality of service. Some services are reliable in a sense that they never lose data. Usually a reliable service is implemented by having the receiver acknowledge the receipt of each message, so the sender is sure that it has arrived. The acknowledge process introduces overhead and delays, which are often worth it, but sometimes unreliable.

A typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly and in the same order they were sent. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it is much faster.

Reliable connection-oriented service has two minor variations: message sequences and byte streams. In the former, the message boundaries are preserved. When two 1K messages are sent, they arrive as two distinct 1K messages, never as one 2K message. In the latter, the connection is simply a stream of bytes, with no message boundaries. When 2K bytes arrive at the receiver, there is no way to tell if they were sent as one 2K message, or two 1K messages, or 2048 one-byte messages. If the pages of a book are sent over a network to a phototypesetter as separate messages, it might be important to preserve the message boundaries. On the other hand, with a terminal logging on to a remote time-sharing system, a byte stream from the terminal to the computer is all that is needed.

As mentioned above, for some applications, the delay introduced by acknowledgements are unacceptable. One such application is digitized voice traffic. It is preferable for telephone users to hear a bit of noise on a line or a garbled word from time to time to introduce a delay to wait for acknowledgements.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

FIG 28: SIX DIFFERENT TYPES OF SERVICES

Primitive	Meaning
Request	An entity wants the service to do some work
Indication	An entity is to be informed about an event
Response	An entity wants to respond to an event
Confirm	An entity is to be informed about its request

FIG 29: FOUR CLASSES OF SERVICE PRIMITIVES

Not all applications require connections. For example, as electronic mail becomes more common, can electric junk mail be far behind? The electronic junk mail sender probably does not want to go into the trouble of setting up and later tearing down a connection just to send one item. Nor is 100% reliable delivery essential, specially if it costs more. All that is needed is a way to send a single message that has a high probability of arrival, but no guarantee. Unreliable (meaning not acknowledged) connectionless service is often called **datagram service**, in analogy with telegram service, which also does not provide an acknowledgement back to the sender.

In other situations, the convenience of not having to establish a connection to send one short message is desired, but reliability is essential. The **acknowledged datagram service** can be provided for these applications. It is like sending a registered letter and requesting a return receipt. When the receipt comes back, the sender is absolutely sure that the letter was delivered to the intended party.

Still another service is the **request-reply service**. In this service the sender transmits a single datagram containing a request; the reply contains the answer. For example, a query to the local library asking Uighur is spoken falls into this category. Fig 28. summarizes the types of services discussed above.

Service Primitives

A service is formally specified by a set of **primitives** (operations) available to a user or other entity to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. In the OSI model, the service primitives can be divided into four classes as shown in Fig 29.

The first class of primitive is the *request* primitive. It is used to get work done, for example, to establish a connection or to send data. When the work has been performed, the peer entity is signaled by an *indication* primitive. For example, after a *CONNECT request* (in OSI notation), the entity being addressed gets a *CONNECT indication* announcing that someone wants to set up a connection to it. The entity getting the *CONNECT indication* then uses the *CONNECT response* primitive to tell whether it wants to accept or reject the proposed connection. Either way, the entity issuing *CONNECT request* finds out what happened via a *CONNECT confirm* primitive.

Primitives can have parameters, and most of them do. The parameters to a *CONNECT request* might specify the machine to connect to, the type of service desired. The parameters to a *CONNECT indication* might contain the caller's identity, the type of service desired, and the proposed maximum message size, it could make a counterproposal in its response primitive, which would be made available to the original caller in the *confirm*. The details of this **negotiation** are part of the protocol. For example, the case of two conflicting proposals about maximum message size, the protocol might specify that the smaller value is always chosen.

As an aside on terminology, the OSI model carefully avoids the terms “open a connection” and “close a connection” because to electrical engineers, an “open circuit” is one with a gap or break in it. Electricity can only flow over “closed circuits”. Computer scientists would never agree to having information flow over a closed circuit. To keep both camps pacified, the official terms are “establish a connection” and “release a connection”.

Services can either **confirmed** or **unconfirmed**. In a confirmed service, there is a *request*, an *indication*, a *response*, and a *confirm*. In an unconfirmed service, there is just a *request* and an *indication*. *CONNECT* is always a confirmed service because the remote peer must agree to establish a connection. Data transfer, on the other hand, can be either confirmed or unconfirmed, depending on whether or not the sender needs an acknowledgement. Both kinds of services are used in networks.

To make the concept of a service more concrete, let us consider as an example a simple connection-oriented service with eight service primitives as follows: -

1. *CONNECT request* - Request a connection to be established
2. *CONNECT indication* - Signal the called party.
3. *CONNECT response* - Used by the caller to accept/reject calls
4. *CONNECTION confirm* - Tell the caller whether the call was accepted
5. *DATA request* - Request that data be sent
6. *DATA indication* - Signal the arrival of data
7. *DATA request* - Request that a connection be released
8. *DISCONNECT indication* - Signal peer about the request

In this example, *CONNECT* is a confirmed service (an explicit response is required), whereas *DISCONNECT* is unconfirmed (no response).

The relationship between Services and Protocols

Services and protocols are distinct concepts, although they are frequently confused. This distinction is so important, however, that we emphasize it again here. A *service* is a set of primitive (operation) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A protocol, in contrast, is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols in order to implement their service definitions. They are free to change their protocol at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled.

An analogy with programming languages is worth making. A service is like an abstract datatype. It defines operations that can be performed on an object, but does not specify how these operations are implemented. A protocol relates to the implementation of the service, and as such is not visible to the user of the service.

Many of the pre OSI protocols did not distinguish the service from the protocol. In effect, a typical layer might have had a service primitive SENT PACKET with the user providing a pointer to a fully assembled packet. This arrangement meant that all changes to a protocol were immediately visible to the users. It is now universally accepted that such a design is a blunder of major proportions.

Network standardization

In the early days of networking, each computer manufacturer had its own network protocols. IBM had more than a dozen. The result was that users who had computers from several vendors could not connect them together into a single network. This chaos led many users to demand standardization.

Not only do standards allow different computers to communicate, but they also increase the market for products adhering to the standards, which leads to mass production, economy of scale in manufacturing, VLSI implementations, and other benefits that decrease price and further increase acceptance. In this section we will take a quick look at the important, but little known, world of international standardization.

Standards fall into two categories: de facto and de jure. De facto (latin for “from the fact”) standards are those that have just happened without any formal plan. The IBM PC and its successors are de facto standards for small office computers because dozens of manufacturers have chosen to copy IBM’s machines very closely. Unix is the de facto standard for operating systems in University computer science departments.

De hure (latin for “by law”) standards, in contrast, are formal, legal standards adopted by some authorized standardization body. International standardization authorities are generally divided into two classes: those established by treaty among national governments and voluntary, non-treaty organizations. In the area of computer networks standards, there are two principle organizations one of each type. Both standards organizations are important and will be discussed below.

Who’s Who in the Standards World

International standards are produced by ISO (International Standards Organization) a voluntary, non-treaty organization founded in 1946. Its members are the national standards organizations of the 89 member countries. These members include ANSI (U.S), BSI (Great Britain), AFNOR (France), DIN (West Germany), and 85 others.

ISO issues standards on a vast number of subjects, ranging from nuts and bolts (literally) to telephone coatings. ISO has almost 200 Technical Committees, numbered in the order of their creation, each dealing with a specific subject. TCI deals with the nuts and bolts (standardizing screws thread pitches). TC97 deals with computers and information processing. Each TC has subcommittees (SC’s) divided into working groups (WGs).

The real work is done largely in the WGs by over 100,000 volunteers world-wide. Many of these “volunteers” are assigned to work on ISO matters by their employers, whose products are being standardized. Others are government officials

ken on having their country's way of doing things become the international standard. Academic experts also are active in many of the WGs.

On issues of the telecommunication standards, ISO and CCITT sometimes cooperate (ISO is a D class member of CCITT) to avoid the irony of two official and mutually incompatible international standards.

The U.S. representative in ISO is **ANSI (American National Standard Institute)**, which despite its name, is a private, nongovernmental, nonprofit organization. Its members are manufacturers, common carriers, and other interested parties. ANSI standards are frequently adopted by ISO as international standards.

The procedure used by the ISO for adopting standards is designed to achieve as broad a consensus as possible. The process begins when one of the national standards organizations feels the need for an international standard in that area. A working group is then formed to come up with a **DP (Draft Proposal)**. The DP is then circulated to all the member bodies, which get six months to criticize it. If a substantial majority approves, a revised document, called a **DIS (Draft International Standard)** is produced and circulated for comments and voting. Based on the results of this round, the final text of the **IS (International Standard)** is prepared, approved, and published. In areas of great controversy, a DP or DIS may have to go through several versions before acquiring enough votes, and the whole process can take years.

NBS (National Bureau of Standards), is an agency of the U.S. department of Commerce. It issues standards that are mandatory for purchases made by the U.S. Government, except for those of the Department of Defence, which has its own.

Another major player in the standards world is **IEEE (Institute of Electrical and Electronics Engineers)**, the largest professional organization in the world. In addition to publishing scores of journals and running numerous conferences each year, IEEE has a standardization group that develops standards in the areas of electrical engineering and computing. IEEE's 802 standard for local area networks is the key standard for LAN's. It has subsequently been taken over by ISO as the basis for ISO 8802.

SATELLITE NETWORKS

Introduction

Man-made satellites have revolutionized communication and, in many instances, the shape of the world politics. For example, the graphic, live images of the Vietnam War, conveyed by satellite to the American public, had a dramatic effect on the public's opinion of the war. The startling satellite broadcasts of Ethiopia's famine shocked an otherwise uninformed world. On the lighter side, the live broadcasts of sports events, such as the British Open golf classic and the French Open tennis tournament, have added significantly to the fans' enjoyment of these sports.

In 1945, Arthur C. Clark, with uncanny foresight, described in the magazine *Wireless World* the satellite technology as it exists today. Clark predicted satellite communications would create a communications revolution as profound as that brought about by the telephone. While satellite technology may not have had such a profound effect, it certainly has altered considerably the way in which we communicate and the way in which we perceive the world.

In this chapter we discuss in general terms reasons satellite technology is widely used, its disadvantages, and how satellite communications facilitates provide for computer and terminal networks.

Satellite Components

Satellite communications use microwave frequency antennas to receive radio signals from transmitting stations on the earth and to relay the signals back down to earth stations. Figure 29(a) illustrates this process. The satellite serves as an electronic relay station. Earth station A transmits signals of a specific frequency (up link) to the satellite. In turn, the satellite receives the signals and retransmits them back down to earth station B on the down-link frequency. The down link can be received by any station that falls within the radiated signal. The signals may be voice images, data transmissions, or television video signals.

The satellite receiving/transmitting capability is supported by a device called a transponder. The satellite transponders operate at very high frequencies, typically in a gigahertz range. The majority of the satellites today use frequencies in the range of 6/4 gigahertz range. As shown in Figure 29(a), the signal is transmitted from the earth station at a different frequency than this same signal being transmitted from the satellite stations. These signals are noted as f_1 from the up link to the satellite station and f_2 on the down link to the earth stations. This approach prevents the two transmissions (the up and down signals) from interfering with each other because they are operating in different frequency ranges.

Pros and Cons of Satellite Networks

Communication satellites provide several attractive features. First, each satellite has a large transmission capacity. Since the satellites are operating in the

broad bandwidth range of the gigahertz level, a satellite can support several thousand voicegrade channels.

Communications satellites have the capability of providing a broad range of coverage. Some satellites can cover the entire United States with one transponder. This feature is quite attractive for organizations that have widely distributed components, for example, branch offices or subsidiaries located around the country or in dispersed regions of the world. However, the wide coverage presents some potential security problems, since a station can pick up another organization's transmission if it is calibrated to the proper channel. Consequently, many satellite carriers implement security measures for the customer, such as encryption devices.

The cost of transmitting the signal is independent of the distance between the two earth sites. It is immaterial if the two sites are five miles or a thousand miles apart. If they are serviced by the same transponder the transmission cost is constant, since the signals transmitted from the transponder can be received by all stations, regardless of their distance from each other.

Communication satellites provide the opportunity to design a switched network without physical switches. In order to establish switches in the LAN based system, an organization must lease carrier lines and interface these lines into the organization's communication facilities (such as computers, front-end processors, multiplexers, etc). In contrast, since the earth stations communicating with the satellite transponder are sending and receiving on the same two channels, they need only listen to the down link frequency to determine if the transmission is destined for them. If it is not, they simply ignore the signal. If it is their data, they copy the signal and present it to the end user. This broadcast capability can translate into significantly reduced costs when compared to the land-based network that uses numerous physical communications lines and switches.

Satellite communications are not without problems, however. As stated earlier, security can be compromised if the signal is not scrambled or encrypted. Poor weather conditions can interfere with the signal as it traverses up and down the communications channel. It is not usual for heavy rainstorms to interfere with the signal. Also, since the signal is transmitted over a very great distance (usually 22,300 miles each way to and from the satellite station), a delay occurs in the reception of the signal at the earth station. In some instance, the delay can present problems with line protocols and response time (this subject is discussed shortly).

Periodically, the sun, the earth station, and the satellite are directly aligned with each other. The sun's rays travel directly into the earth station's antenna, creating a sun transient: excessive thermal noise in relation to the received signal. Conversely, a solar eclipse occurs during the spring and fall when the earth is between the sun and the satellite for a few minutes during a 23 day period. During this time, the solar cells on the satellite may deplete, which creates loss of power to the satellite electronic components.

The communications signal from the satellite may also interfere with other radio signals from land-based systems. Consequently, a careful allocation of frequency spectrum is necessary to prevent such interference.

Finally, a finite amount of frequencies exist for the 6/4 and 14/12 GHz satellites, and a finite number of satellites can be placed in orbit. While spectrum and orbit space has not been a hindrance to the technology in the past, it is becoming a problem and will require increased cooperation of the many nations using communications satellite technology.

Brief History

Before discussion how satellite communications are used in computer and terminal networks, it should prove useful to discuss some of the major landmarks in satellite communications. The interest in satellites happened rather suddenly in 1957, when the Russians launched the famous Sputnik into space. The incident spurred the U.S and Canada (and later Europe) to increase their efforts in the satellite communications field as well as in rocket-launching technology. An interesting aside to these early incidents is the somewhat misconceived notion that U.S technology was inadvertently lagging far behind that of Russia. While the U.S was lagging, it was because the U.S had made the first breakthrough in nuclear warfare and did not think it necessary to spend the necessary research money to develop the large launch-capacity vehicles necessary for warfare missiles (and later, the satellite communications field). Nonetheless, the United States soon followed the Russians with the launch of Explorer 1 in January 1958.

Neither Sputnik nor Explorer had communication facilities. The United States Army created the first communications satellite, launched in December 1958. The famous Early Bird, the world's first commercial satellite, was launched from Cape Kennedy in 1965.

These earlier satellites had elliptical orbits. They were launched usually no higher than 6,000 miles above the earth. The low orbit resulted in the satellite moving around the earth's horizon faster than the earth's rotation. This presented tracking problems between the earth stations and the satellite because of the satellite's frequent disappearance over the horizon. It is estimated that the North Atlantic region alone would have required 50 elliptically orbiting satellites for continuous coverage.

Today's satellites are in a geosynchronous (or geostationary) orbit. These satellites are launched 22,300 miles above the earth and are positioned on a plane perpendicular to the equator. They are designed to achieve a rotating speed around the earth of 6,879 miles per hour, and the gravitational pull of the earth, counterbalanced by the velocity of the satellite, gives the satellite the appearance of being stationary relative to the earth's rotation. Consequently, the earth station's antenna can remain in a relatively fixed position (which is called an orbital slot), since the satellite's motion is fixed relative to the earth's position. The geosynchronous satellites are often launched in groups of three. These satellites, positioned 120 degrees apart, achieve nearly world wide coverage.

Using Satellites to Communicate

Conventional Multiplexing

Communications between the satellite and the earth stations can be controlled in a number of ways. One approach, frequency division multiplexing (FDM), is used on some systems. The entire channel spectrum is divided into subchannels, and users are assigned the various subchannels to transmit any traffic they wish within their prescribed spectrum space. Frequency division multiplexing has two significant drawbacks, however. First, most of the available bandwidth has to be utilized as a guardband to prevent adjacent channels from interfering with each other. Second, if the users are not all transmitting regularly, then much of the subchannel bandwidth is wasted because of the idle channel conditions.

