Data Communications

for

Minicomputer Users

A General Introduction

TABLE OF CONTENTS

Table of Contents (Cont)

ILLUSTRATIONS

1. INTRODUCTION

The data communications requirements of the minicomputer user differ from those of the mainframe user in several important respects. Communication typically is over shorter distances--fifty or a hundred miles, rather than one or two thousand miles. Minicomputer users typically have only one or two discrete data communication 'links' to remote offices, rather than a complex multi-node data communications 'network'. But the fundamental difference between the minicomputer user's data communications requirements and those of the mainframe user is that the minicomputer only supports, with very few exceptions, asynchronous 'dumb terminals', whereas the mainframe user typically expects his terminals to benefit from some kind of built-in 'communication protocol'.

This introduction will discuss the significance of 'communication protocol' and will clarify the distinction between asynchronous and synchronous terminals. Finally, it will provide a definition of what is meant by the term 'dumb terminal'.

In the second section, we will review how terminals communicate with a computer and the various methods of connecting terminals to a computer. We will also examine why modems are necessary, how they work, how much they cost, and what facilities are required from the telephone company to use with them.

In section 3, we will discuss what can be done to reduce the cost of phone lines and modems when many terminals are connected to a computer. The available types of multiplexor will be described and compared.

Section 4 discusses the concept of 'Add-On Data Link Control' for users of 'dumb terminals'.

Finally, in the last section, we will discuss what can be done to reduce the cost of computer ports; we will also address the special needs of the user with multiple minicomputers, or with a mainframe and one or more minicomputers, all supporting the same 'dumb terminal' population.

1.1 The Significance of Communication Protocol

Communication protocol is a set of rules governing information flow in a communication system. The rules include a definition of the block format or 'message envelope' which will be used to 'packetize' each message to be transmitted. The 'message envelope' usually contains special control characters to mark its beginning and end, along with an address, so that messages can be directed to selected terminals. It also usually includes a sequence number and/or block check character so that the receiving terminal may check the incoming message for errors. The protocol includes a set of rules which define how a terminal 'acknowledges' a message or, in the event it detects an error, how a terminal requests a retransmission.

If a terminal conforms to one of the defined communication protocols, it operates error-free because it has the benefit of automatic retransmission-on-error; it can also be 'multidropped', either individually or in clusters, with other terminals on a single line because it can be selectively addressed or 'polled' by the host computer (see Figure 1-1).

1.1.1 Polling Explained

'Polling' involves the addressing by the host computer of each terminal on the line, one after the other. The computer polls the first terminal which responds 'NAK' if it has nothing to transmit, or 'ACK', followed by its message if it has a message to transmit. The computer then polls the next terminal in sequence. If any terminal does not respond, the computer will 'time out' and proceed to poll the next terminal. Polling takes place constantly, in round-robin fashion. Outbound messages from the computer are also transmitted to each terminal when it is due to be polled. Typically, the ACK-NAK protocol is also used to verify correct receipt of messages or request retransmission.

Terminal polling is only possible if three criteria are satisfied:

(a) The terminal must be 'smart' enough to have an 'address', and be able to respond when it reads its address in a message received on the line.

(b) The terminal must be 'buffered', since it only has access to the line at the discretion of the computer. Such a system is only efficient if the message has been entered at the terminal and is ready for transmission in the terminal's buffer when the terminal is polled.

(c) The host computer must have software available to perform the polling procedure and support the particular communication protocol which the terminals are designed to use.

Typically, these criteria are only satisfied in terminals and data communication systems supplied by the large mainframe manufacturers or designed to be compatible with the product offerings of the mainframe manufacturers. Thus, the cost advantages of being able to put multiple terminals on one telephone line, and the benefits of automatic retransmission-on-error, are typically not available to