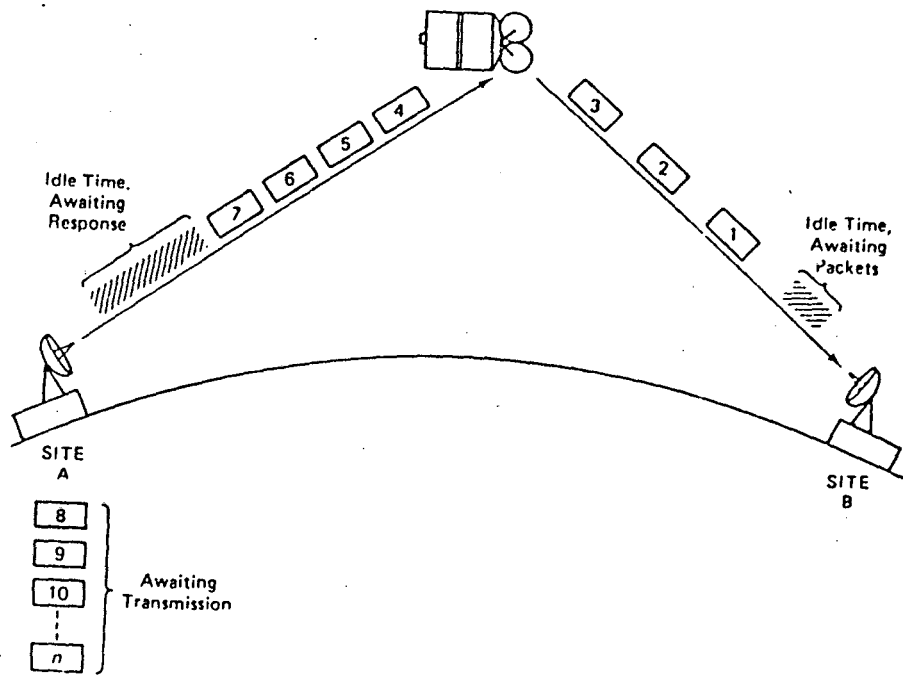
Another approach is the use of time division multiplexing (TDM) in which the time spectrum is divided and users share time slots on the communications channel. The major shortcoming of time division multiplexing is similar to that of FDM. Since the capacity of the channel is preallocated to each potential user, the channel is wasted if the user is not transmitting regularly. (We address this problem shortly when we look at a form of multiplexing called time division multiple access (TDMA).

Polling/Selection

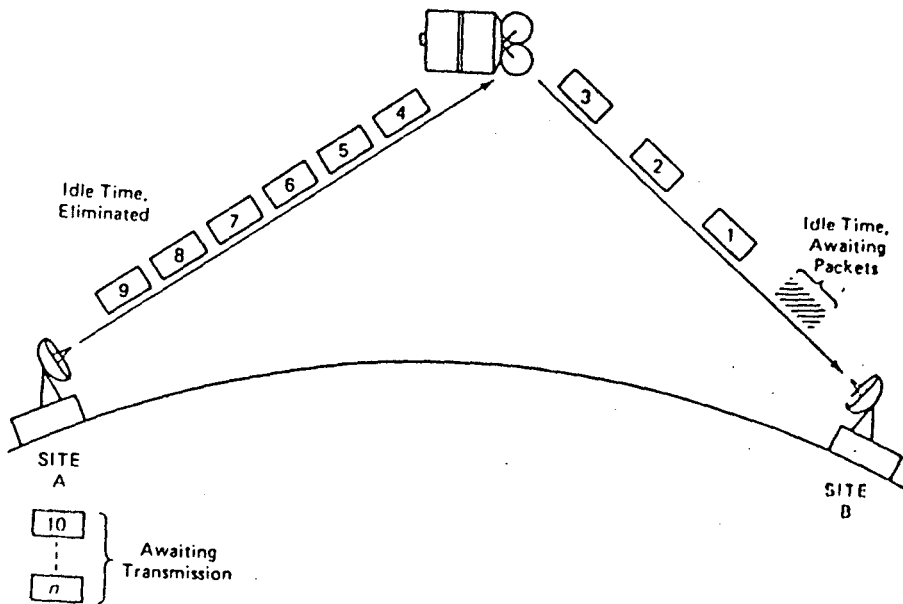
Satellite communications can also be controlled by a conventional primary/secondary relationship using polling/selection techniques (see Figure 16). The primary traffic is managed by an earth station (designated as a primary site) sending polls and selects up to the satellite to be relayed back down to secondary earth stations. An alternate approach (not used much) is to have the satellite station provide the polls and selects to control the network. Let us examine both approaches to determine the advantages and disadvantages of polling/selection in satellite systems.

First, we assume a satellite computer performs the polling and selection. Since the satellite is located 22,300 miles above the earth and the signal propagates at a rate of 186,000 miles per second, it takes a minimum of 120 milliseconds (ms) for the poll or select to reach an earth station ($23,000 \text{ miles} \div 186,000 \text{ mps} = .120 \text{ sec}$). It requires another 120 ms for the response to the poll and select to reach the communications satellite. Consequently, each polling and selection cycle takes 240 ms. Assuming n users are to be polled and selected within the network, a full polling and selection cycle would take $.240 \times 100$, or 24 seconds, for a full polling/selection cycle to take place. Obviously, the delay presents some rather serious response-time problems. If a ground station controls the polls and selects, the performance is even worse, since the poll or select is sent up to the satellite and down to the earth would require 48 seconds for the full polling/selection cycle.

The delay is also evident for a session in which only two stations are using the channel. If user A from one station sends a frame on the satellite channel to user B at another site, user A must pause and wait for an acknowledgement (assuming the use of a stop-and-wait, half-duplex protocol). If the two users are sending multiple frames



(a) Limited Window Size



(b) Expanded Window Size

FIG 30: EFFECT OF PROPAGATION DELAY

to each other (as in a file-transfer batch transmission), the accumulated delays create an extended time to complete the process, which reduces the effective utilization of the channel. As shown in Table 6, it experiences considerable degradation in the utilization of a channel (especially a satellite channel). This table illustrates some of the problems encountered with older protocols and reinforces the idea that full-duplex continuous ARQ protocols are better methods for use on satellite channels.

Table 6 :CHANNEL UTILIZATION

Block (Frame) Size	10 ms delay	38 ms delay	500 ms delay
40 bytes	76.9%	46.7%	6.2%
132 bytes	91.7%	74.3%	18.0%
516 bytes	97.7%	91.9%	46.2%

The larger the block or frame size, the better the channel utilization because the larger blocks mask the delay effect of the long-distance circuit. Half-duplex delay does not create a problem or short-distance channels with short delays (of 10 ms, for example). It is more evident on circuits of several hundred miles (40 ms delay) to several thousand miles (500 ms delay, or more).

The use of a full-duplex Continuous ARQ protocol instead of a stop-and-wait polling system decreases the response time and increases the throughput. As the table shows, satellite delay using polling/selection is especially evident using the half-duplex, stop- and-wait approach. The Continuous ARQ allows the overlapping of transmissions and acknowledgements across a full-duplex channel and reduces the amount of delay incurred in the polling cycle. For example, one station can be polled, and while the poll is being transmitted to that station, yet another station can transmit data on the return channel.

Yet continuous ARQ also has problems. For Example, if the system is transmitting 1,000 bit frames and the channel is operating at 50,000 bits per second, the channel provides the speed for multiple frames to be send on the up and down links in succession before any responses or data are transmitted back. High-speed channels actually increase the effect of propagation delay, because it takes less time to send each frame up and down the channel. Consequently, the window closes faster with high-speed channels and short blocks of data. This problem is illustrated in Figure 30.

In order to prevent the channel from becoming idle (as in Figure 30(a)), the conventional ARQ window of seven is often expanded. The window expansion prevents the transmitting side from closing its window while awaiting acknowledgements. Some systems use the HDLC (High level Data Link Control) extended-sequencing option and expand the acknowledgement from the receiver. The expanded window allows the system to compensate for the propagation delay (see figure 30(b)) and provides for more efficient channel utilization.

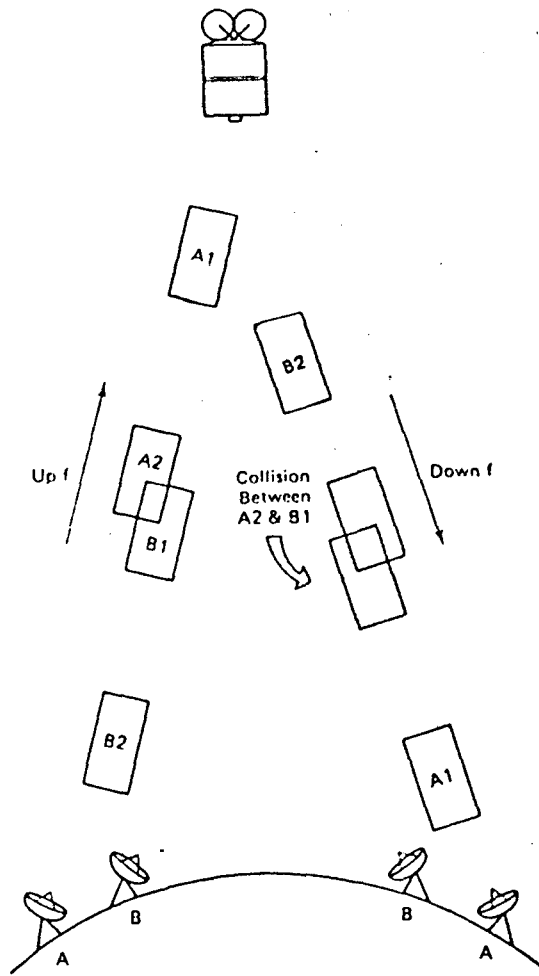


FIG 31: RANDOM ALOHA ON SATELLITE NETWORKS.

Nonpolling Peer/Peer systems

ALOHA. In the early 1970s, Norman Abramson, at the University of Hawaii, devised a technique for uncoordinated users to effectively compete for a channel. The approach is called the ALOHA system; it is so named because the Hawaiian word is used without regard to whether a person is arriving or departing. The original ALOHA technique used a ground-based radio packet system rather than satellites, but the ideas are applicable to any channel media when users are contending for its use.

As depicted in Figure 31, ALOHA is considered to be a peer-to-peer system. Several variations of ALOHA exist. One approach fits the carrier sense collision-detect protocol (Random ALOHA). Another variation can be used as a priority slot system (Slotted ALOHA). We will use ALOHA to introduce peer-to-peer systems, even though most satellite protocols have implemented more efficient techniques (these will be discussed later in this chapter).

The premise of ALOHA is that users are acting on a peer-to-peer basis – they all have equal access to the channel. A user station transmits whenever it has data to send. Since the channel is not allocated by any primary/secondary structure, it is possible (and probable) that users will occasionally transmit at approximately the same time. Simultaneous transmission results in the signals interfering and distorting each other as the separate signals propagate up to the satellite transponder. (the term to describe several stations transmitting on one frequency to one station is *narrowcasting*. The transmission of one station [the satellite] to many stations is called broadcasting). These “packet collisions” necessitate the retransmission of the damaged packets. (The term “packet” is used in place of “frame” under the ALOHA scheme). Since the users of the satellite link know exactly what was transmitted on to the up-link channel and when it was transmitted, they need listen only to the down-link channel at the prescribed time to determine if the broadcast packet arrived without damage. If the packet is damaged due to a collision, the station are required to retransmit the damaged packet. In essence, the idea is listen to the down-link channel one up-and-down delay time required to wait a short random period and then retransmit. The randomized wait period diminishes the chances of the competing stations colliding again, since the waiting times will likely differ and result in retransmissions at different times.

Fig 31 depicts a typical ALOHA system using satellite communications. Stations A and B are transmitting packets at will on a shared channel. The down-link channel shows that packet 1 from station A is transmitted up and down safely; packet 2 from station B is also transmitted without error. However, the second packet from A and the first packet from B are transmitted at approximately the same time. As the transmission of the two stations are narrowcasted up into the satellite station, the signals interfere with each other, resulting in collision.

The satellite station is not responsible for error detection or error correction; it transmits what it receives from the up link. On the down link, station A and B note the packets have collided and, upon waiting a random period of time (usually a few milliseconds), attempt to retransmit. This approach is quite effective when the users

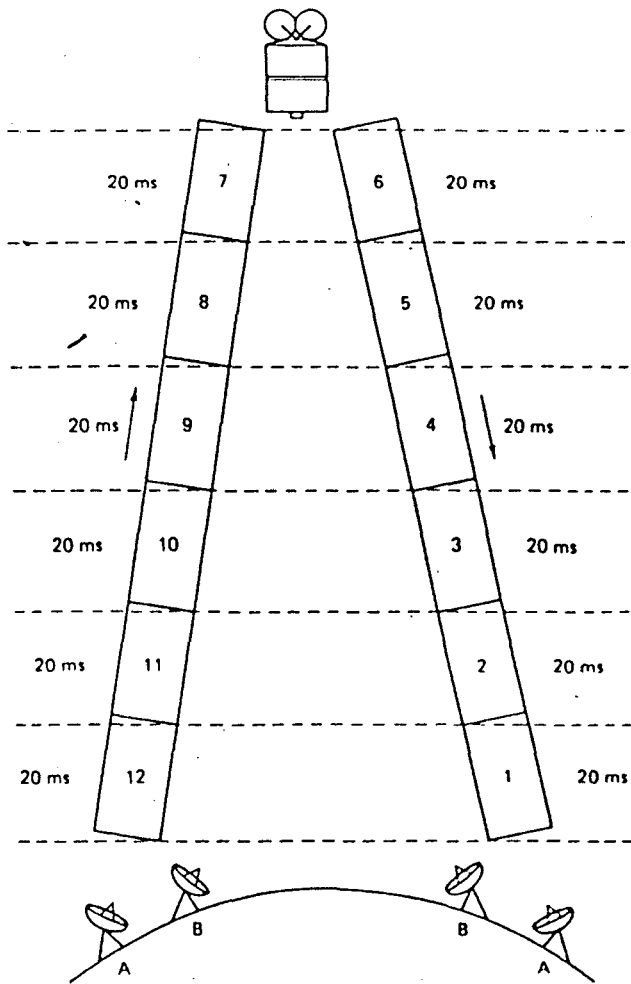


FIG 32: SLOTTED ALOHA

are uncoordinated and are sending traffic in bursts, such a data from keyboard terminals.

Random ALOHA experiences considerable degradation of throughput when the channel is heavily utilized. However, keep in mind that what is transmitted across the channel is all end-user data. Unlike the primary/secondary polling systems, ALOHA uses no polls, selects, nor negative responses to polls. Only end-user information is transmitted. Nevertheless, the pure random scheme can be improved by adapting a more efficient strategy, called Slotted ALOHA, for using the uncoordinated channel.

Slotted ALOHA requires that common clocks be established at the earth stations and the satellite. The clocks are synchronized to send traffic at specific periods. For example, the clocks may require that packets are transmitted only on 20 ms (.020 sec) increments. In this example, the 20 ms increment is derived from a 50,000-bit/s channel and 1,000-bit packets ($1000 / 50000 = .020$ second)

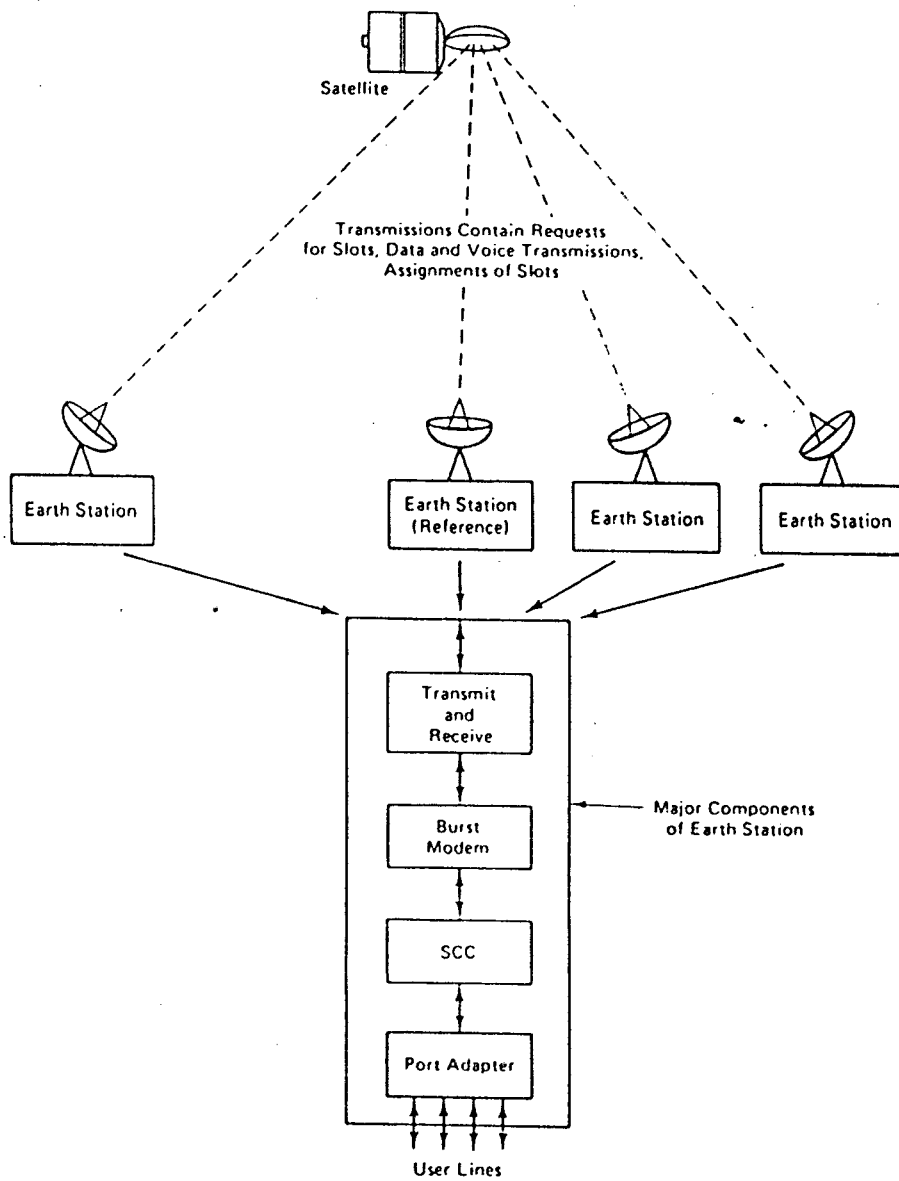
The 20 ms increment is referred to as the packet duration, which is the time in which the packet is transmitted on the channel. All stations are required to transmit at the beginning of the slot period. A packet cannot be transmitted if it overlaps more than one slot.

The Slotted ALOHA approach increases throughput substantially on the channel, because if packets overlap or collide, they do so completely; at most, only one slot is damaged. However, like pure random ALOHA, the Slotted ALOHA does offer opportunities for collisions. For example, if two stations transmit in the same clock period, their packets collide. As on the Random ALOHA approach, the stations are required to wait a random period of time before attempting to seize a slot for retransmission.

Another refinement to Slotted ALOHA is Slotted ALOHA with nonowner. The channel slots are combined into an ALOHA frame. (Fig 32.). The ALOHA frame must equal to exceed the up-and-down propagation delay. Consequently, a 1,000-bit packet lasting 20 ms would require a minimum of 12 slots to make up the ALOHA frame: $12 \text{ slots} \times 20 \text{ ms} = 240 \text{ ms}$. The 240 ms period represents the minimum up and down propagation delay ($120 \text{ ms}(\text{up}) \times 120 \text{ ms}(\text{down}) = 240 \text{ ms}$).

Slotted ALOHA with nonowner requires that a station select an empty slot in the frame. Once the user has seized the slot, it is reserved for the user successive frames until the user relinquishes the slot. The relinquishment occurs by the station sending a protocol control code, such as EOT (end of transmission). Upon receiving an EOT, the next frame transmitted is empty for that particular slot. A user station then allowed to contend for the slot with the next subsequent frame. The only collisions occurring on Slotted ALOHA with nonowner are when stations pick the same slot in the 240 ms frame.

Another variation of Slotted ALOHA is Slotted ALOHA with owner. The slots of each frame are now owned by users. The user has exclusive use of its slot within the frame as long as it has data to transmit. In the event that the user relinquishes the slot, it so indicates with an established code. The slot becomes empty



SCC: Satellite Communications Control

nature of

FIG 33: TDMA

and is available for any other use to seize it. Once another user has seized the slot, it has exclusive rights to the use of the slot until the original owner seized the slot. The rightful owner can claim the slot at any time by beginning transmission within its designated slot in the frame. The relinquishment is required when the rightful owner transmits. Obviously, the first time the owner transmits in its slot a collision may occur. On the subsequent frame, the rightful owner retransmits. The relinquishing station then must look for another free slot or go its own slots if it has then. This refined approach of ALOHA is classified as a peer-to-peer priority structure, since some stations can be given ownership over other stations. Thus, it fits into the classification tree as a priority slot system.

Nonpolling Primary/Secondary Systems

TDMA. In 1981, Satellite Business Systems (SBS) (now defunct) began offering communications services to private and public organizations via geosynchronous satellites and earth stations. Notwithstanding, the system is used in this chapter because it provides an excellent means of discussing another widely used technique for satellite communications. The example in this chapter shows how time division multiple access (TDMA) is used to achieve primary/secondary nonpolling system. The TDMA technique is used by the other satellites carriers, as well. For example, the European Telecommunication Satellite Organization (EUTELSAT) uses TDMA on its Telecom 1 satellite. This explanation covers the specific SBS protocol.

TDMA assigns slots as needed. However, unlike the ALOHA system, the slots are assigned by a primary station called the reference (REF). As depicted in Fig 33, the reference station accepts requests from the other stations, and, based on the nature of the traffic and the channel available, the REF assigns these requests to specific frames for subsequent transmission. Every 20 frames, the reference station sends the assignments to the secondary stations. One reference station is assigned to each other of the system. TDMA provides for as many as ten active transponders per satellite.

Fig 33 also shows the earth station components. The major components consist of the port adapter, the satellite communications controller (SCC), a burst modem, the transmit/receive device, and an antenna.

The port adapter is responsible for interfacing the user lines into earth station. The adapter accepts voice images at a rate of 32kbit/sec and data at rates varying from 2.4 kbit/sec to 1.544 Mbit/sec.

All digital images are passed to the satellite communications controller, which is a software oriented unit that consolidates the functions of timing, station assignment, switching, and processing of voice and data cells. It calculates channel requirements based on the number of voice connections, the number of data ports available, and the number of queued data connection requests. It then assigns these requests to TDMA frames.

The burst modem send out a 48 Mbit/s signal with 15 ms frames (0.015 sec) under the direction of the satellite controller. Thus, each transponder has the capability of operating at 48 magabits per second.

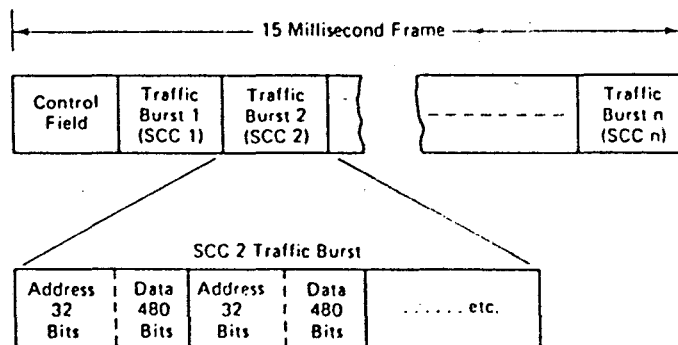


FIG 34: THE TDMA FRAME

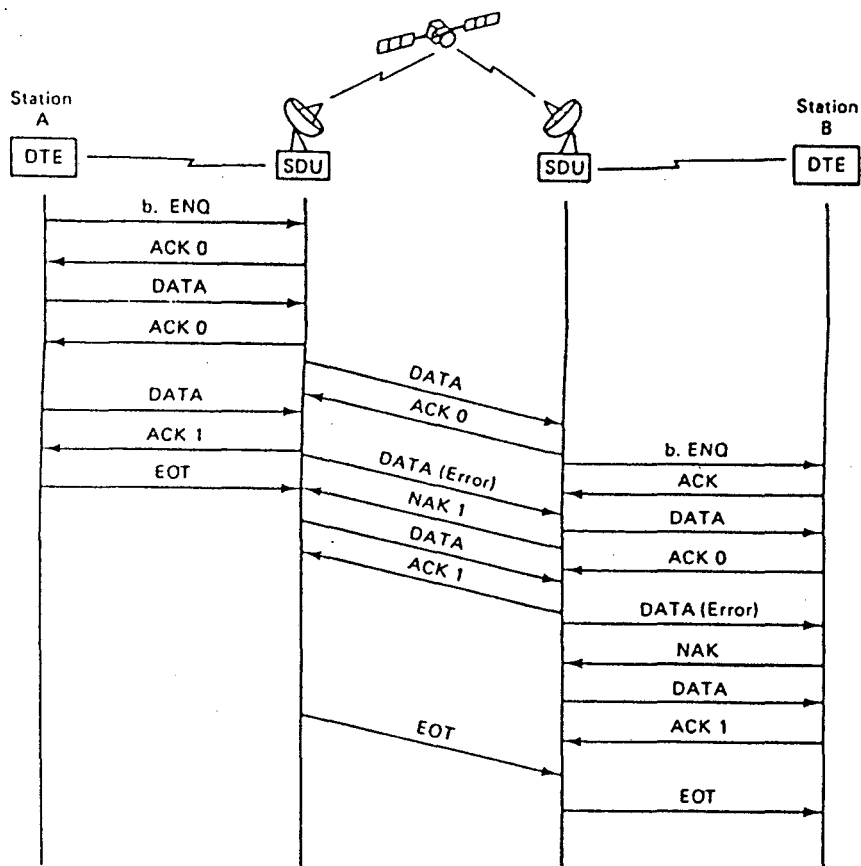


FIG 35: SATELLITE DELAY COMPENSATION
UNITS (SDU_s) FOR HALF-DUPLEX PROTOCOLS

The transmit/receive antennas are responsible for transmitting and receiving the up and down channel link. This protocol operates at 14 gigahertz on the up link and 12 gigahertz on the down link. This transmission band was chosen because it is relatively free from other satellite transmissions and it allows the earth stations to operate relatively free from the terrestrial microwave operations 4/6 gigahertz.

On a 15 ms frame, illustrated in Fig 34, the reference station (REF) transmits an assignment set for all SCC's using the transponder. As mentioned earlier, this transmission is sent every 20 frames. The assignment set specifies the capacity and position of each SCC's traffic burst to the transponder. Recall that assignments are made in response to the requests in earlier frames. The control field of the frame contains the assignments and the requests from the competing stations. The remainder of the frame consists of the traffic, which contains the traffic bursts from each SCC that was assigned a position by the reference station.

The traffic is packed in 512-bit channel consisting of a 32-bit destination address and 480 bits of data. The 480-bit data frame was chosen to accommodate the requirement for the voice transmission rate of 32 kilobits per second [$480 \times (1 \text{ second} / .015 \text{ slot}) = 32000$].

The 32 kilobits per second rate uses only a small fraction of the total 48 megabit channel capacity. Consequently, many voice and data transmissions can be time division multiplexed (TDM) efficiently onto the high-speed 48 kbit/s channel.

Satellite delay Units (SDUs)

In the earlier days of the satellite communications (early 1960's), half-duplex protocols were widely employed on user machines. Although half-duplex protocols have fallen into disuse, we discuss an approach that handles this protocol on satellite links because half-duplex systems still exist today. Satellite vendors have developed methods to compensate for the inherent inefficiency of a half-duplex systems on the satellite circuit. The satellite delay compensation unit (SDU) illustrated in Fig 35 is one such tool. Stations A and B are to communicate through a satellite channel. However, instead of communicating directly with each other, the two stations transmit and receive through a SDU. The SDU is connected to each of the stations through a land-based terrestrial link, such as microwave or optical fibers. Consequently, the delay of signal transmissions between the DTE and the SDU is very short.

The SDU is actually a protocol convertor. It accepts bisync traffic from station A and station B and buffers the traffic locally. Consequently, when station A sends a select command with an ACK0. The data is transmitted, checked for errors at the SDU, and then acknowledged. The SDU for A then transmits the data using its own protocol through the satellite circuit for transmission down to the down-link SDU (B). the SDU for B provides an error check and responds with an acknowledgement. The SDU servicing station B then goes through the same sequence of events that DTE A and SDU A performed – it sends a select to station B, which acknowledges the select, receives the data, checks for errors, and responds with an ACK0.

Fig 35 shows that the second block of data must be retransmitted between the two SDU's when the data are distorted during satellite transmission process.

Likewise, SDU B and DTE B must pass the data error-free. The SDU servicing station B must retransmit the second data frame, because sent a NAK to SDU B. Finally, the bisync EOT is sent from station A to tell station B it has no more data to transmit. The EOT is transported across the communication channel, and the remote SDU provides the bisync EOT to terminate the transmission process.

The satellite delay-compensation units provide some immunity from the cumulative effects of delay on half-duplex protocols. However, certain protocols, even though they may be half-duplex, may not benefit from the SDU compensation. For example, if half-duplex messages, such as bisync, are sent one at a time for an interactive session, there is no cumulative effect on the delay of these transmissions, even though the long delay could be a problem for extremely high speed applications. However, half-duplex systems which utilize batch transmission can benefit substantially from the use of SDU, because the session between the DTE's usually encompasses *many* blocks of transmission. In batch systems, the transmitting SDU can receive and buffer an entire file before it activates the remote SDU session. Likewise, the receiving SDU can buffer the batch file completely and then establish the session with the receiving DTE.

The Teleport

The concept of teleport has received considerable attention in the industry. The teleport is a satellite or several satellites shared by multiple users. Typically, the users are tenants in an office building with an industrial complex. The users of the teleport are inbuilt to the satellite through cable, optical fibers or microwave links. The basic idea is to share the high capacity satellite channel in order to reduce user's overall communication costs. The teleport transmits all types of images (voice, data, facsimile and video) with a wide diversity of data rates. The digital transmission speeds range from 44 kbit/s to 1.544 Mbit/s. Of course, users have the option of lesser data rates through multiplexing techniques.

The primary focus of the teleport is to support private business. However, other users are targeted as well. Some teleport companies are marketing to residential users in the primary form of closed circuit television broadcast. Other vendors are supporting hotel and educational organizations.

A teleport provides several offers to the users. A teleport can be located within an industrial complex or at user's site. The users can be located far away from the teleport and communicate with the teleport through microwave, optical fiber links, coaxial cable or a telephone channel. The satellite communication then takes the transmission and transports it to the other users throughout the country.

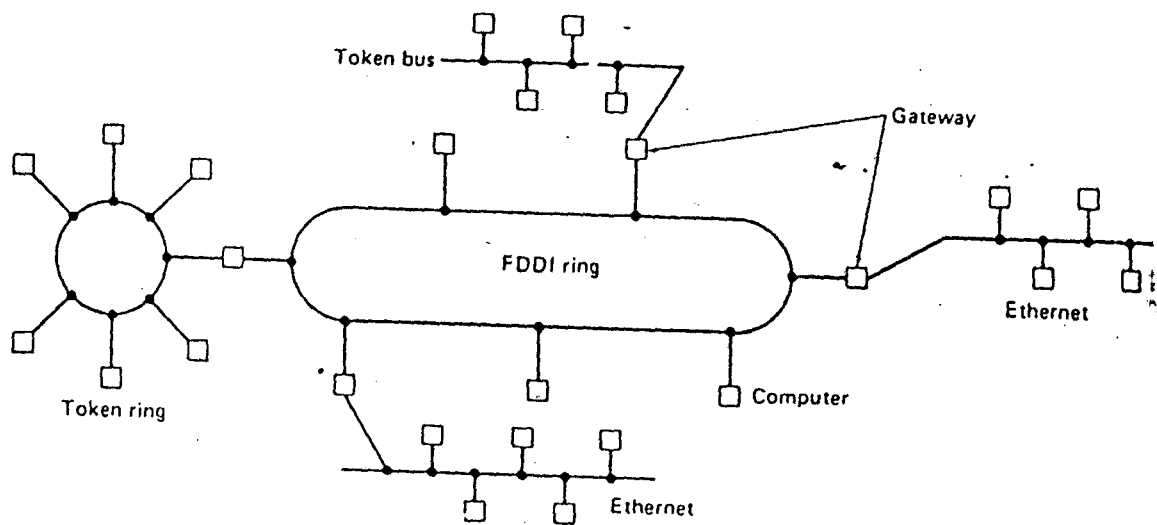


FIG 36: AN FDDI RING BEING USED AS
A BACKBONE TO CONNECT LANS AND COMPUTERS.

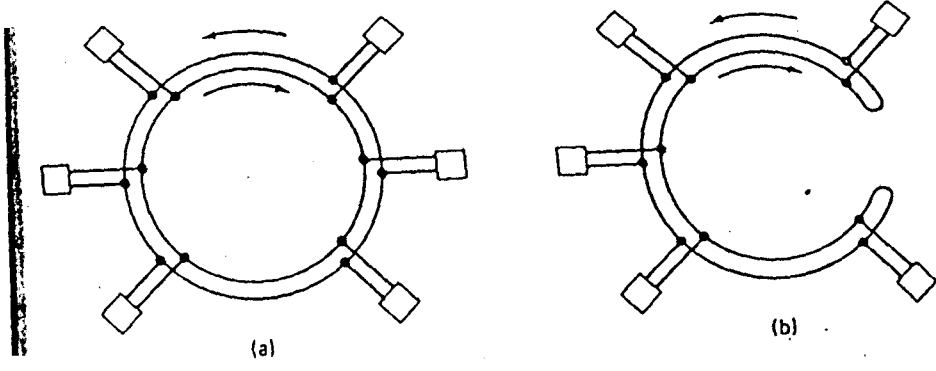


FIG 37 : AN FOOD RING

FIBER OPTIC NETWORKS

Fiber Optic is becoming increasingly important, not only for the wide-area point-to-point links, but also for metropolitan and local area networks. Fiber has high bandwidth, is thin and light weight, is not affected by electromagnetic interference from heavy machinery (important when cabling runs through elevator shafts), power surges, or lightning, and has excellent security because it is nearly impossible to wiretap without detection. In the following sections we will look at some networks that use fiber optics, wither exclusively or in combination with copper.