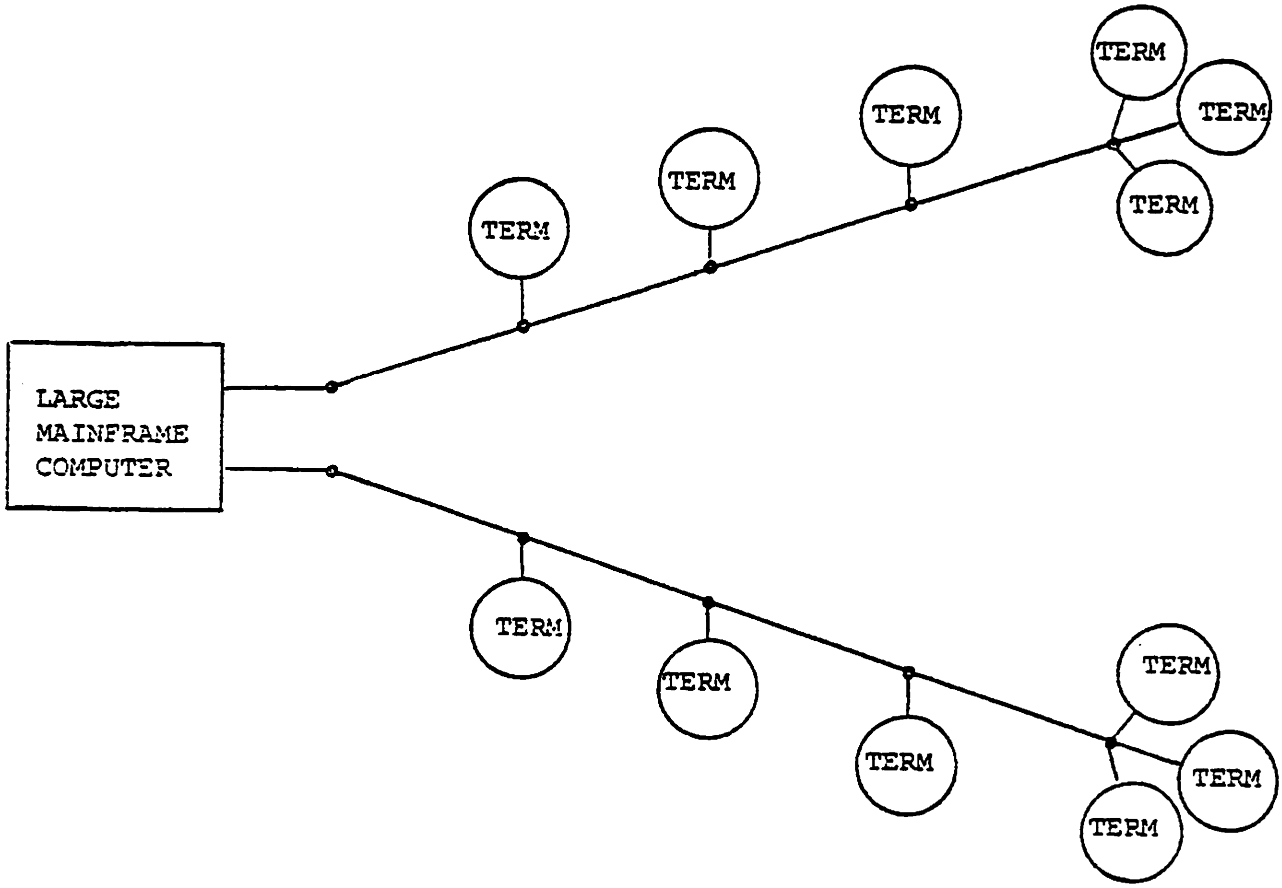


Figure 1-1. <u>Typical Polled Terminal Configuration</u>

All of the minicomputer manufacturers offer software support for various communication protocols, especially the popular IBM Bisync protocol. In addition, DEC, for example, supports its own DDCMP, and Data General offers support for the international standard packet-network access protocol, X.25. But the software support is not designed for communication with terminals. It is designed to permit the minicomputer to be connected to an IBM mainframe, emulating an IBM BISYNC terminal, or it is designed to facilitate computer to computer communication using packet-switched networks (in the case of Data General) or DEC's own DECNET network architecture (see Figure 1-3).

the minicomputer user. He is typically forced to use the so-called 'dumb terminals' described in Section 1.3 and configure his terminals one per computer port as shown in Figure 1-2.

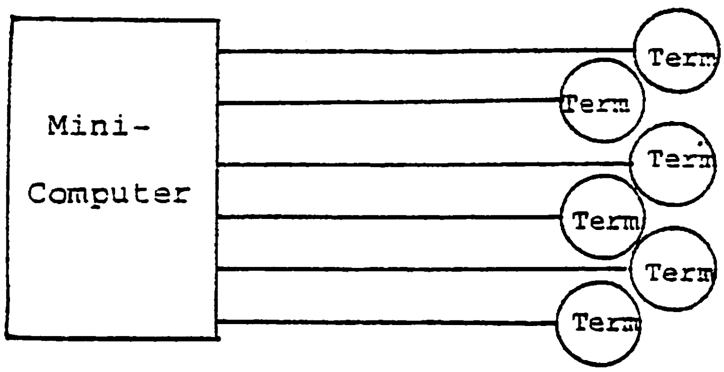


Figure 1-2. <u>Typical 'Protocol-less' Terminal Configuration</u>

1.2 Synchronous and Asynchronous Terminals Compared

Data characters in a computer system are represented by a data code, each element of which consists of a group of bits (binary digits) each one of which can take the value 1 or 0. The group of bits is called a word or byte. The object of serial data transmission is to send these bytes from point to point along a single line or channel.

The method used is to define successive short intervals of time as representing successive bits in the byte, as in Figure 1-4. Two possible conditions, termed mark and space, (from telegraph terminology) are imposed on the line by the transmitting terminal to represent binary 1 and 0 respectively. If the receiving terminal samples the line at the same intervals starting at the same time, it will be able to recreate the byte at the receiving end. (Note that we have used the word terminal to designate either end of the line. In practice the terminal at one end is usually the computer).

Two techniques are used to 'synchronize' the receiving terminal with the transmitting terminal. The simpler technique, more commonly used in the minicomputer world, is called 'asynchronous' data transmission; the more complex; higher-speed terminals operate 'synchronously'.

In asynchronous terminals, the transmitter always reverts to a mark condition for at least one interval at the end of each byte and always goes to a space condition for one interval before starting to send the next byte, as shown in Figure 1-5. This allows the receiver to synchronize with the transmitter at the start of each byte and start its sampling at the correct instant.

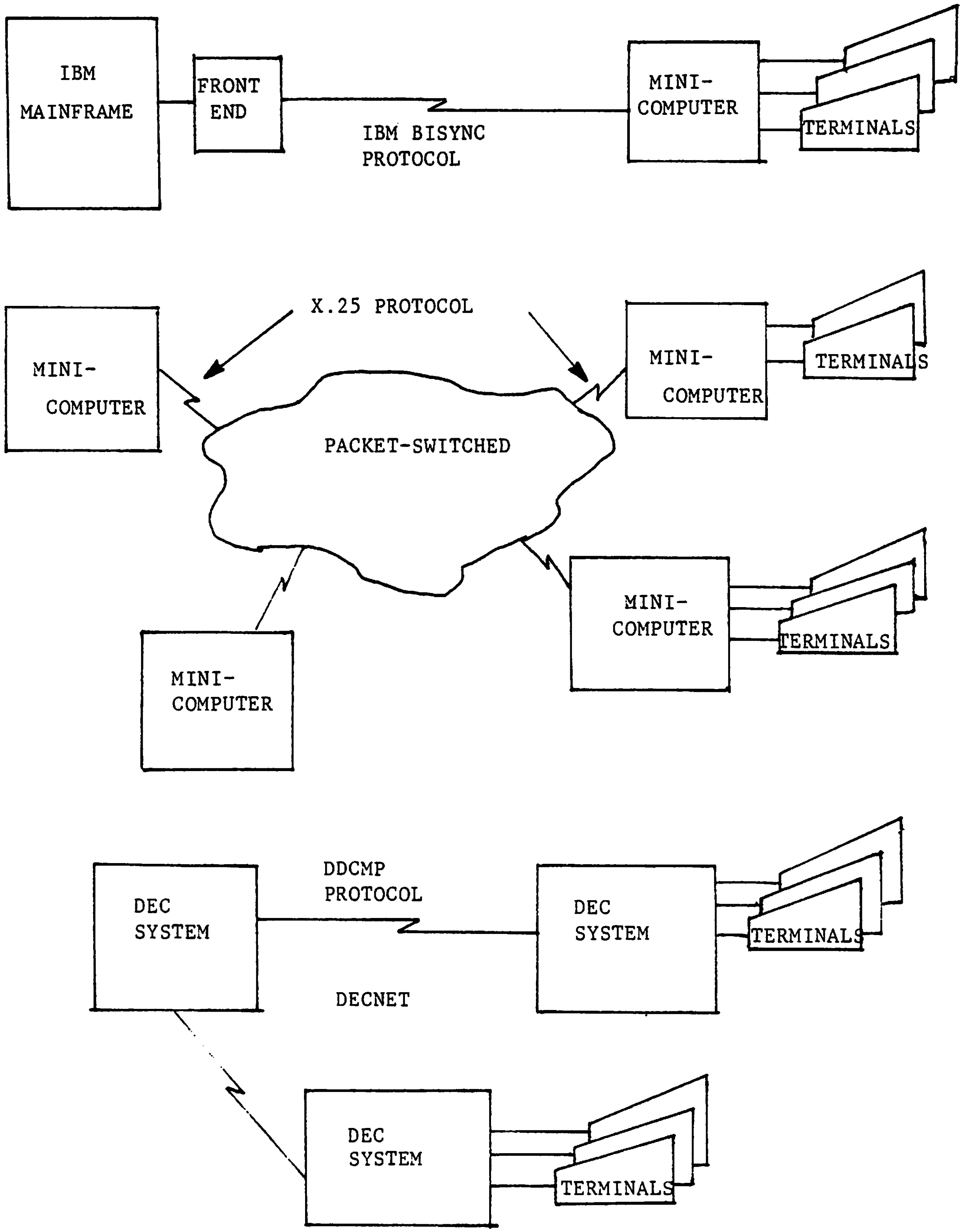


Figure 1-3. Communication Protocols and the Minicomputer

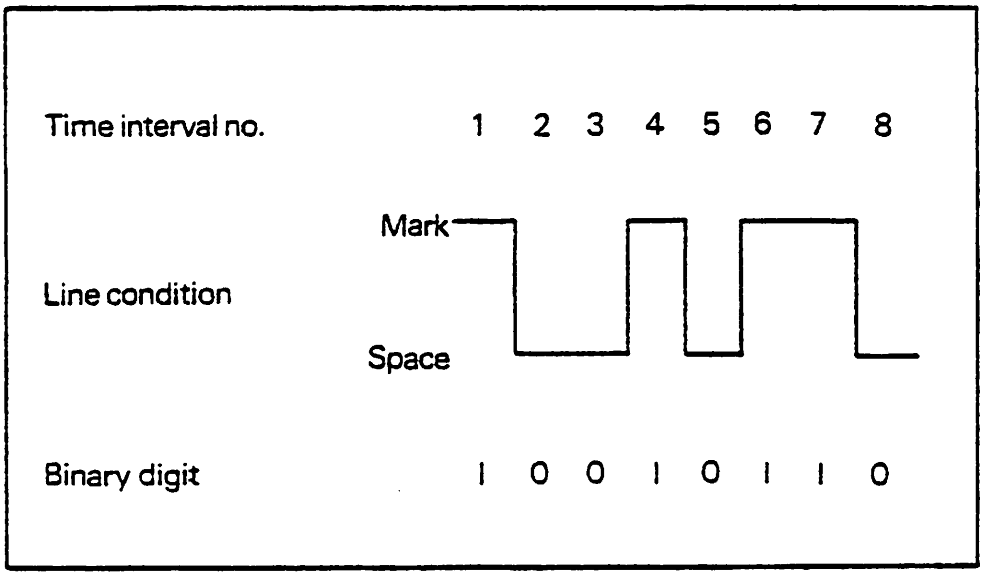


Figure 1-4. Serial Data Representation

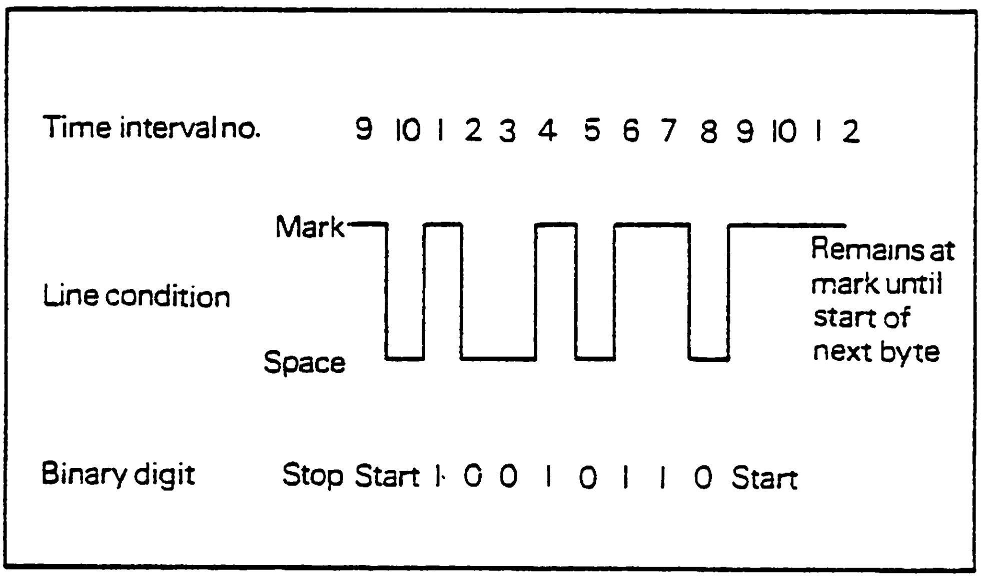


Figure 1-5. Asynchronous Character Format

The eight time intervals have now been increased to ten. There is always a transition from mark to space at the start of the first interval and this is used by the receiver to indicate that it must start sampling at the second interval. After this, the receiver continues to sample until the ninth interval whereupon it resets itself and waits until it sees another transition from mark to space. The first and tenth intervals are called respectively the 'start bit' and 'stop bit'. They do not convey data but serve merely to provide an unambiguous indication that a byte is about to be transmitted. The complete sequence is often referred to as a 10 unit envelope or 10 bit character. Certain mechanical terminals require more than one time interval to reset after each byte. For example the model 33 teletype uses an 11 unit character including two stop bits, and the same 11 unit character is often used by teletype-compatible CRT's.

To ensure correct operation, the time intervals used by transmitting and receiving terminals must obviously be the same. The transmitting and receiving rates, usually expressed in bits per second (bps) are controlled by internal timing devices termed 'clocks', on the two terminals. In the method of transmission just described the receiving clock is restarted for each byte received and there may be a gap of any length of time between bytes. This technique of data transmission is known either as 'asynchronous' or 'start-stop' operation.