FDDI

FDDI (Fiber Distributed Data Transfer) is a high performance fiber optic token ring LAN running at 100 Mbps over distances up to 200 km with up to 1000 stations connected. It can be used in the same way as any of the 802 LANs, but with its high bandwidth, another common use is as a backbone to connect copper LANs, as shown in Fig 36. FDDI-II is the successor to FDDI, modified to handle synchronous circuit switched PCM data for voice or ISDN traffic, in addition to ordinary data. We will refer to both of them as just FDDI. This section deals with both the physical layer and the MAC sublayer of FDDI.

FDDI uses multimode fibers because the additional expenses of single mode fibers is not needed for networks running at only 100 Mbps. It also uses LEDs rather than lasers, not only due to their lower cost, but also because FDDI may sometimes be used to connect directly to user workstations. There is a danger that curious users may occasionally unplug the fiber connector and look directly into it to watch the bits go by at 100 Mbps. With a laser the curious user might end up with a hole in his retina. LEDs are too weak to do any eye damage but are strong enough to transfer data accurately at 100 Mbps. The FDDI design specifications calls for no more than 1 error in 2.5×10^{10} bits. Many implementations will do much better.

The FDDI cabling consists of two fiber rings, one transmitting clockwise and the other transmitting counterclockwise, as illustrated in Fig 37(a). If either one breaks, the other can be used as a backup. If both break at the same point, for example, due to a fire or other accident in the cable duct, the two rings can be joined into a single ring approximately twice as long, as shown in Fig 37(b). Each station contains relays that can be used to join the two rings of bypass the stations in the event if station problems. Wire centers can also be used, as in 802.5 IEEE standard.

FDDI defines two classes of stations, A and B. Class A stations connect to both rings. The cheaper class B stations only connect to one of the rings. Depending on how important fault tolerance is, an installation can choose class A or B stations, or some of each.

The physical layer does not use Manchester encoding because 100 Mbps Manchester encoding requires 200 megabaud, which was deemed too expensive. Instead a scheme called **4 out of 5** encoding is used. Each group of 4 MAC symbols (0s, 1s, and certain nondata symbols such as start-of-frame) are encoded as a group of 5 bits on the medium. Sixteen of the 32 combinations are from data, 3 are for delimiters, 2 are for control, 3 are for hardware signaling, and 8 are unused.

The advantage of this scheme is that it saves bandwidth, but the disadvantage is the loss of the self-clocking property of Manchester encoding. To compensate this loss, a long preamble is used to synchronize the receiver to the sender's clock. Furthermore, all clocks are required to be stable to at least 0.005 percent. With this stability, frames up to 4500 bytes can be sent without danger of the receiver's clock drifting too far out of the sync with the data stream.

The basic FDDI protocols are closely modeled on the 802.5 protocols. To transmit data, a station must first capture the token. Then it transmits a frame and removes it when it comes again. One difference between FDDI and 802.5 is that, in 802.5, a station may not generate a new token until its frame has gone all the way and come back. In FDDI, with potentially 1000 stations and 200 km of fiber, the amount of time wasted waiting for the frame to circumnavigate the ring could be substantial. For this reason, it was decided to allow a station to put a new token back onto the ring as soon as it is done transmitting its frames. In a large ring, several frames might be on the ring at the same time.

FDDI permits data frames similar to 802.5, including the acknowledgement bits in the *frames status* byte. However, it also permits special synchronous frames for circuits switched PCM or ISDN data. The synchronous frames are generated every 125 sec by a master station to provide the 8000 samples/sec needed by PCM systems. Each of these frames has a header, 16 bytes of noncircuit switched data, and up to 96 bytes of circuit-switched data. (i.e., up to 96 PCM channels per frame)

The number 96 was chosen because it allows four T1 channels (4x24) at 1.544 Mbps or three CCITT channels (3x32) at 2.048 Mbps to fit in a frame, thus making it suitable for use anywhere in the world. One synchronous frame every 125 sec consumes 6.144 Mbps of bandwidth for the 96 circuit-switched channels. A maximum of 16 synchronous frames every 125 sec allows up to 1536 PCM channels and eats up 98.3 Mbps.

Once a station has acquired one or more time slots in a synchronous frame, those slots are reserved for it till they are explicitly released. The total bandwidth not used by the synchronous frames to indicate which slots are available for demand assignment. The nonsynchronous traffic is divided into priority classes, with the higher priorities getting first shot at the leftover bandwidth.

The MAC protocol requires each station to have a token rotation timer to keep track of how long it was since the last token was seen. A priority algorithm similar to 802.4 is used to determine which priority classes may transmit on a given token pass. If the token is ahead of schedule, all priorities may transmit, but if it is behind schedule, only the highest ones may end.

Fibernet II

A group of researchers have built a fiber LAN known as Fibernet II which is compatible with ethernet at the transceiver interface so that stations could be plugged into it using their existing stations to transceiver cable. The hard part about

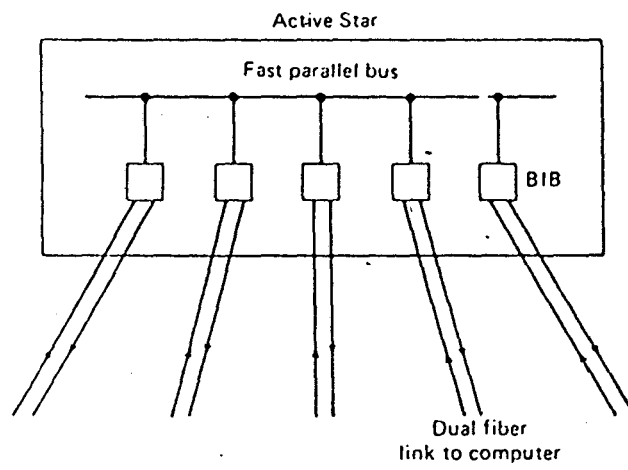


FIG 40: S/NET

building any CSMA/CD network out of fiber optics is getting the collision detection to work. Several methods are possible using the passive star configuration of Fig 38.

1. **Power sensing.** If a station senses more power than it is putting out, its transmission must be colliding with another station's.
2. **Pulse width.** If two stations collide, the incoming pulse will be wider than the outgoing pulse. This difference can be detected.
3. **Time delay.** When two stations collide, the one transmitting last will receive power from the first one earlier than its own signal should be coming back. It can detect this difference.
4. **Directional coupling.** It is possible to design the transmit and receive equipment so that a receiver does not pick up transmissions from its own station. Any transmissions it does receive while transmitting must therefore be collisions.

All of these are quite tricky to implement. Furthermore, passive stars greatly weaken the signal because the incoming energy has to be divided over all the outgoing lines. Consequently, Schmidt et al used an active star instead of a passive one. In their system, each transceiver has two point-to-point fibers running to the central star, one of them for input and the other for output. The active star is sketched in Fig 39.

At the star, each incoming optical signal is converted to an electrical signal, processed electronically, and then reconverted to an optical signal for output on the other fiber. Connecting to each incoming fiber is an opto-electrical coupler that converts the pulse stream coming off the fiber into an electrical signal. This signal is fed into a tiny CSMA/CD driver inside the active star. All the signals from the drivers go onto a common electrical bus.

If two signals collide, this event will be detected electrically the same way as in an ordinary CSMA/CD. If there is no collision, the one incoming signal is transferred to a second internal bus that drives the transmitter module. The incoming signal is broadcast to all the transceivers by modulating each other's individual LED. This scheme does not require dividing the incoming power among N ways to spread it among the N transceivers. Each transceiver gets full power.

S/Net

S/Net is another fiber optic network with an active star for switching. It was designed and implemented at Bell Laboratories. Unlike Fibernet II, whose goal was compatibility with Ethernet, the goal of S/Net was very fast switching. The structure of the active star is illustrated in Fig 40.

Each computer in the network has two 20-Mbps fibers running to the switch, one for input and one for output. The fibers terminate in a **BIB (Bus Interface Board)**. The CPUs each have an I/O device register that acts like a one-word window into BIB memory. When a word is written to that device register, the interface board in the CPU transmits the bits serially over the fiber to the BIB memory, where they are reassembled as a word in BIB memory. When the whole frame to be transmitted

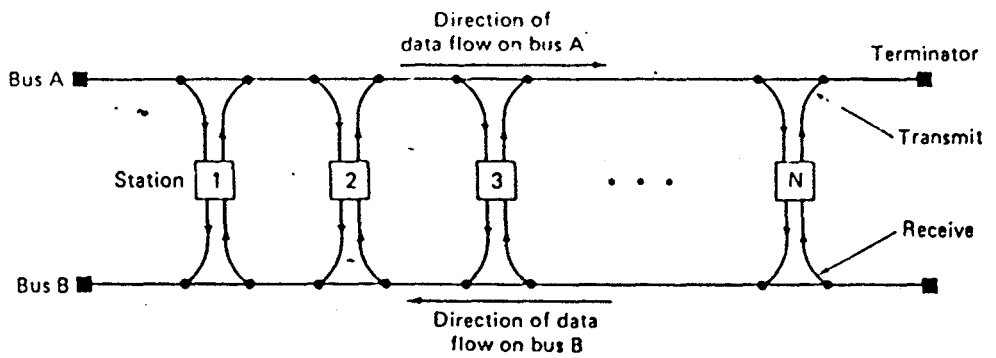


FIG 41: FASNET

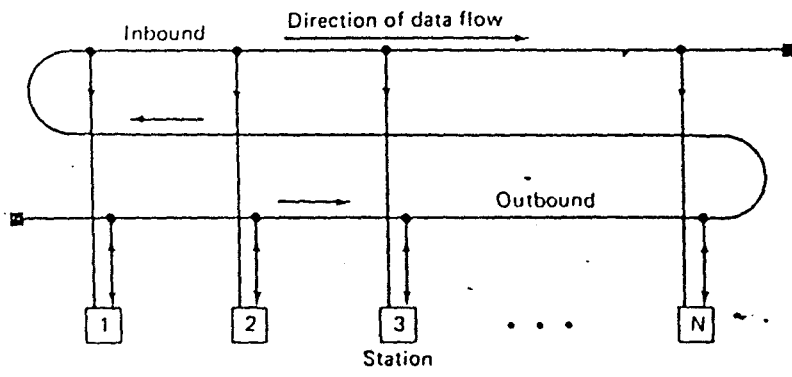


FIG 42: EXPRESS NET

has been copied to BIB memory, the CPU writes a command to another I/O device register to cause the switch to copy the frame to the memory of the destination BIB and interrupt the destination CPU.

Access to the bus is done by a priority algorithm. Each BIB has a unique priority. When a BIB wants access to the bus it asserts a signal on the bus line corresponding to its priority. The requests are recorded and granted in priority order, with one word transferred (16 bit in parallel) at a time. When all requests have been granted, another round of bidding is started and BIBs can again request the bus. No bus cycles are lost to connection, so switching speed is 16 bits every 200 nsec, or 80 Mbps.

Fasnet and Expressnet

FASNET is a high performance network suitable for use as a LAN or MAN. It was designed by Bell Laboratories.

FASNET uses two linear unidirectional buses, as shown in fig 41. Each station taps on to both buses and can send and receive on either one. When a station wants to send a frame to a higher numbered station it transmits on bus A; when it wants to send to a lower numbered station, it transmits on bus B. stations 1 and N play a special roles in the network, as we shall see.

A transmission is started when a station 1 begins transmitting a sequence of fixed-size slots on bus A and station N begins transmitting an identical sequence in the opposite direction on bus B. These slots provide the clocking for the bus. Other stations synchronize their transmissions to them as they propagate by. The slots can be thought of as a train of empty flatcars onto which data can be loaded.

When a station wanting to transmit to a higher numbered station detects the start of the train of bus A, it waits until the first empty slot passes by. The station then sets a bit in the first byte of the slot marking it as busy, and places the source and destination addresses and the data in the empty slot. If the data does not fit in a single slot, several consecutive slots may be allocated. When the downstream station to whom the frame is addressed sees the frame, it just copies it to its memory, leaving the slot on the bus, still marked as busy. And analogous mechanism is used on the bus.

When the last station on either bus detects the end of the train, it sends an announcement frame on the other bus. When it arrives, a new train is started. The interval between train departures in wither bus is equal the round trip propagation times plus the time required to transmit all the frames on that cycle.

EXPRESSNET is similar to FASNET in a number of ways, than using two buses, EXPRESSNET uses a single bus folded as illustrated in Fig 42. Each station attaches to the bus in two places, once on the outbound portion, for transmission, and once on the inbound portion, for reception.

Unlike FASNET, which is synchronous (the first station on each bus generates the complete train and the other station have to align their clocks with it),

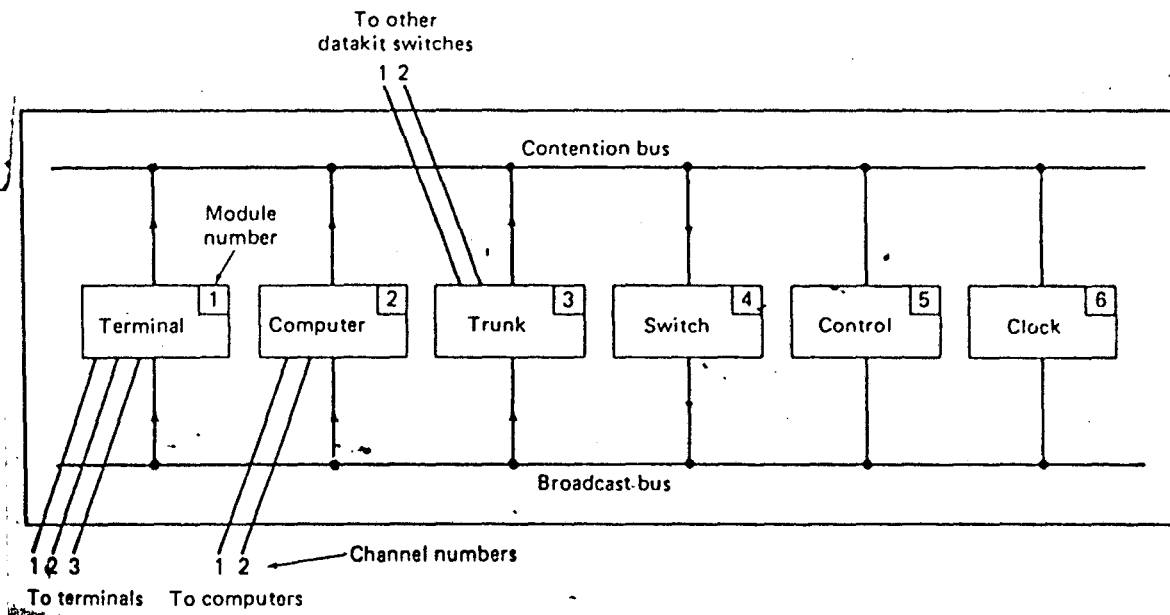


FIG 43: DAKIT

EXPRESSNET is asynchronous. When a station has a frame to transmit, it senses the inbound channel to see if the cable is already in use. If it is, the station just waits until the cable is quiet. Then the station hooks its frame on the end, to form a train. When the train reaches the inbound portion it is accepted by the station to whom it is addressed.

Due to the nonzero propagation time, a problem can occur if two stations, say 2 and 3, try to transmit simultaneously. A few microseconds after station 3 has started sending, the front of station 2's frame will arrive on the outbound channel and collide with station 3's frame. To deal with this problem, all stations monitor the outbound channel and terminate transmission instantly if they detect a frame from a lower-numbered station coming in from the left. Implicitly, this algorithm resolves collisions in favor of the lower-numbered station, but it also garbles a few bits at the front of its frame.

The obvious solution is for each station to transmit a few bytes of preamble before its frames. The preamble is not part of the data. It is designed to absorb collisions, if we stick to the train model, the preamble is a cowcatcher.

EXPRESSNET differs from FASNET in a few ways. For one thing, there is the issue of synchronous versus asynchronous transmission (analogous to the difference between a slotted ring and a token ring). For another, FASNET requires N taps on each of two cables, whereas EXPRESSNET requires $2N$ taps on one cable, potentially a significant difference with the fiber optics since the taps are lossy. Finally, the propagation time for a frame from station 2 to station N is three times as long as EXPRESSNET as FASNET.