Although it is relatively simple to implement, asynchronous transmission is inefficient because ten or eleven bits have to be transmitted to convey each eight data bits. Better use of the channel would result if only the data bits had to be sent, the start and stop bits being eliminated. This can be achieved by the technique known as 'synchronous' transmission. It uses fundamentally the same method as asynchronous but instead of restarting on each byte, the receiver clock is allowed to run continuously and is synchronised with the transmitter clock. Bytes are transmitted continously without gaps, as long as the transmission lasts, and if there are gaps in the data stream the transmitting terminal must insert 'idle bytes' as 'padding.'

Clearly, for correct operation, the receiver must start to sample the line at the correct instant, namely on the first data bit of the first byte. Otherwise it will be out of step and will misinterpret every succeeding byte. The necessary 'byte-synchronisation' is achieved by preceding the data stream by two or more SYNC (synchronisation) bytes having a predetermined bit pattern which can be recognised by the receiving terminal and used to identify the start of the first data byte.

Since synchronous transmission involves block message formats and a set of rules to define the SYNC pattern, synchronous terminals always incorporate a communication protocol. Thus, almost without exception, they perform automatic retransmission on error and can be polled or multidropped on a single line. Most synchronous terminals conform to IBM's BISYNC protocol, or IBM's newer SDLC (Synchronous Data Link Control), but other mainframe manufacturers offer similar protocols.

Despite the efficiency of synchronous transmission, minicomputer systems do not typically support synchronous terminals. They are restricted, for the most part to support of the so-called 'dumb terminal'. But what is a 'dumb terminal'?

1.3 What is a Dumb Terminal?

A dumb terminal is a teletype® or teletype®-compatible terminal. It operates asynchronously, even though it may operate at speeds to 9600 bps, or even higher. It uses no communication protocol or block format: it displays or prints data just as it receives it, without needing to recognize any predefined addressing sequence or check for block errors; it transmits data directly as entered from the terminal keyboard, or from its buffer, without adding any kind of block sequence number or check character. A dumb terminal is a 'protocol-less' terminal. It may incorporate a microcomputer and a very extensive control program which permits it to do very 'intelligent' local screen-formatting, prompting, data field validation and range checking, but such a terminal is still 'dumb' if it is teletype-compatible. Without a protocol, a terminal cannot be addressed, so it cannot be clustered or multidropped with other terminals on the same telephone line, and it has no means of assuring error-free data over real-world telephone circuits. As a data communication device, such a terminal is indeed 'dumb'.

The reason why such terminals are the rule in the minicomputer world and the exception in the mainframe world is twofold:

First, whereas terminals in the batch-oriented world of the mainframe computer are normally remote from the computer site, the interactive, transaction-processing design philosophy of the minicomputer manufacturers means that terminals are the sole means of computer access to local users, as well as remote users. Most of these terminals are hard-wired to the minicomputer, not connected over telephone lines, so they do not need sophisticated retransmission-on-error or line sharing capabilities.

Second, there is no standard communication protocol. The communication protocols used in the mainframe world are the inventions of IBM, Univac, Burroughs, and others. Each protocol was defined by a single manufacturer so that terminals and computer software could be developed to support it. With such a large majority of terminals attached <u>locally</u> to their host minicomputer, minicomputer manufacturers have not felt the need to develop and aggressively market a protocol of their own, except to enhance computer-to-computer communication. Furthermore, with so many of their users satisfied with the present arrangement, and so many of their OEM's and system houses anxious to preserve the freedom of choice which the teletype®-compatible industry standard provides, it appears that the minicomputer manufacturers will not be strongly motivated to invest in a protocol of their own invention. They will wait for an international standard to materialize, perhaps based on the CCITT X.25 protocol defined for access to international packet-switched networks. But such a standard will evolve only very slowly.

In the meantime, the 'dumb terminal' will continue to thrive. But for the increasing number of minicomputer users who cannot tolerate the risk of undetected data transmission errors and the telephone line costs associated with using multiple 'dumb terminals' at a single office remote from the computer, reliable cost-effective data communication may still be achieved using the appropriate external 'black boxes'. These will be described in the Sections which follow.

2. HOW TERMINALS COMMUNICATE WITH A COMPUTER

2.1 The Communication Mode

Communications between two data terminals or between a terminal and a computer can be either Simplex, Half-Duplex (HDX), or Full-Duplex (FDX).

Simplex means that transmission is in one direction only. A typical example would be transmission to a Read-Only (RO) printer.

Half-Duplex means transmission in both directions but only in one direction at a time. In the context of modems, half-duplex operation may be likened to CB radio operation. At the end of the transmission it is necessary to advise the other party that you are through and ready to receive by saying "over". Then the other party can begin transmitting. Some data terminals, especially 'dumb terminals', while only transmitting in one direction at a time, do not have the ability to say "over" (i.e. they do not provide control of the Request-To-Send control signal to the modem). These terminals must therefore be used with full-duplex modems, even though they are half-duplex in operation.

Full-Duplex means transmission in both directions simultaneously. Most terminals do not operate truly full-duplex, except IBM's newer SDLC protocol terminals, although most 'dumb terminals' need to be used with full-duplex modems as stated above. In the context of 'dumb terminals' the term 'full-duplex' normally implies 'echoplex' operation, in which all characters entered from the terminal keyboard are echoed back to the terminal by the computer before being printed or displayed. Echoplex permits verification that data entered from the terminal keyboard was transmitted to the computer without error.

2.2 The Transmission Code

The great majority of data terminals are designed to use one of two international data codes, the USA standard code for information interchange, usually abbreviated to USASCII or simply ASCII, or EBCDIC, short for extended binary coded decimal interchange code.

ASCII is a 7 bit code giving 128 combinations of which 32 are reserved for control functions. In serial data transmission systems the ASCII code is always sent as an 8-bit group, the additional bit being used as a parity check bit, usually with even parity. ASCII is also known internationally as 7-bit ISO code, or CCITT Alphabet No. 5. ASCII is the only code used by 'dumb terminals'. A code chart for the ASCII code is shown in Figure 2-2.

EBCDIC is an 8-bit code giving 256 combinations. It is the standard code for IBM terminals.

In addition to these data codes, message oriented teleprinter terminals, primarily used on the international telex network, utilize a 5-bit code, International Telegraph Alphabet No. 2, more often called Baudot code. A historic relic from the days of telegraphy, Baudot uses a 'shift' code to achieve 64 combinations from 5 bits.

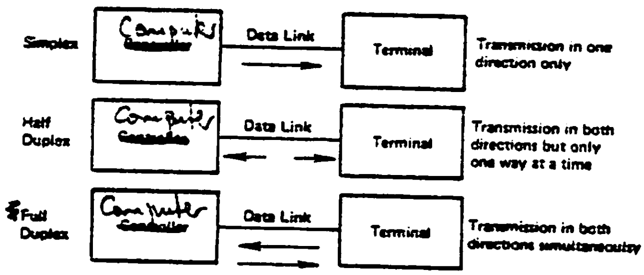


Figure 2-1. Communication Modes

2.3 The Physical Connection

There are several methods of connecting terminals to a computer. The connection methods which are chosen vary depending on the distances involved and the length of time that a terminal needs to be connected. In some applications, such as order-entry, it may be necessary or convenient for the terminal to be connected to the computer throughout the whole day, while others such as time-sharing involve a 'session' lasting only for a hour or so. Applications involving a 'session' may be best suited to a 'dial-up' telephone connection. However, most connection methods are 'dedicated' in nature.

There are seven commonly used methods, all shown in Figure 2-3. Most involve use of the EIA RS-232 or CCITT V.24 interface which is the industry standard terminal interface.

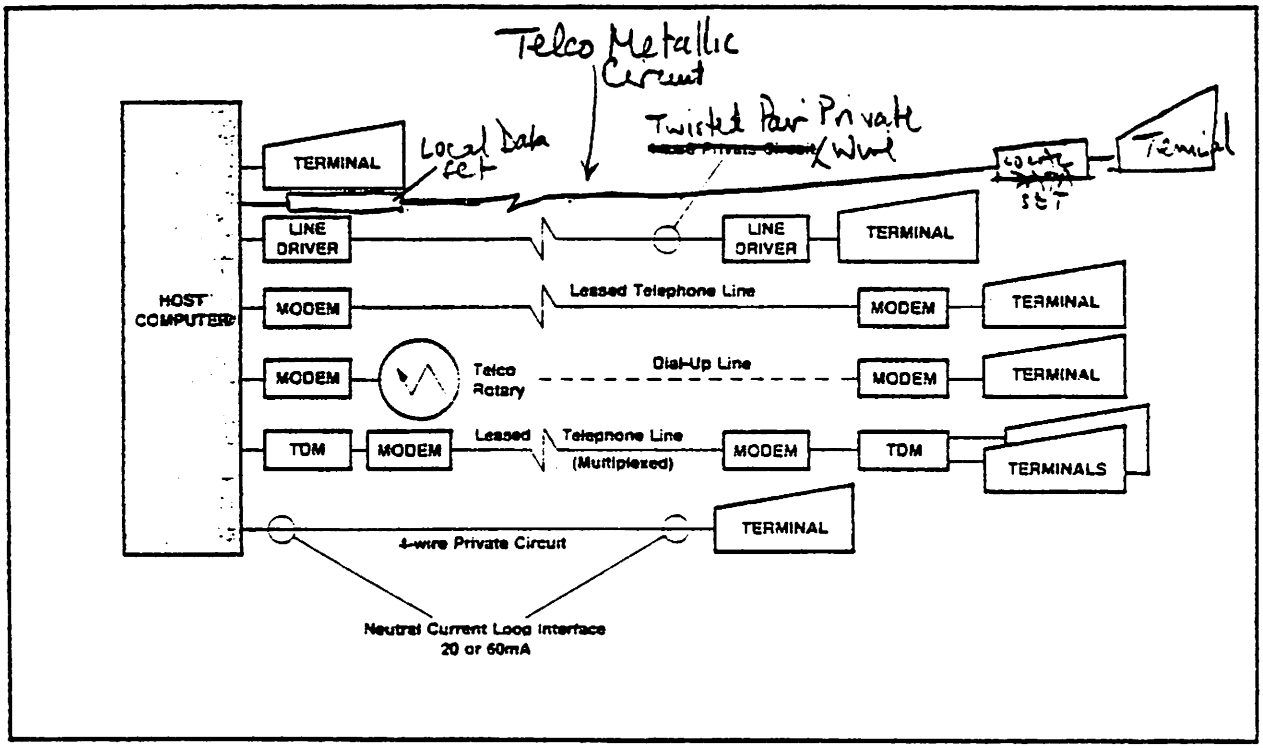


Figure 2-3. Methods of Terminal Connection

Bits b_7 b_6 b_5					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	COLUMN / ROW	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	¦
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Figure 2-2. ASCII Code Chart

The presence of a ring signal on the telephone line causes Ring Indicator to be turned ON by the Modem. If the computer is functioning, it will either turn ON Data Terminal Ready, or it will be programmed to have Data Terminal Ready already ON; in either case, the ON condition of Data Terminal Ready causes the modem to connect itself to the line, equivalent to lifting the handset on a telephone and thereby answering the call. The modem at the same time turns ON Data Set Ready telling the computer that the modem is connected to the line. The computer then turns ON Request-To-Send, switching on the modem transmitter which responds with Clear-To-Send. The terminal user can hear the carrier tone emitted by the modem transmitter and can connect his own modem and terminal to the line by using the pushbutton provided on his telephone or putting the handset in his acoustic coupler. The computer senses the presence of this modem when it sees Received Line Signal Detector turned ON, indicating that the link is now established in both directions.

When the user has finished and hangs up, this action disconnects his modem from the line. As a result, Received Line Signal Detector at the computer end turns OFF, indicating to the computer program that the call is finished. The computer turns OFF Request-To-Send and Data Terminal Ready. The latter action causes the modem to disconnect from the line, equivalent to 'hanging up'. The modem then turns OFF Data Set Ready, confirming to the computer that the disconnection has been made.

Other control circuits permit speed selection on certain modems, provide timing for synchronous modems and terminals, and provide support for the so-called secondary channels which are available on some less commonly used modems.

2.3.1 The EIA RS-232 Interface

The EIA (Electronic Industries Association) standard RS-232 defines the interface between data terminal equipment and data communications equipment. With its international counterpart, CCITT recommendation V.24, it is the only interface standard that is accepted internationally by all computer manufacturers. Using this interface, system designers can be sure at least that their terminals will be compatible with the computer at the hardware level.