DATAKIT

DATAKIT is a network that was designed at the Bell Labs and is currently sold by AT&T. It differs from most other networks in that it is a single integrated network to be used in LAN, MAN and WAN. Furthermore, it allows copper and fiber to be intermixed in arbitrary ways.

Architecturally, DATAKIT consists of switches, each with various kinds of lines coming out of it. These lines may go to terminals, computers, or DATAKIT switches. The lines going to terminals are typically twisted pairs, but the lines connecting two DATAKIT switches may be fiber optic trunks running at T1 speed (1.544 Mbps) or higher. Thus a DATAKIT network consists of multiple interconnected stars, rather than a bus or a ring.

The structure of a DATAKIT switch is given in Fig 43. The switch back plane has two buses, a connection bus and a broadcast bus. Various cards can be inserted into the switch, each card connected to both buses. Some cards contain RS-232-C interfaces for terminals, others contain computer interfaces, and still others contain interfaces for copper or fiber optic trunks to other switches. When a terminal, computer or other DATAKIT switch has a byte to transmit to a device attached to a different card, it completes for the connection bus, and when it acquires the bus, puts the bytes on the bus. The switch card removes the byte from the contention bus and puts it in the broadcast bus, where the destination card takes it off.

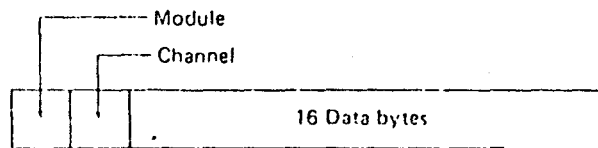


Fig 44

FIG 44 : A DATAKIT FRAME

Unlike all the networks we have studied so far, DATAKIT uses virtual circuits. When a terminal or computer wants to communicate with another terminal or computer, it sends a request to the control computer in its local switch to find a path to the destination. The control computer records the path in its tables, and uses that path when the data arrive later.

Consider for example, what happens when terminal 3 connected to module 1 in Fig 43 wants to communicate with the computer connected to the same switch via line 1 on module 2. First a virtual circuit must be established by the control computer. Establishment of the virtual circuit results in a table entry being made in the switch. For example (1,3) (2,1) might map module 1, channel (i.e., line) 3 onto module 2, channel 1.

Later, when a character arrives from the terminal, the terminal card builds a DATAKIT frame (Fig. 44) and tries to acquire the contention bus. The contention bus protocol is the binary countdown algorithm described earlier. When the bus has been acquired, it outputs the frame with its own module number (1) in the module field and the terminal's channel number (3) in the channel field.

When the switch module sees the frame, it looks up the combination (1,3) in the switching table and performs the mapping onto (2,1). The switch then copies the frame from the contention bus to the broadcast bus, replacing the source module and channel by the destination module and channel. All the cards receive the frame from the broadcast bus, but only card 2 accepts it, sending it off on its line 1.

This switching scheme not only works locally, but also over metropolitan or wide area networks. A frame destined for different DATAKIT switch will be mapped onto one of the channels of the appropriate outgoing trunk. When a data frame arrives, it will be switched to the trunk card, where it is buffered until it can acquire the contention bus in that system and be switched again.

In a way, a DATAKIT switch resembles a PBX. It has incoming and outgoing lines, and uses circuit switching. On the other hand, the DATAKIT switched data streams are just byte streams, in any format (not just PCM), and at any speed (not just 64 kpbs). Furthermore, DATAKIT is not synchronous like a time division switch, so an idle module consumes no bandwidth. In that respect, internally DATAKIT is more like a packet switch than a circuit switch, even though externally it appears to do circuit switching.

LOCAL/METROPOLITAN/WIDE AREA NETWORKS

In the past twenty years, the communications industry has focused on systems which transport data over long distances. The wide area network (WAN) industry has now matured and is a relatively stable field. The local area network (LAN) constitutes a relatively new arena for data communication. LAN technology began to gain attention in the mid-70's, and today it is one of the fastest growing industries in data communications.

WAN's, in contrast, use point-to-point links, except for satellite networks. Multiaccess channels and LAN's are so closely related, we will use this chapter for discussing LANs in general, as well as WANs with the future of networking.

To start off, let us say what do we mean by local area network. LANs generally have three characteristic features:

1. A diameter of not more than a few kilometers
2. A total data rate of at least several Mbps.
3. Complete ownership by a single organization

WANs in contrast, typically span entire countries, have data rates below 1 Mbps, and are owned by multiple organizations (the carrier owns the communications subnet and numerous clients own the host).

In between the LAN and the WAN is the MAN (Metropolitan Area Network). A MAN is a network that covers an entire city, but uses LAN technology. Cable television (CATV) networks are examples of analog MANs for television distribution. The MANs we are interested in are digital and are intended to connect computers together, not television sets, although some of them may use broadband coaxial cable as the transmission medium. Most of the discussions of LAN protocol in this chapter also holds for Mans.

Why is anyone interested in building LAN? Basically the reason are the same as for building networks in general. In some cases, the purpose of the LAN is to connect existing machines together, for example, departmental computers on a campus, to allow all of them to communicate. In other cases, incremental growth is the goal. In still others, the superior price/performance ratio of a network of workstations is the attractions. In any event, there is a great deal of interest in LANs nowadays.

Local area networks differ from wide area cousins in several ways. The key difference is that WAN designers are nearly always forced by legal, economic and political reasons to use the existing public telephone network, regardless of its technical suitability. In contrast, nothing prevents LAN designers from laying their own high-bandwidth cable, which they nearly always do.

From this one difference spring numerous advantages. To start with, bandwidth is no longer the precious resource that it is in the long haul case, so the protocol designers so not have to stand on their heads to squeeze out the last drop of

performance. Quite different, and usually much simpler, protocols can be used, making the implementation easier.

Another difference is that LAN cable is highly reliable; error rates 1000 times lower than in WANs are normal. This difference also impacts the protocols. With WANs, the low reliability means that error handling must be done in each layer.

With LANs, it may be sufficient to skip the error in the lower layers and just do it higher up, leading to simpler and more efficient protocols in the lower layers.

LAN Protocols

Protocols are the formal rules and conventions governing the exchange of information between computers, defined to provide reliable and efficient transfer of information. Without protocols to guide the orderly exchange of data between points in a network, there would be chaos, not communication.

Detailed protocols are required to precisely define the format in which data and system messages are to be sent; describe how a message is addressed; and govern network traffic flow by controlling priority, routing and sequencing of messages. Only when two devices agree on the specific conventions to be used can conversation take place.

No protocol works in isolation. Rather, it functions as part of the total set of instructions which determine the operations of a device or a network. Each set of protocols is designed to work under different conditions and to satisfy different requirements.

The range of possible methods for passing messages between computers is enormous. The following are specific communication protocols of interest for local area networking:

*** Contention**

- Simple contention
- Carrier Sense Multiple Access (CSMA)
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
- Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
- Polling
- Token Passing

Protocol Evaluation Factors

Precise protocol specification could fill several books with complex mathematical formulas; we will avoid mathematics in favor of general functional

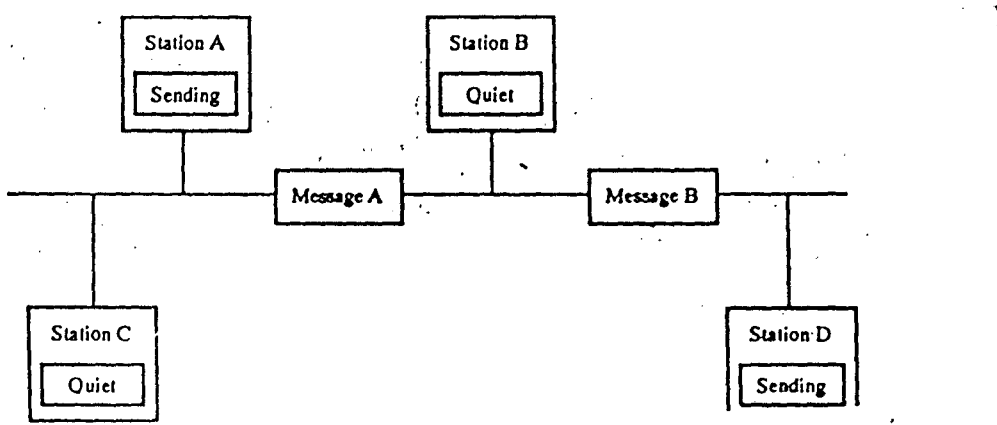


FIG 4.5: WORKSTATIONS ON A CONTENTION NETWORK

any one time?

Traffic volume: How many messages can be passed

Network size constraints: How large can a network using a protocol be?

Performance: Under what conditions does the protocol perform well? Poorly?

Overhead: How much traffic capacity is required to pass control messages?

Access delay: Does the protocol have built-in delays before a workstation can access the network?

Station failures: What happens to the network if the workstation fails?

Expansion: How easily can the protocol accommodate additional workstations?

Contention

Contention is what happens at a staff meeting when several people start to talk at the same time. In contention protocols, no “policeman” controls usage of the communication channels.

All workstations on a contention network share a common transmission channel. Messages are broadcast on that channel and may be overheard by all attached workstations (see Fig 45). A workstation responds only to a message with its address: message intended for different destinations are ignored. When not responding to a specific message, workstations are passive, simply listening in on the channel rather than being actively involved in transmitting messages.

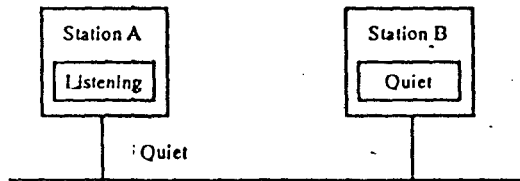
Messages to be transmitted are converted to packets and are sent when ready, without verifying the availability of the channel. When transmission of a packet from one workstation overlaps with that of another, collision occurs. Colliding packets, with their embedded message, are destroyed.

While the basic contention protocol makes no provision for knowing if another message is already underway, it does not provide for acknowledging the successful receipt of the packet. If the originating workstation does not receive an acknowledge, it assumes that transmission was garbled or destroyed. The sending workstation waits a random amount of time and then retransmits the packet. The waiting time must be random or the same message will collide repeatedly.

In some cases, the receiving workstations receives only part of the packet. The receiver may then return a negative acknowledgment to the originator, requesting retransmission.

More than any other network, the connection-based network is characterized by bursty traffic: the time interval needed for each transmission is short in relation to the interval between transmission.

Step 1 :



Step : 2

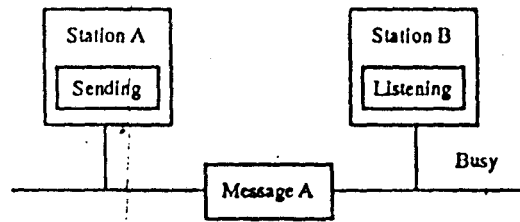


FIG 46 : CSMA

Contention Evaluation Factors.

Smooth functioning of a contention-based network is dependant upon high availability of the transmission media and low collision rate. The network is characterized by the following:

- **Message length:** Messages are divided into short packets in order to reduce the amount of the data that must be rebroadcast after the collisions. Normally, the original message is also fairly short
- **Traffic Volume:** Contention protocols are designed for networks with low traffic volumes, that is, one with few time. Low traffic volume implies a limited number of attached workstations
- **Network length constraints:** The longer the network, the greater the chance of collision. Contention networks are limited by the time needed for a signal to travel the length of the transmission media and have an acknowledgement returned (that is, propagation delay)
- **Performance:** Contention networks are most effective under light to medium load. Performance under those conditions is excellent. Under heavy load, a contention network tends to be unstable, with severe rapidly degrading.
- **Overhead:** Contention networks have high overheads because of the collisions and the need to acknowledge the successful receipt of messages.
- **Access delay:** Delay in the network is generally moderate to long, depending on traffic. Delay under heavy load can be significantly higher than load alone would seem to dictate
- **Station failures:** Because operation of the network is not dependant on the presence or the absence of any one workstation, failure of a workstation inconveniences only its users. Rarely does failure of a single station disrupt service on the whole network
- **Expansion:** Addition of new workstations is relatively easy because to be included in the network, the workstation simply must recognize it own unique address. Expansion may be achieved with minimal disruption of the network.

Refinements on the contention procedures are used by many of the current microcomputer local networks. These refinements are: Carrier Sense Multiple Access (CSMA) in Fig 46.; Carrier Sense Multiple Access with Collision Detection (CSMA/CD); and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

Each will be discussed in separate sections. For all practical purposes, characteristics of these refinements are identical to the characteristics of simple contention.

Carrier Sense Multiple Access

Carrier Sense Multiple Access (CSMA) is a polite staff meeting, with colleagues beginning to talk only when one else is talking. As in simple contention, members of the network share a single communication channel (see Fig 46)

Before information is sent, the workstation “listens” – usually on a secondary frequency – to sense whether any other workstation is using the primary transmission channel (the “carrier”). Only when the line is clear will the workstation transmit.

If a workstation becomes ready to transmit while another workstation is active, it detects the signal passing on the cable and does not send it message until the current transmission is complete. For microcomputer networks the waiting station has two options, depending on system design:

1. It can continually sense the channel while waiting for the busy signal to cease and then transmit immediately. This is called persistent carrier sense, as the terminal actively waits to seize the channel as soon as it becomes free. If other workstations are equally persistent, a collision may occur immediately after the busy signal ceases.
2. Alternatively, if the channel is sensed busy, the terminal reschedules its transmission for a later time, using a random delay, and tries again nonpersistent carrier sense. Fewer collisions occur, resulting in higher throughput. However, delays may be slightly longer, at least in networks with low channel utilization.

In addition to transmitting its message on the main channel, the active workstation broadcasts a carrier-sense signal on the secondary channel to inform the other workstations that the line is busy.

After transmitting, the workstation waits for an acknowledgement, indicating that transmission was successful. If no acknowledgement is received or if a negative acknowledgement (indicating unsuccessful transmission) is received, the workstations assumes a collision has occurred. The workstation then waits a random amount of time before starting the process again.

In a CSMA network, collision between transmitting workstations is still inevitable. A “ready” signal on the secondary channel does not necessarily mean that the network is free of other traffic. Because of the length of time required for a signal to travel the channel (propagation delay), two or more workstations may sense an idle time simultaneously, and thus both attempt to transmit at the same time. If the propagation time is short, the information the workstation hears by monitoring the channel is sufficiently current to permit a useful decision. The probability of success is significantly higher than in simple contention. If, however, the information is old, that is, the propagation delay is long, CSMA offers only slight improvement over plain contention.

The secondary channel which carries the busy tone does require some bandwidth. This bandwidth is generally minimal. There is a brief delay involved in sensing the busy signal.

Carrier Sense Multiple Access with Collision Detection

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) provides the proper etiquette for the times when polite colleagues inadvertently start talking at the same instant. At our theoretical staff meeting, both speakers would stop and wait for the other to continue. The one who resumes first would have the floor.

In CSMA/CD, in addition to sensing whether the transmission channel is in use before beginning to transmit, workstations monitor the link during transmission. When a collision is detected, transmission is halted.

As in the other contention-based protocols, the messages is retransmitted after a brief interval. For CSMA/CD, the interval may either be random or pre-defined as a unique period for each workstation. Because of the ability to listen before and during transmission, the number of collisions is relatively low. Successive collisions between the same workstations is rare. Additionally, since transmission ceases as soon as a collision is detected, less delay occurs

Carrier Sense Multiple Access with Collision Avoidance

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is analogous to several people wishing to contribute information at a meeting, and therefore all raising their hands at the same time. At the meeting, the moderate selects the next speaker. In CSMA/CA, the protocol determines who speaks next.

A station with a message to transmit monitors the medium and waits for the line to be available. When the channel is clear, the workstation signals its intention to broadcast. If multiple workstations are waiting, the order of precedence is determined by a pre-established table.

Just as human meetings tend to be biased in favor of allowing the main speaker or resident expert the greatest opportunity to talk, most CSMA/CA schemes are biased in favor of the lower numbered workstations. That is, after any transmission, the workstation designated as first by the table has the right to transmit. If it has no message to send, or if it fails to transmit within a pre-defined time for any reason, the next workstation has the chance, and so on.

Once any workstation transmits, the network begins again at the top of the list of precedence. Refinements are possible. Some implementations are designed to avoid having a network dominated by any one workstation: a station which has just transmitted may not be allowed to transmit again until all other stations have had an opportunity.

To accommodate multi message dialogue, the workstation receiving a message may have the first right to transmit. If it does reply, the original sender would again have a chance. Unfortunately, two workstations may seize the medium, with one station sending messages and the other replying.