The EIA standard does not lay down the method to be used for data transmission across the interface. This is a function of the computer software and varies from system to system. What the RS-232 specification does define is a set of control functions and the way each function or 'interchange circuit' is to be used. A standard pin position on the 25-pin interface connector is given for the circuit controlling each function and the electrical characteristics of the control circuits are precisely defined. A male connector is used on the data terminal equipment; a female connector is used on the data communications equipment. Figure 2-4 shows the interface voltage levels defined by RS-232. Negative (OFF condition) and positive (ON condition) voltages must be in the range five to fifteen volts.

Figure 2-5 shows the interchange circuits by category and their associated pin designations on the interface. The majority of data terminals, however, make use of only a small number of the control circuits. This is particularly true of 'dumb terminals'. Some of them totally ignore the modem's control circuits, but most are designed to check for an ON condition in one or more of the following control circuits before transmitting data: Data Set Ready, Clear-To-Send, or Received Line Signal Detector (sometimes called Carrier Detector). Simarly, most 'dumb terminals' turn ON Request-To-Send and/or Data Terminal Ready before transmitting data. There are no rigorous standards however.

Most of the control circuits are used in support of dial-up operation as follows:

708

Pin Number	EIA Interchange Circuit	C.C.I.T.T. V.24 Equivalent	Description	Gnd	Data		Control		Timing	
					From DCE	To DCE	From DCE	To DCE	From DCE	To DCE
1	AA	101	Protective Ground	X						
7	AB	102	Signal Ground/Common Return	X						
2	BA	103	Transmitted Data			X				
3	BB	104	Received Data		X					
4	CA	105	Request to Send					X		
5	CB	106	Clear to Send				X			
6	CC	107	Data Set Ready				X			
20	CD	108.2	Data Terminal Ready					X		
22	CE	125	Ring Indicator				X			
8	CF	109	Received Line Signal Detector				X			
21	CG	110	Signal Quality Detector				X			
23	CH	111	Data Signal Rate Selector (DTE)					X		
23	CI	112	Data Signal Rate Selector (DCE)				X			
24	DA	113	Transmitter Signal Element Timing (DTE)							X
15	DB	114	Transmitter Signal Element Timing (DCE)						X	
17	DD	115	Receiver Signal Element Timing (DCE)						X	
14	SBA	118	Secondary Transmitted Data			X				
16	SBB	119	Secondary Received Data		X					
19	SCA	120	Secondary Request to Send					X		
13	SCB	121	Secondary Clear to Send				X			
12	SCF	122	Secondary Rec'd Line Signal Detector				X			

Figure 2-5. <u>EIA RS-232 Interface Circuits</u>

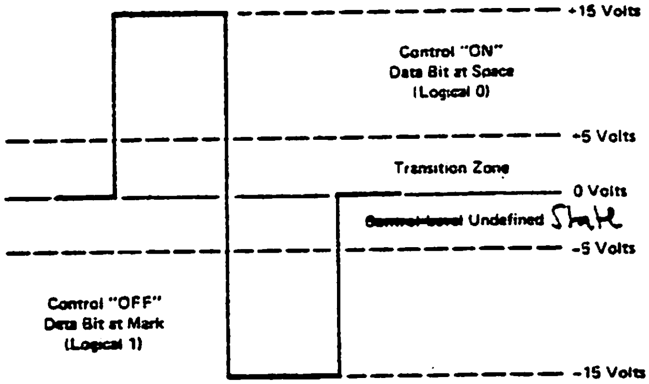


Figure 2-4. <u>EIA RS-232 Interface Voltage Levels</u>

2.3.4 Local Terminals Connected Over Current Loops

Data transmission over distances of several thousand feet within a building may also be accomplished using 4-wire twisted pair and neutral current loops, normally of 20mA. This type of connection requires that both the computer port and the terminal be equipped with current loop interface, not EIA RS-232. Since most computers have the need to connect at least some terminals remotely by means of modems (See 2.3.5 below), which normally use the EIA RS-232 interface, and since it is normally inconvenient to mix EIA and current loop interfaces on the same computer, current loop interfacing is on the decline. An increasing number of computers and terminals do not offer a current loop interface, even as an option.

2.3.5 Remote Terminals Connected Over Telephone Lines Using Modems

A terminal located remotely may be connected to the computer port by means of a leased telephone line with 'modems' at each end. The terminal or computer port is connected by interface cable to the RS-232 interface of the 'modem'.

MO-DEM is a contraction of the words modulator-demodulator and is an electronic device used to convert digital signals to analog form for transmission over the telephone network. Since the telephone network was designed for analog voice transmission, it is not possible to transmit digital information from a terminal or a computer in its binary form. The telephone network has a bandwidth of approximately 3000 Hz as shown in Figure 2-6, so the modems used on the telephone network must condition signals to fit within this band. They achieve this by using the incoming digital data to 'modulate' a sine wave or 'carrier' which may be reliably transmitted through the telephone network and 'demodulated' by the modem at the other end of the line.

Modems use one or a combination of three basic modulation schemes. Figure 2-7 illustrates the three types. A sinusoidal

2.3.2 Connecting Local Terminals to the Computer

Although the EIA interface is intended for transmission of data over telephone lines equipped with modems, it can be used for local attachment of terminals to a computer if the cable run is no more than about 50 feet. In practice, EIA interface cables are normally adequate for distances of several hundred feet, but since the EIA RS-232 specification only requires operation up to 50 feet, no vendor, unless he is supplying the equipment at both ends of the cable and has checked out the operating environment, can guarantee error-free operation over distances greater than 50 feet. Note that if both the terminal and computer port are configured as data terminal equipment (which will normally be the case), it will either be necessary to make the connection by means of a 'crossover' cable, crossing Transmitted Data to Received Data and vice-versa, or it will be necessary to connect the devices together through a 'modem emulator', a simple unit which performs the crossover function and makes each device 'think' it is connected to a modem.

2.3.3 Local Terminals Connected Over Private Wires

Terminals located more than 50 feet from the computer but within the same building or site may be connected to it over twisted pair wire, normally two twisted pairs* using 'Line Drivers', one installed at the terminal location and one installed in front of the computer port. They connect to the terminal or computer port with a regular EIA RS-232 interface cable. A 'Line Driver' is a signal converter which conditions the digital signal transmitted by the EIA interface driver to ensure reliable transmission beyond the 50 feet EIA limit and often up to distances of several miles. It should be noted that if the 'private wires' being used for data transmission within a building or site are owned and maintained by the telephone company, the telephone company may demand the use of Local Datasets conforming to Bell Publication 43401, as described in Section 2.3.6 for operation over telephone company short-haul metallic circuits.

*NOTE: The MICOM Micro400 Asynchronous Line Driver employs a proprietary transmission scheme which permits full-duplex operation at speeds to 19200 bps over 2 wires.

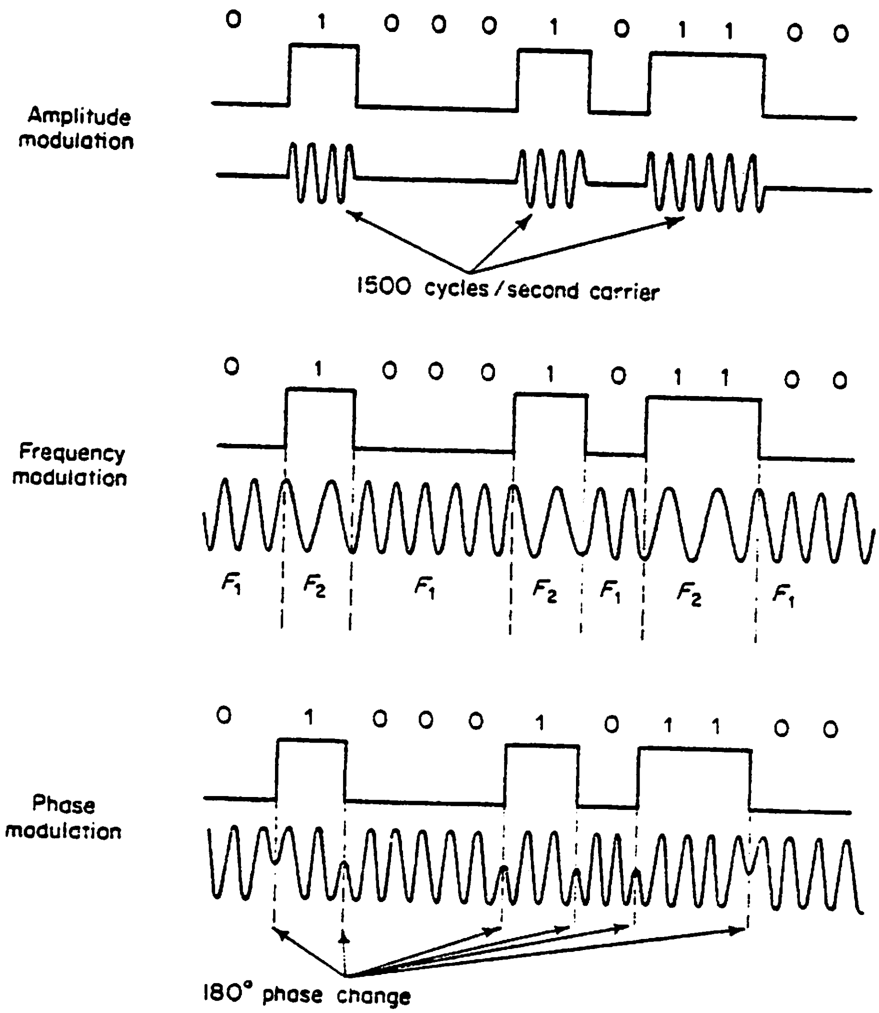


Figure 2-7. Basic Modulation Techniques

carrier wave (sine wave) of, for example, 1500 hertz--in the center of the telephone voice band--is modulated to carry the information bits 01000101100.

In the top diagram of Figure 2-7, the <u>amplitude</u> is varied in accordance with the bit pattern. In the middle diagram, the <u>frequency</u> is varied; and in the bottom one, the phase is varied. In these simplified diagrams, the channel is being operated inefficiently because far more bits could be packed into the carrier oscillations shown. The tightness of this packing determines the speed of operation. Many variations are possible within these three main types of modulation.

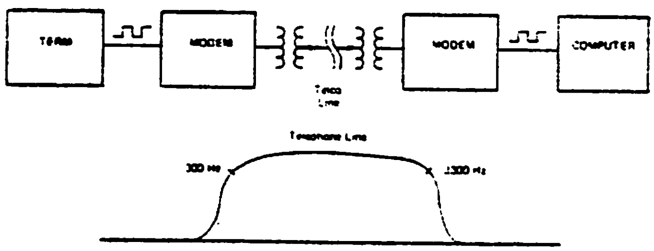
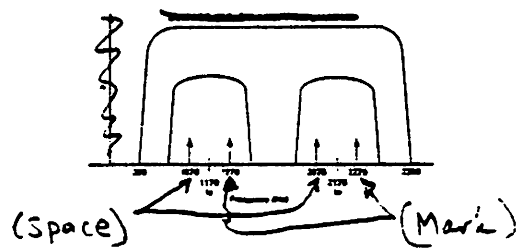


Figure 2-6. <u>Why Modems?</u>

BELL 103/113



BELL 202

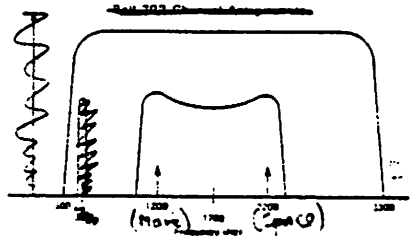


Figure 2-8. How FSK Modems Work

2.3.5.1 Synchronous and Asynchronous Modems Compared

Modems fall into two categories, asynchronous and synchronous.

Most asynchronous modems use the simple Frequency Modulation, sometimes called FSK (Frequency Shift Keyed) Modulation. They transmit a signal at one frequency to indicate a space condition on the modem's Transmitted Data interface circuit and a different frequency to indicate a mark. They operate asynchronously in the sense that the appropriate frequency is transmitted for exactly the length of time that the mark or space condition is present. No special timing constraints are imposed by the design of the modem.

The most popular FSK modems are the Bell 103/113 and the Bell 202 types. The international equivalent, though not directly compatible, are known as CCITT V.21 and CCITT V.23 respectively. Although the Bell System nomenclature is always used in the USA to define these two types of modems, many independent modem manufacturers offer compatible products. The 103/113 operates up to 300 bps full-duplex on two-wire lines, dial-up or leased. The 202 operates up to 1200 bps, or optionally up to 1800 bps, full-duplex on four-wire leased lines and half-duplex on two-wire lines, dial-up or leased.