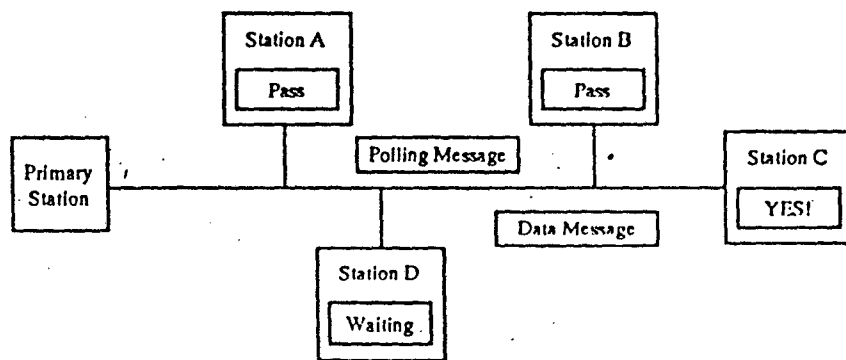


FIG 47: POLLING

In the case, when no workstation has a message to transmit, the network may reinitialize, starting again from the first workstation. In other cases, the network may enter a free-for-all period, in which the first workstation to transmit gains the channel. During the contention period, collisions are allowed.

Polling

Polling (see Fig 47) involves the central control of all workstations in a network. The central, or primary, workstation acts like a teacher going down the rows of the classroom asking each student for homework. When one student has answered, the next is given a chance to respond.

A polling network contains two classes of workstations, the primary workstation (also termed the central controller or server), and the multiple secondary workstations connected to it. A buffer that can temporarily store messages is associated with each secondary workstation. When a workstation has information to transmit, the data is passed to the buffer. The message is held until the workstation is polled by the central controller.

The primary workstation queries each secondary in turn to determine if it has a message to transmit. If the answer is affirmative, the workstation is either given permission to transmit immediately or assigned a transmission time. The amount of time a workstation may have in which to transmit once channel is gained is determined by system parameters.

If the workstation does not have data to transmit, it still must respond with a short control message. Rather than returning a message to the primary, some networks allow a polled workstation with no data to send to pass the polling signal to the next secondary station.

Each time a secondary workstation is polled, the primary workstation must wait for a response to be returned. After a workstation responds, the next station is polled. The primary workstation determines which workstation has access to the network at any one time.

There are two possibilities for the path of a message from source to destination workstation:

1. All messages may be required to pass to the central workstation, which routed them to their destination
2. Messages may be sent directly from the originator to their destinations.

In either case, communication between workstations is possible only under the direction of the polling computer.

Variations on polling tend to be concerned with how often workstations are queried. The basic protocol calls for all stations to have equal opportunity to broadcast. This is not always so. In some networks, workstations considered to be very active or to have priority may be polled several times within a single cycle. In other cases, an inactive device may not be polled every cycle. A third alternative is

that the frequency with which individual workstations are polled may be varied to reflect their current activity level.

Polling techniques can be said to maintain a tighter control over the network than so contention-based protocol.

Polling Evaluation Factors

The polling network is characterized by the following:

Message length: Allowable message length tends to be longer than in contention networks, as no workstations can pre-empt the network through frequent, lengthy messages. However, if all workstations have long messages, the transmission delay is high.

Traffic Volume: Polling networks support moderate to high traffic volume, limited primarily by the need to wait for permission to transmit. Direct conflict for time to transmit, as in contention techniques, is avoided. Therefore a large number of workstations can share the common channel.

Network Length Constraints: The distance between workstations and the overall length of the network is limited by the transmission medium, rather than by the polling protocol. As in any network, the greater the length, the longer the time required for messages to travel between sending and receiving stations.

Performance: Polling networks perform best under moderate load. Under heavy load, transmission delays may become unacceptably long. Polling is inefficient for networks with light loads. In the extreme case, most network traffic may be comprised of polling signals and negative acknowledgements, rather than actual messages.

Overhead: Administrative overhead on a polling network is high. The query and response use a measurable part of the total network capacity. Furthermore, in many polling networks, the server units cannot be used as a workstation.

Access Delay: Delays in most polling networks are relatively long. In most implementations, a workstation is polled only once each cycle. If the network is very large, delay might be unacceptably long.

Station Failures: Little or no disruption of the network is caused by a failed secondary workstation. The inactive workstation is “invisible” to the network; that is, it simply fails to respond when polled. However, if the central workstation fails, all communication on the network ceases.

Expansion: In order for the network to be expanded, the primary workstation must be informed and the order of polling revised to reflect the addition. Therefore, expansion is more complex than expansion of a contention network.

Token Passing

Empty Token :

Header	Data Field	Trailer
--------	------------	---------

In Use :

New Header	Destination Address	Source Address	Routing	Data Message	New Trailer
------------	---------------------	----------------	---------	--------------	-------------

FIG 48: TOKEN

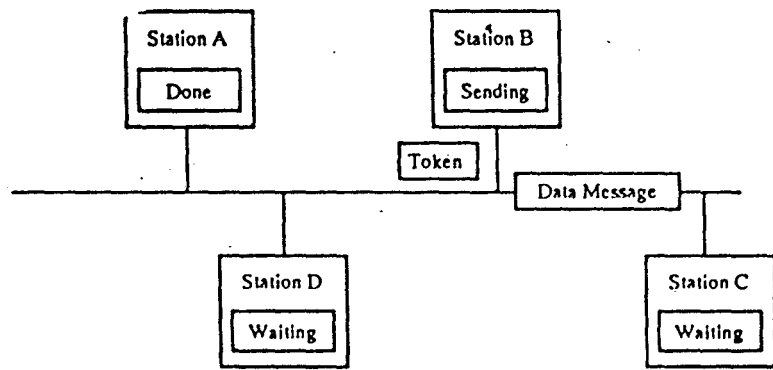


FIG 49 : TOKEN RING

Token can be seen as a children's game of hot potato in reverse. Like the players, the network continuously circulates a special bit pattern known as a token. Rather than being out if you are holding the object being passed, holding the token confers the right to communicate. Only the workstation holding the token can put a message onto the network. Control of the network is decentralized.

Each token contains network information, comprising if a header, a data field and a trailer. (see Fig 48)

When a workstation that wants to transmit receives an empty token, it inserts routing information, inserts the data and sends the token on a complete circuit of the network.

The workstation holding the token may transmit messages up to a specified maximum length. If it does not have anything to communicate, it passes the token to the next station in the network. Fig 49 illustrates token passing.

In this token passing networks, the token passes from one workstation to the one immediately adjacent. However, the token may pass from workstation to workstation in a sequence established at network implementations time, without requiring nodes to be physically adjacent. In such implementations the workstation knows the address of the next workstation to receive the token, as well as its own address.

All workstations on the network read the address in an occupied token; if it is for a different workstation, it is passed on unchanged and unread. At the destination, the receiving workstation reads the message, marks the token as copied or rejected, and continues passing it. Only the workstation which has placed a particular message onto the network may remove that message. If a message is disassociated from the network, it simply does not read the message.

When the token returns to its original sender, the message is removed. The token is marked as empty and forwarded to the next workstation. The sender can either save the message, in order to compare it with the original data as part of the network reliability monitoring scheme, or discard it. Acknowledgement or lack thereof notifies the sender of the status of the message. If the receiving workstation is absent from the network, no acknowledge will be received. If the message has been rejected because it has been garbled, the sending workstation can retransmit it.

Complex error recovery protocols are required to recognize and recover from events, such as lost or garbled tokens, the failure of a workstation to forward the token; or any time that no token exists, such as at network startup. In such cases, a method of generating a token and beginning to circulate it must be specified. Controlled contention or a priority access scheme based on workstation address may be used to re-establish token ownership.

Token passing ensures relatively tight control over the network. The elimination of inefficiencies caused by collisions between contending units is a major advantage for users.

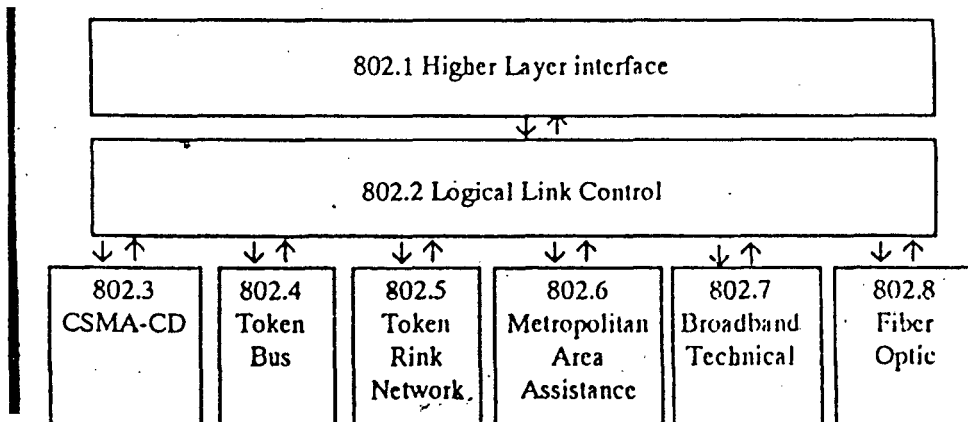


FIG 50: IEEE 802 COMMITTEES

	Transmission Medium	Data Signaling Technique	Rate (Mbps)	Maximum Length (m)
IEEE 802.3 (CSMA/CD)				
Original (10 BASE5)	Coaxial Cable (50 Ω) (Manchester)	Baseband	10	500
Chaperner (10 BASE2)	Coaxial Cable (50 Ω) (Manchester)	Baseband	10	185
10 BASE-T	Unshielded Twisted Pair	Baseband (Manchester)	10	100
Broadband (10 BROAD36)	Coaxial Cable (75 Ω)	DPSK	10	3600
IEEE 802.4 (Token Bus)				
Broadband	Coaxial Cable (75 Ω)	duobinary AM/PSK	1,5,10	a
Carrierband	Coaxial Cable (75 Ω)	FSK	1,5,10	7600
IEEE 802.5 (Token Ring)				
Twisted Pair	Shielded Twisted Pair	Differential Manchester	1,4	b

a = not specified
b = not specified; a maximum of 250 repeaters allowed

TABLE 7: PHYSICAL LAYER SPECIFICATION FOR
IEEE 802 LAN STANDARD

IEEE 802 PROJECT

A local area network, as defined by the Institute of Electrical and Electronics Engineer's (IEEE) Project 802, is a data communication system allowing a number of independent devices to communicate directly with each other, within a moderately sized geographic area, over a physical communication channel of moderate data rate.

The number and kinds of devices a data local area network should connect, the types of services supported, reliability, and so forth, have precisely defined in the various 802 committees. Two levels of requirements exist: requirement applicable to networks in general and requirements for specific types of networks. A few of the more general (and hence more relevant for our purposes) requirements have been summarized below. The list is not all the general requirements, however, nor even of all the relevant requirements.

- **Size:** One LAN should support at least 200 devices and should be able to span at least 2 kilometers. LAN must be capable of being linked to provide service over a greater area
- **Transmission Rate:** Data shall be transmitted through the network at a rate between 1 Megabit per second and 20 Megabits per second.
- **Data Communication Functions:** The data communication supported should include, but are not restricted to: file transfer and transaction processing; file and database access; terminal support ("dumb" terminals, "smart" terminals, high speed graphics terminals, etc.); electronic mail; and voicegrams
- **Attached devices:** Devices interconnected by the LAN should include computers and terminals; mass storage devices, printers, plotters, network and site monitoring and control equipment, bridges and gateways to other networks, telephones; video cameras and monitors; photocopiers; facsimile transreceivers
- **Services:** The LAN should allow a variety of network processes to coexist
- **Expandability:** Adding or removing devices must be easy. Changes should cause minimal disruption, defined as a transient fault lasting no more than one second
- **Resource sharing:** When devices need to share LAN facilities, especially the bandwidth of the bus, this sharing must be fair to all devices, even in overload conditions
- **Reliability:** The LAN should be highly reliable. No more than one packet per year may contain an undetected error.

Very early in the process of defining a LAN and developing standards, the members of the 802 project recognized that no single technology would satisfy all requirements. Specific applications, with their differing priorities, demand different technologies. Therefore, the 802 project divided into several different committees, listed in Fig 50, each focusing on creating separate standards. Table 7 gives physical layer specifications for IEEE 802 LAN standards.

802.1 – Higher Layer Interface Standard

The 802.1 committee is not developing standards, but has focused on issues relevant to all other committees such as addressing of messages, internetworking, network management and higher layer interfaces.

802.2 – Logical Link Control Standard

The 802.2 committee has concentrated on functions necessary to provide a reliable communication path between two devices. Levels of service and standard frame format have been defined and accepted by the IEEE as a standard. Currently, the committee is working on network management.

802.3 – CSMA/CD Bus

The 802.3 is aimed at developing a contention bus network. The standards proposed by the committee were virtually identical to Ethernet specifications published by the DEC-Intel-Xerox collaboration: a 10 megabit per second network, which would allow up to 1000 devices to share a baseband coaxial cable. Although the initial proposal was accepted by the IEEE and the ISO as a standard, the committee has been considering changing the type of cable specified to permit use of thinner wire.

Currently, two other efforts are underway by 802.3. Part of the committee is studying low cost CSMA/CD broadband using a star topology and is in the process of defining how it would work. A second subcommittee has defined a broadband modem which will allow Ethernet to plug into a broadband cable.

802.4 – Token Passing Bus

The 802.4 committee has focused on defining a logical ring on a physical bus so that token passing protocol can be used. Broadband operation with a variety of data rates has been defined.

The specifications have been accepted and are being published as standards. One major problem affecting the usability of token bus is that the committee is standardizing protocols for a complex network that has not been yet widely implemented.

802.5 – Token Passing Ring

The 802.5 has defined a token ring using the star topology to access workstations sequentially. Baseband and broadband versions have been developed. IBM has contributed extensively to this subcommittee. The proposal of the token ring committee was accepted in the fall, 1984.

New Committees

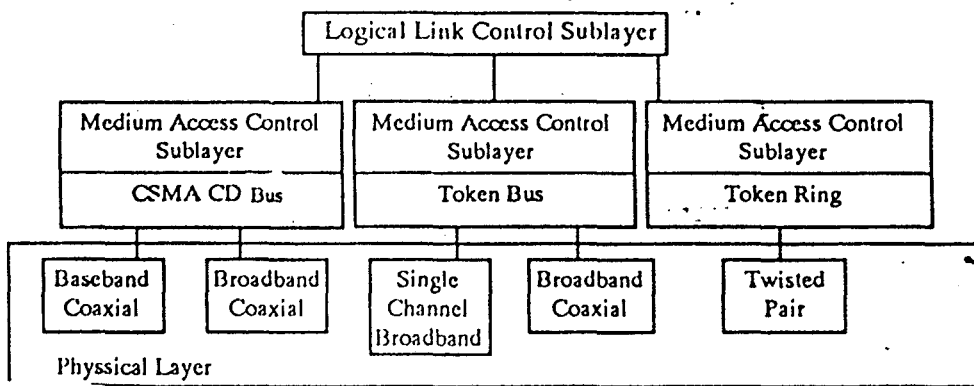


FIG 51: IEEE 802 LAYERS

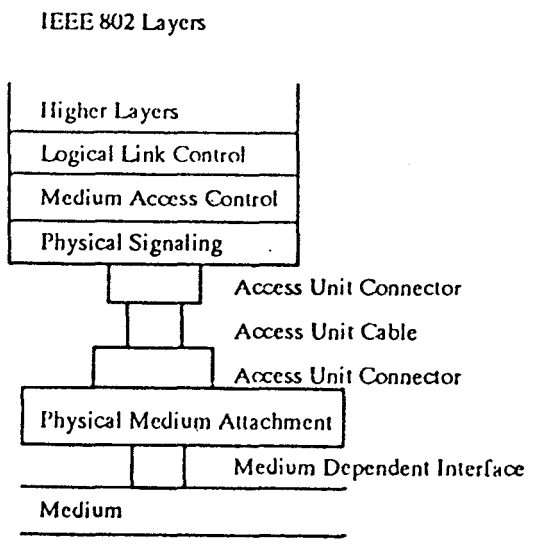
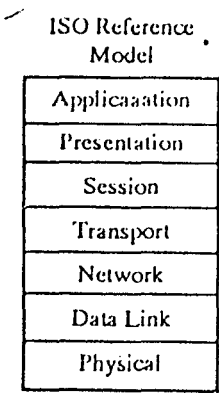


FIG 52(a) : IEEE 802 LAYERS
COMPARED TO ISO LAYERS

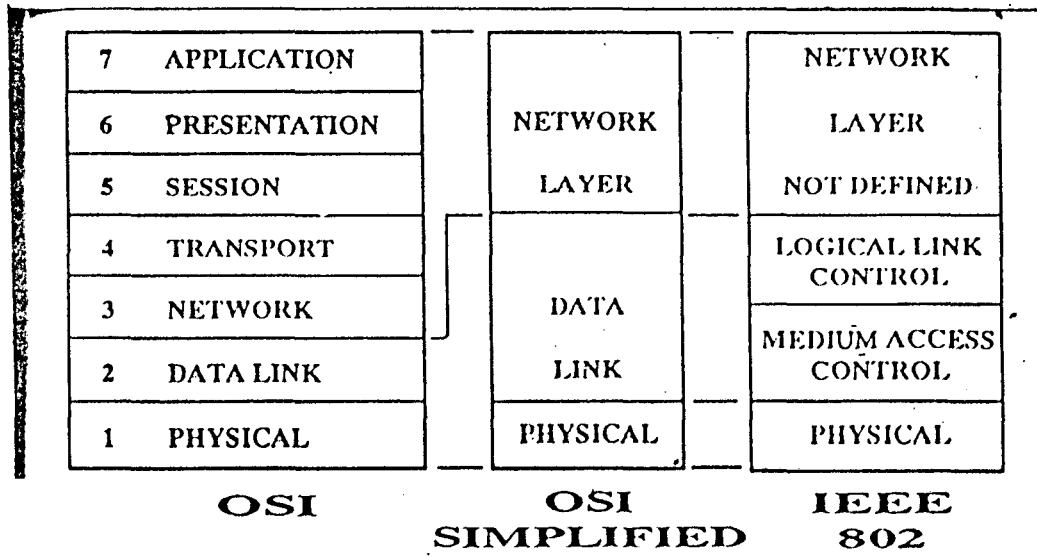


FIG 52(b) : OSI AND IEEE 802.

Three new 802 committees were formed during 1984:

- 802.6 – Metropolitan Area Network: 802.6 is exploring the use of a variety of techniques to send data in a city-wide area. Both CATV and cellular radio are being considered.
- 802.7 – Broadband Technical Assistance: 802.4 was formed to define broadband LAN standards.
- 802.8 – Fiber Optic: The 802.8 committee is the most recently formed group. Members have begun exploring the use of fiber optics for a very high speed local area network.

802 Reference Model

Although the 802 Project committee's overall approach is based upon the structure of the International Standards Organization (ISO) Reference Model of Open System Interconnection, it does differ from the layers defined by the ISO. Moreover, 802 is focused on the lowest ISO layers only, that is, sharing the media. No higher layer protocols are under 802 study.

The 802 Reference Model has three layers, as shown in fig 51

- **Physical:** concerned with the nature of the transmission medium and the details of the device attachment and the electrical signaling
- **Medium Access Control:** focused on methods of sharing a single transmission medium. Typical issues centre on controlling access to the medium, capacity sharing, algorithms and station addressing.
- **Logical Link Control:** concerned with providing a reliable communication path between two devices. The relevant protocols cover the flow of frames between stations; establishing, maintaining and terminating communication between devices; and error control.

Fig 52 contrasts the 802 layers to the ISO layers. Basically, the 802 Physical Layer is designed to correspond to the ISO Physical Layer. However, rather than being a single layer as in the ISO model, the 802 Physical Layer is itself divided into three parts:

- The physical signaling sublayer
- Access-unit interface
- The physical medium attachment

This division is intended to permit the most complex part of the layer, the physical signaling sublayer, to be physically combined with the station logic in the network controller.

Jointly, the 802 Medium Access Control Layer and the Logical Link Control Layer correspond to the ISO Data Link Layer. Again, the division has been made to allow combining with other network functions.

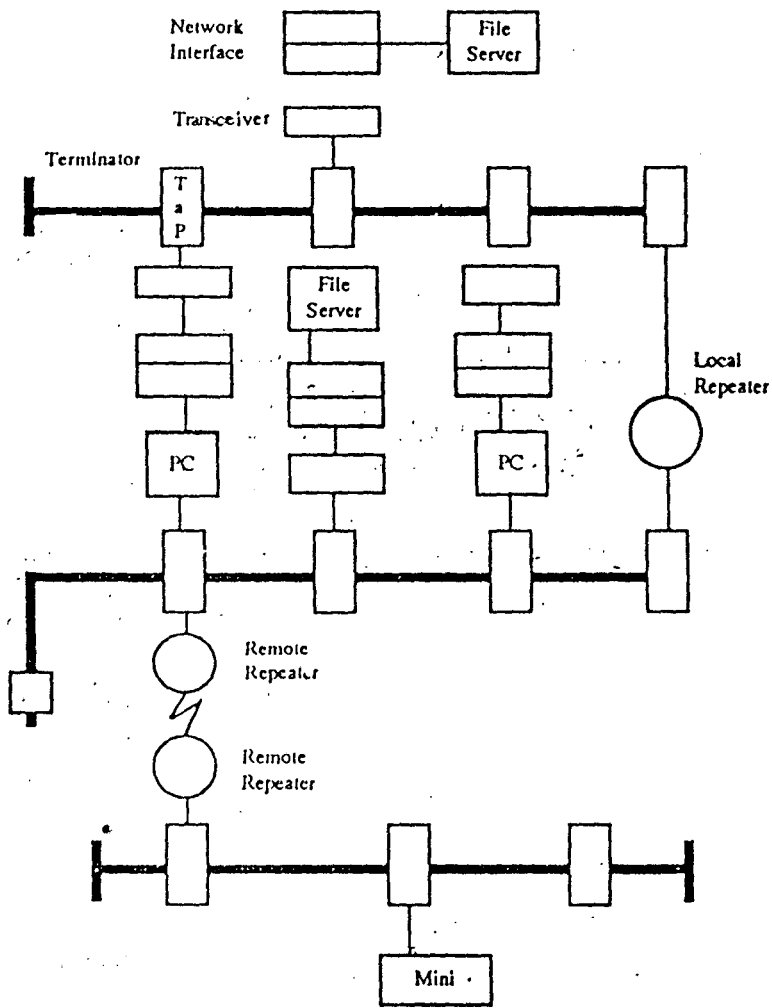


FIG 53: ETHERNET NETWORKS

Ethernet

One of the best known and more successful local area networks is Ethernet (see Fig 53) developed at Xerox's Palo Alto Research Laboratories. The original Ethernet was designed to link a set of single-user minicomputers that were scattered throughout the research centre. Xerox's immediate goals were to enable the exchange of programs and data and to provide access to various specialized peripherals.

Ethernet workstations are connected by a single, multidrop, baseband coaxial cable bus using CSMA/CD. As in all contention schemes, the shared channel is a passive broadcast medium with no central control. Access to the channel by stations wishing to transmit is coordinated by the stations themselves.

Workstations are attached to the main bus through a network interface module. The interface buffers and formats messages and subsequently broadcasts the data onto the cable in bursts. The data is in fixed-length packet containing address information in the header. Current packet size has been defined as 256 bytes, although the technical specification allows packets ranging from 72 bytes to 1526 bytes.

Each station contains address recognition mechanisms, used to identify and accept packets. Every Ethernet workstation, no matter what network it is on, has a unique 48-bit address that is assigned to it and to no other workstation. Hence, when a workstation is moved from one network to another, there is no chance of conflict. This assigning of unique identities has the advantage of flexibility: networks within a company can be physically reconfigured with minimal operating systems reconfiguration.

Data on the network moves at a speed of 10 Megabytes per second, over a maximum distance of 2.5 kilometers. No more than 100 workstations can be connected in a 500 meter segment.

Ethernet's strength is that it provides efficient, high speed resource sharing services within a limited geographic area, at a relatively low cost.

Interest in Ethernet and in local area networks in general, was focused by the 1980 announcement for Digital, Intel and Xerox of a joint project to develop specifications for a local communication network. The project's aim was compatibility, providing sufficient information for various manufacturers such that their widely differing machines could communicate with one another. In effect, the group was establishing a de facto standard.

The attempt was largely successful. The IEEE 802.3 contention bus specification is similar to Ethernet in most details. The two designs are not identical, but are extremely close.

Spurred by these factors, quite a number of vendors have announced hardware and software intended to connect microcomputers into an Ethernet compatible network. At the moment, in fact, the majority of microcomputers LANs use a variation on Ethernet.

- “Ethernet on a chip”, that is, an implementation of Ethernet protocols on a single silicon chip, is available from Intel and other chip vendors
- A fiber optic implementation on Ethernet
- “Cheapernet”, a low cost implementation of Ethernet, is close to being accepted as an IEEE 802.3 standard

Ethernet Specification

The information in this section is based on the Ethernet Data Link and Physical Layer Specification, Version 1.0 published in 1980 by Xerox Corporation.

Specification of Ethernet are:

Topology	-	Bus
Medium	-	coaxial cable
Access method	-	CSMA/CD
Speed	-	10 MB
Range	-	2.5 km
Number of nodes	-	1024
Band	-	baseband

Ethernet is designed to do the following things:

1. To be simple – features which would complicate the design without improving the performance are omitted.
2. To be low cost – in order to be suitable medium for interconnection of equipment whose cost continues to fall, Ethernet itself should be cheap.
3. To allow compatibility of all Ethernet installations – the specification avoids optional features, thus allowing any Ethernet station to communicate directly with any other, at physical link and data link levels.
4. To allow single nodes, groups or the whole network to be addressed by a transmission
5. To allow all nodes equal access to the network, on average
6. To prevent any node interfacing with the proper functioning of any other node
7. To be high-speed – the network should operate at a data rate of 10 Mbs
8. To be stable – the network performance in case of data successfully transmitted should not degrade as the amount of data for transmission increases. In other words the system should not clog up s the load increases

9. To keep delays to the to the minimum – no data should be kept waiting longer than necessary for transmission
10. To have a layered architecture – the physical and data link layers specified are completely independent and correspond to the two lowest layers of the ISO model

Ethernet does not do the following things:

1. Provide full-duplex communication. Only one device can talk at once. The appearance of two-way communication can only be provided by two devices talking alternatively in rapid succession
2. Provide Error Control. The layer specified only detect bit errors and collisions. Recovery from these and other errors must be handled by the higher layers of the network
3. Provide Security. There is no encryption or restricted access implied in this specification
4. Provide variable speed. The network operates at 10 Mbps/second
5. Provide priority control. All nodes have equal access right to the network.

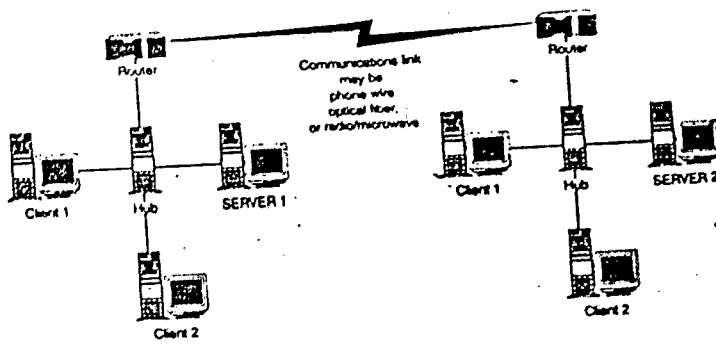


FIG 53 (a): A BASIC WAN CONFIGURATION

WIDE AREA NETWORK

When networks are distributed beyond the limits of local area network topology, it is time to begin thinking about building a WAN.

Wide Area Network(WAN) are so named because it is a network that links together geographically dispersed computer networks. A WAN is basically two or more LANs tied together using high-speed phone lines(such as T1s or 56K frame relay). Beyond that, it is difficult to make generalizations about WANs, other than saying they link many sites into common resources.

Virtual Private Network (VPN) It is a method of connecting networks that uses the Internet to carry data

Tunneling protocol A protocol that ensures that data passing over a company's Virtual Private Network is secure. Tunneling is similar to putting a letter /envelope addressed to a non-local company mailstop in another, larger envelope that uses postal mail to send it to another company location. When the mail gets to the non local company mailstop, the mail clerks take it out of the large envelope and send it on to the person to whom it's addressed.

Dial-on-Demand Whenever a user needs a resource on another LAN (a file database access, or whatever), the local LAN catches the request and dials the remote LAN using either garden variety Plain Old Telephone Service (POTS) or a switched digital phone line such as Integrated Digital Network (ISDN).

All of these kinds of networks have been considered WANs; the concept has expanded to include all of them. In the final analysis, a WAN is just a way to extend your network resources beyond the local area. In the Internet, there are a host of ways to do so ranging from expensive digital phone lines to VPNs to dial-up network access to other ways we have not even considered yet.

However, the basic WAN configuration (in which computer users on multiple LANs share resources) is the focus of this chapter (see Figure 53 a.).

WAN Hardware

Several pieces of hardware can be used to link WANs. Of these, the most common devices are bridges, gateway, and routers.

Bridges

A bridge is a network device that essentially does what its name describes: bridges two LANs together. The difference between a bridge and router is based on the way they link networks. In a telecommunications network, a bridge is a hardware device or software that copies Layer 2 packets (see the following note) from one network to another network. For example, two LANs can be connected with a bridge, a digital phone line, and another bridge at the other end. A bridge connects networks

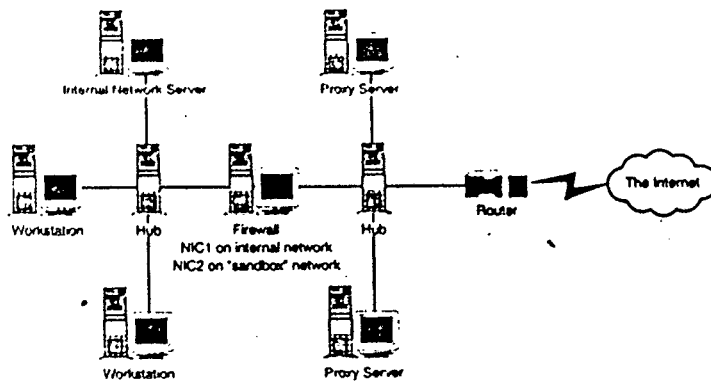


FIG 54: PROXY SERVERS
ACTING AS GATEWAYS

that use similar protocols (such as IP) at both ends of a connection. By contrast, a gateway connects networks using different protocols (such as IPX and IP).

Bridges resemble their newer and more widely used siblings, routers. Bridges are generally single-rack-space units that have a network connection at one end and a frame relay connection at the other. The network connection ties into the local hub/switch, and the frame relay connection ties into a digital phone.

When a bridge links networks, for all practical purposes, users see a larger version of their current network- they can access remote resources using the same methods they use in their local LAN. Bridges, however, are slow and resource intensive, which why most inter LAN networking today is don over router.

Bridges are often used for networks that use protocols that cannot be routed (for example, Net BIOS or NetBEUI). However, these protocols can be carried over a bridge because the bridge works at the Data-Link layer (which is still concerned with hardware) rather than the Network layer (where data packet routing depends on software)

Gateways: Protocol Translators

The term gateway can refer to variety of different devices. At its most basic, a gateway is a device that acts as a two-way path between networks. For example, in an Internet-connected network, a proxy server can be a gateway between the internal network and the external Internet(see Figure 54).

Another common example of a gateway is any device that passes packets from one network to another network around the Internet. Routers and bridges loosely belong to the global group called gateways, so the gateways discussed in this section have specific purposes other than routing or bridging packets.

Gateways link networks together. As noted earlier, gateways are different from bridges in that they can create junctions between dissimilar networks; this can come in very useful for networks that do not run TCP/IP. Gateways that can translate one protocol to another are protocol translators.

Protocol translator:- A device that can translate between two network protocols. Typically, protocol translators translate Net Ware IPX to TCP/IP so that users on an IPX network can access the Internet or IP resources.

Protocol translators are not commonly used to link LANs into WANs; these days, protocol translators are often used to translate between Net Ware's IPX protocol and the TCP/IP protocol so that an IPX based network can connect to the Internet. If you have chosen to build (or have inherited) a Net Ware LAN, protocol translation may be the best way to provide Internet access to your users. However, Net Ware's main protocol, IPX, is routable, so a router is probably a better choice if you want to join two LANs into a WAN.

Routers

A router is a device that passes data between multiple networks. It works at the OSI Network layer(Layer3), which means that it must be able to understand the data packets so that it can route them to their destination. Routers are essentially computers optimized for handling packets that have to be transferred between separate networks. Not surprisingly, routers attempt to send packets from their source to their destination in the fastest way possible, which (as you'll see) is not always the absolute shortest path.

Router:- A device that handles the traffic flow for data packets that are not addressed inside the local network. In other words, a router is the long-distance post office sorting machine.

If you send mail locally, it can go to the local post office and they'll deliver it. However, if the mail has to go to Timbuktu (which we'll assume isn't local for you), someone will notice that the address on the mail isn't local and will forward it to a post office location that handles long-distance mail. The person who sorts mail and separates long-distance mail from the rest of the local mail is like a router.

How Routers Route Data: Gateway Protocols

Routers variously use one or more of four pieces of the TCP/IP protocol suite to determine which route a packet should take to a given destination at any time. These four pieces are collectively referred to as gateway protocols, a needlessly confusing term. A more descriptive and accurate term would be router protocols because routers use them to determine the proper way to forward data packets.

Gateway protocols :- Members of the TCP/IP protocol suite that routers use to determine the best route for data packets.

These four pieces are called Routing Information Protocol (RIP), Open Shortest Path First(OSPF), Border Gateway Protocol (BGP), and Exterior Gateway Protocol (EGP).

Two of these four protocols (RIP and OSPF) are called interior gateway protocols; they deal with routing data only within a self-contained network such as a LAN or WAN. The other two protocols (BGP and EGP) are called (not surprisingly) exterior gateway protocols and are used to route data outside a LAN or WAN.

IsoEthernet

Many of you have probably not heard much about isochronous Ethernet, more widely known as isoEthernet. That's because isoEthernet has fairly limited industry support that it does have is from some very large and influential companies, such as National Semiconductor and IBM, isoEthernet is potentially an important means of

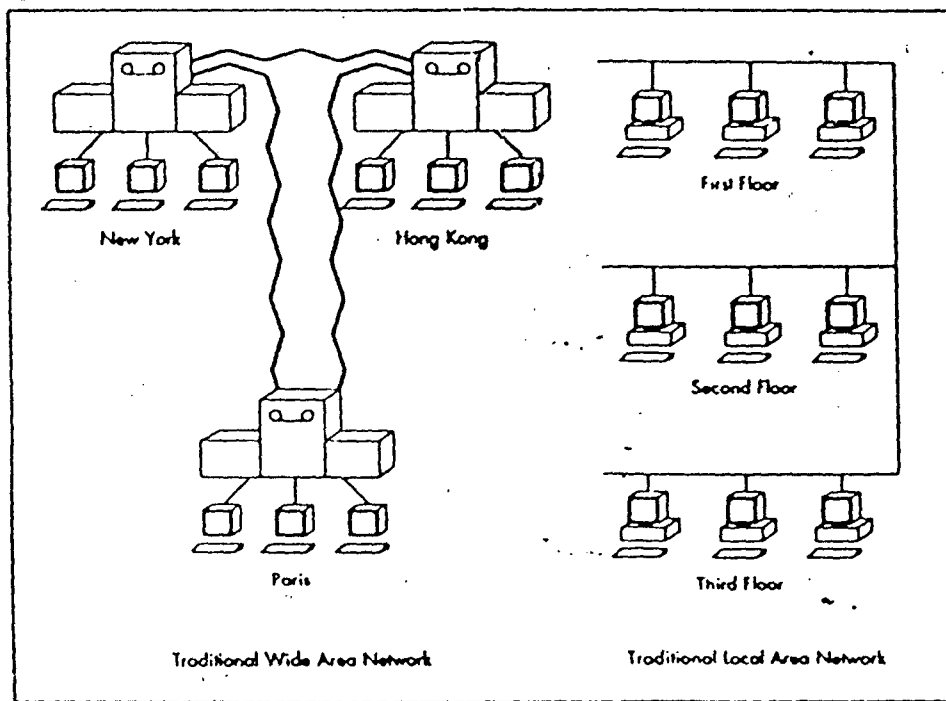


FIG 55: DIFFERENCE BETWEEN TRADITIONAL
WIDE AREA NETWORKS AND LOCAL AREA
NETWORKS

integrating LANs with wide area links. IsoEthernet is especially viable because it can be integrated with ISDN public networks and is capable of upward migration to ATM.

IsoEthernet is not a high-speed networking protocol per se. Instead, it is a technology that lets you revitalize your existing 10 Base-T network by letting you take multimedia, wide area, and telephony traffic out of the LAN infrastructure and support it on a separate channel.

The IEEE 802.9a (IsoEthernet) Standard

Introduced in 1992 by National Semiconductor, isoEthernet is similar to the more familiar 10Base-T. It offers 10Mbps Ethernet support on shared, twisted-pair copper cable, and it adheres to all of the cabling rules of 10Base-T. It varies from 10Base-T, however, in that it dedicates some circuits to time-sensitive traffic, such as video. The IEEE 802.9a spec is similar to standards such as 802.3 Ethernet and 802.5 Token Ring in that it defines the media access-control(MAC) and physical layers of the standard. In addition, 802.9a specifies a signaling layer based on ISDN protocols and using existing 10Base-t infrastructure. Therefore, it can be implemented in a legacy network without replacing cable or equipment.

WAN APPLICATIONS

Wide area networking has its own caveats and concerns. And when your goal is to make a WAN out of LANs, the job gets even more complicated! Often a company starts building LANs that work fine within the head office, only to find a huge job and a lot of compromises facing them when the time comes to connect all those LANs together.

WANs evolved from early telegraph networks and grew up to enable computers in widely dispersed locations to communicate. WANs predate LANs, partly because in the early days there usually were not enough computers in one location for a LAN to make any kind of sense.

WANs and LANs have many things in common, but they differ in important ways too(see Figure 55). While LANs are usually built for speed, WANs tend to emphasize short, reliable lines; traditional WANs, those that grew up with the telephone system, have featured to deal with unreliable lines, complex meshed networks, multiple paths between nodes, and a variety of communications technologies. LAN manufacturers are steadily adding WAN capabilities to their products, though the job is not complete.

But what are WANs good for in today's networking environment? E-mail probably first springs to mind. Companies with offices spread across town or across the world often can benefit from single, global E-mail system that lets any user easily send mail to any other user. Files can be attached to E-mail messages, allowing a limited form of file transfer.

Another important use of WAN is in data collection. Organisations with large numbers of retail outlets, such as fast food chains and convenience shops, benefit by

being able to collect sales and inventory data from each location either at the end of or during the day. This allows central computers to plan for restocking and to get accurate information on the state of the company. Public utilities are also making use of data collection to read meters remotely, reducing the need to send people out into the field.

By making file transfer easier, WANs are allowing work group to be spread across the globe. In one typical application, a securities firm has financial researchers in major cities such as Tokyo, HongKong, London and New York. Each researcher adds his analysis to the growing report via a distributed desktop publishing system, and the document is formatted and published in New York.

A more advanced application is the networked database, which allows users all over a network to access and update a single, consistent view of data. Depending on the sophistication of the database, it may either store the data in a central location, or it may distribute and replicate data at various sites. A centrally stored database is easier to write and implement, but it has the disadvantages of requiring heavy network usage to access the data, of being a single point of failure, and of being totally inaccessible if the WAN link into it is down. A distributed and replicated database puts data where it is most often used and keeps copies of frequently used data at each local network. The database application then communicates between nodes, keeping records in each location consistent and fetching requested records from remote servers as needed. If the WAN is down, users can generally use the database locally, though data on remote nodes can not be accessed.

WANs have many other uses, including software distribution; the ability to use expensive resources, such as computer servers and high resolution colour graphic printers, remotely; and, importantly, the ability to manage networks centrally.

As networks get bigger and more complex, and as corporate management begins to realize how strategically important LANs have become, there is often a desire to incorporate LANs into the existing corporate information structure in order to assure that they are managed competently. Also, by centralizing network management, it becomes less necessary to have permanent employees with network management expertise at each remote LAN location, possibly saving money. Remote network management lets this centralization happen.

FUTURE OF NETWORKING

Wireless LANs

Mobility is important. We like to be able to move around the office and think while staying in contact with our network. On factory and warehouse floors, we like to be able to hook into the network while out on the shop floor. Doctors in hospitals love the thought of being able to carry a live, dynamically updated patient chart around with them.

Unfortunately, wired networks can't readily provide these things. Wired networks tend to make the tacit assumption that the user works in one physical location and is readily tied to that location.

However, new technologies spurred by the ratification of IEEE standard 802.11 for wireless networking have recently come onto the market. Wireless networking holds a great deal of promise to connect workers who move around their workspace – doctors, factory workers, anyone who doesn't work in a fixed office location. Wireless networks enable people to have data where they need it, when they need it. The technologies are new and relatively expensive, but they herald a new phase of networking that's more focused on the user's needs than ever before.

Why Use Wireless Networks?

For most of us, a hardwired connection between our desktop computer and the rest of the network is sufficient. Basically, all most users need is a single connection to a network; almost no one carries a computer while walking around a workspace, expecting to remain logged into a network. After all, most of us work at our desks and leave our desks only to get additional materials with which to do our jobs.

That's all fine and well for office workers, whose jobs do fit the preceding description. But there is a small and steadily increasing group of people who have to be able to connect to network from a variety of locations in multiple locations in the office, while they're on the road, and so forth. Often, the dual solution of Ethernet in the office and dial-up networking on the road is inadequate for one reason or another – and then the poor network manager has to figure out how to provide the services his or her users require, which can be quite a conundrum.

Why? The simple answer is that wireless networking is complex and not entirely standardized. It's currently a relatively obscure branch of networking. Many administrators (and, of course, users) will forgo the convenience of near-universal connectivity when confronted with the complexity inherent in wireless networking.

Recent advances in standardization – not technology – have enabled the wireless networking market to grow. The growth of the web stirred interests in wireless web access. That, combined with the push for wireless-network standardization over the last several years, has led to a surge of growth in the industry.

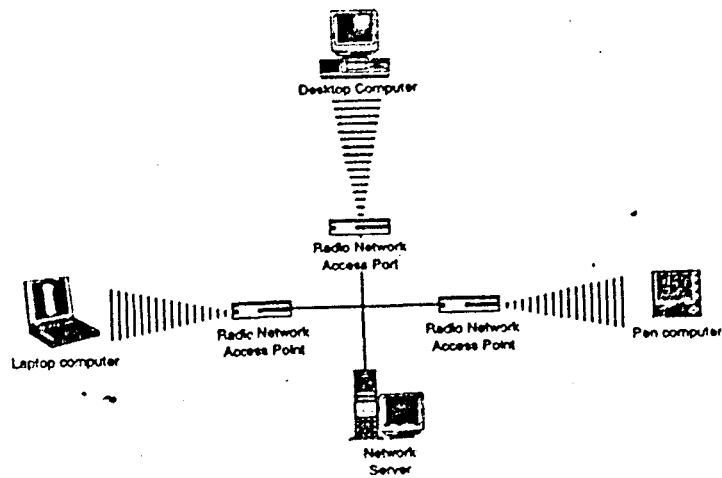


FIG 56: A WIRELESS NETWORK
WITH SEVERAL ETHERNET ACCESS
POINTS

It's still not a mainstream technology, but it's just a matter of time until it will be possible to order a new computer with a wireless network adapter.

What is Wireless Networking?

Here are some identifying features:

- (a) First and foremost, it's data carried over radio waves or with infrared light
- (b) It's a plethora of standards built around the IEEE 802.11 standards and the TCP/IP protocol. IEEE 802 standards specify the physical layer of the OSI network model. TCP/IP does not care what underlying network media it runs over; between the two, it's possible to build a network that runs over any media – coaxial cable, unshielded twisted-pair wire, glass fiber, and, of course, radio waves.
- (c) Wireless networks are versatile ways to transfer data. They can run over a variety of radio waves, from the infrared spectrum to cellular phone bands.

Typically, wireless LANs are not completely wireless (although completely wireless LANs do exist). Most often, wireless LANs are built in a way similar to cellular networks, with several wireless access points connected to a standard Ethernet network. Take a look at Figure 56 for a common wireless LAN configuration.

When a computer connected to a wireless LAN moves around the network, the computer senses which access point the laptop is closest to and uses that access point. The local range of spread-spectrum networks covers up to about 50,000 square feet (which isn't as big as it sounds: 50,000 square feet is a two-dimensional square about 225 feet on a side). Some manufacturers make systems that, with the use of external antennae, can transmit as much as five miles.

Wireless Network Applications

Different kinds of wireless networks lend themselves to different applications. A list of applications and the common wireless topologies is provided in Table 8.

Table 8 : Some Common Wireless Networking Applications

Application	Topology
In-office	Spread spectrum connectivity; the ability to roam in the office with a laptop computer and remain connected to the LAN.
Wireless networking at fixed location	Infrared LANs, radio LANs
Wireless networking Out-of-office	Radio modems connected with VPN wireless network provider (RadioMail, Sprint, MCI, and so on)

Within each of these divisions are multiple (and often competing) standards. As we can see, there's quite a bit of variance with regard to connection type; some use the laptop computers existing infrared equipment, some use special wireless Ethernet hardware, and some use third-party service providers with still more specialized modems.

In-Office Wireless Networking Technologies

In the office, the two most common topologies are spread spectrum and infrared connections. Of the two, spread spectrum, a radio technology that uses the 902-to-928 MHz and 2.4-to-2.484 GHz Industrial, Scientific, and Medical (ISM) radio frequency (which fortunately, requires no FCC license) is more useful for intraoffice mobile workers because it can connect all over a building. By contrast, infrared is used only for line-of-sight applications.

Spread Spectrum

Spread spectrum, the most common topology for wireless LANs, was developed for the U.S Army to guard against enemy radio frequency jamming and eavesdropping. Spread spectrum spreads the signal across a range of frequencies in the ISM bands.

Frequency Hopping Spread Spectrum

Frequency hopping spread spectrum (FHSS), the first implementation of the spread spectrum technology, hops from frequency to frequency in a set pattern. The receiver can receive frequency hopping spread spectrum data only if the sender and the receiver use the same hopping pattern (which is controlled by what is called a hopping-sequence algorithm). According to FCC rules, no transmitter can stay on a single band for more than 0.4 seconds within a period of 20 seconds for the 902 MHz band (30 seconds for the 2.4 GHz band). Each transmitter must also cycle through 50 to 75 radio bands before restarting the hopping sequence algorithm. The IEEE 802.11 standard proposes limiting FHSS to the 2.4 GHz band.

Direct Sequence Spread Spectrum

The other, more recent, type of spread spectrum technology is direct sequence spread spectrum (DSSS), which is more commonly used in wireless LANs. In DSSS, the transmitter modifies the data with "chips" or extra data bits (a minimum of 10) inserted in the data stream. Only a receiver that knows the algorithm for the insertion of chips can decipher the code. Because of the effect of the chips, the effective throughput of DSSS is currently limited to 2 megabits per second in the 902 MHz band and a 8 megabits per second – a usable speed – in the 2.4 GHz band.

Infrared Technologies

Infrared technology is typically used in a single office where the user moves around the room with a laptop. Infrared is a line-of-sight technology and is most useful for offices. Infrared is the same technology used for television remote controls; it can carry a lot of bandwidth, but because it's line-of-sight, it's easily interrupted by any visual obstruction.

Out-of-Office Wireless Network Technologies

The third topology, radio modems, is currently in a state of flux. The technology used is called Cellular Digital Packet Data (CDPD), and it enables a user to send data packets using a cellular network similar to what is used for cellphones. CDPD is still in a nascent state. CDPD is the fastest wireless networking protocol available, but even so, it's limited to a 19.2 kilobits-per-second data transmission speed – quite a bit slower than today's regular wired modems and an order of magnitude slower than an Ethernet connection. Nonetheless, the allure of being able to connect to a network without a phone jack is difficult to resist, and many corporate networkers are cautiously testing CDPD networking for their remote users.

CONCLUSION

The wide area network is where the LAN and the telephone switch both become instruments of data telecommunications. While traditionally LAN managers have known a lot more than their telephony colleagues about data communications, the telephone managers have also had an advantage over LAN managers in their knowledge of telecommunications. Therefore, no matter from which camp we begin, we will have gaps in our knowledge that we'll need to fill in before successfully managing the WAN.

As a general rule, local area networks of all types are going to become more common, and at the same time, more invisible. Microcomputer manufacturers will eventually come to some sort of agreement about networking and start to include LAN facilities within their equipment as standard.

In conclusion, it seems inevitable that computer networks will play an increasingly important role in all our lives. Local area networks will play a large part in network development, although one can expect the divisions between local and wide area networks to become increasingly blurred.

BIBLIOGRAPHY

1. Data Networks by Uyles Black - Prentice Hall.
2. Computer Networking and Distributed Processing - by James Martin - Prentice Hall.
3. Computer Networks by Andrew S Tanenbaum - Prentice Hall.
4. Computer Networks by Uyles Black - Prentice Hall.
5. Telecommuniation Switching System and Networks by Thiagarajan Viswanathan - Prentice Hall.
6. Internet Working with TCP/IP by Douglas E Corner/David L Stevens - Prentice Hall.
7. Telecommunications and the Computer by James Martin - Prentice Hall.
8. Optical Fiber Communication by Gerd Keiser - Mcgraw-Hill International.
10. The Local Area Networks by E C Brooner - Howard W. Sams & Co.
11. Networking by Matt Hayden - SAMS Publishing.
12. Local Area Networks by S K Basandra/S J Jaiswal - Galgotia Publications.
13. Wide Area Networks by Tere Parnell - Tata Mcgraw-Hill.