In the context of asynchronous modems, speed is sometimes referred to in terms of 'baud' (for example, the Bell 103 is a 300 baud modem). One 'baud' is one signal element per second. An FSK modem transmits one bit per baud.

The disadvantage of asynchronous FSK modems is that they are limited in speed. This is because greater separation must be allowed between the mark and space frequencies for higher data transmission rates, and the 3000 Hz telephone line bandwidth limits the amount of separation which can be achieved. The practical upper limit for data transmission with this type of modem is 1800 baud, but the Bell System requires C2 line conditioning (lines specially 'conditioned' for data transmission) for operation at this speed with the Bell 202. The signalling frequencies used by the Bell 103/113 and Bell 202 are shown in Figure 2-8.

Independent modem manufactuers are extremely strong in the 4800 and 9600 bps modem field, typically offering products which are not compatible with the Bell 208 and 209. But independent manufacturers' 2400 bps modems are all compatible with the Bell 201. The international equivalents are CCITT V.26 at 2400 bps, V.27 at 4800 bps, and V.29 at 9600 bps. V.27 and V.29 are totally incompatible with the Bell 208 and 209.

Dibit	Phase Shift (Bell 201B/C)
00	45°
01	135°
11	225°
10	315°

Figure 2-9. How DPSK Modems Work

Modems which operate at speeds higher than 1800 bps beat the bandwidth limitation by encoding several data bits in a baud. The Bell 201, for example, uses Phase Modulation (PM) and encodes two bits (called a 'dibit') in one baud. This type of modulation is more properly referred to as Dibit Phase Shift Keying (DPSK). As shown in Figure 2-9, it shifts the phase angle of the sine wave carrier one of four ways to indicate the four 'dibit' combinations. Thus, though the modem operates at 1200 baud (1200 signal elements per second) it transmits data at 2400 bps. Modems designed to transmit data at 4800 and 9600 bps employ considerably more sophisticated modulation and encoding schemes, and their complexity produces a much more costly product.

Since these modems encode several bits in one signal element, they must receive bits from the data terminal equipment at predefined time intervals, and conversely, they must know at what time intervals to present the decoded bits to the terminal. This necessitates strict synchronization between the modem and the data terminal and a totally synchronous modem design. Synchronization of modem and terminal is normally achieved by allowing the modem to 'clock' the terminal. The modem provides on pin 17 a clock pulse with every data bit it presents to the terminal, and generates clock pulses on pin 15 to control the output of data bits from the terminal to the modem (please refer to Figure 2-5 for a full listing of all EIA RS-232 Interface Circuits). Thus terminals used with synchronous modems must be able to accept a clock on pin 15 to control their data transmission.

The Bell System offers synchronous modems for operation at 2400, 4800 and 9600 bps. All operate full-duplex on four-wire leased lines and half-duplex on two-wire lines. The 201B, 208A and 209A are for leased line operation at 2400, 4800 and 9600 bps respectively; the 201C and 208B operate half-duplex dial-up at 2400 and 4800 bps respectively.

tariff includes the cost of the Data Service Unit (DSU), which is the short-haul modem used by the Bell System on the local loop to the DDS network. With present tariffs, DDS offer few cost savings to the user. Users benefit instead from improved performance and reliability because of the nature of digital transmission and the design of the digital network.

2.3.6 Remote Terminals Connected Over Telephone Company Short-Haul Metallic Circuits

A terminal located remotely, but within a short enough distance to obtain DC-continuous metallic circuits from the telephone company, may be connected to the computer port by means of leased telephone wires with 'Local Datasets', sometimes referred to as 'Short-Haul Modems' or 'Baseband Modems', installed at each end. These private lines are offered under the 3081 copper wire tariff by Bell operating companies in more than 20 states. However, there is sometimes a one-time fee of as much as $300 per line to cover the cost of removing the loading coils and bridge taps required for normal voice usage. The Local Datasets used on the lines must conform to Bell Publication 43401. The terminal or computer port is connected by interface cable to the RS-232 interface of the Local Dataset.

Since the DC-continuous metallic circuits used on telephone company local loops do not have the limited bandwidth characteristic of the long-distance telephone system, a Local Dataset is more like the 'Line Driver' described in Section 2.3.3 than the modems required for long haul transmission. The Local Dataset uses baseband signalling. It does not modulate a carrier but simply conditions the square-edged pulses that emerge from the terminal suitable for transmission over distances of ten or twenty miles at the most. Unlike the simpler Line Driver, however, the Local Dataset must include some filtering to ensure that the transmitted telephone line signal does not interfere with adjacent pairs in the same telephone cable.

As Figure 2-10 shows, high-speed data transmission is relatively expensive. The complexity of the higher speed modems is fully reflected in their price.

Bell Model	Speed (bps)	Type	Typical Cost	Typical Purchase
103/113	300	ASYNC	$25 to $35	$400
202	1200	ASYNC	$35 to $50	$500
201	2400	SYNC	$55	$750
208	4800	SYNC	$125	$3,000
209	9600	SYNC	$230	$5,000

Figure 2-10. Leased Line Modem Summary Chart

Full-duplex operation with synchronous modems requires a 4-wire private line from the phone company, sometimes called a 3002 circuit. These voice-grade leased lines are available under AT&T's Multi-Schedule Private Line (MPL) service. These lines need to be specially tuned or 'conditioned' for use with some high-speed modems, especially those supplied by the phone company. Most independent modem manufacturers, however, now offer modems at speeds to 9600 bps which run on 'unconditioned' telephone lines, resulting in substantial monthly cost savings.

In many parts of the U.S.A. a new Dataphone Digital Service (DDS) is now available from the phone company which provides full-duplex synchronous data transmission at speeds of 2400, 4800, and 9600 bps. Modems are not required on DDS circuits, but the

When these higher-speed modems are used on the dial network, they must be used half-duplex. Transmission can occur only in one direction at once. The terminal must turn on Request-To-Send when it wishes to transmit, and wait for Clear-To-Send from the modem before transmitting. The terminal must drop Request-To-Send when its message has been transmitted to allow the computer to respond and the modems to 'turn the line around'.

Unfortunately, most 'dumb terminals' are not designed for message transmission. They operate 'character-by-character'. They are not able to control Request-To-Send. Thus, although the terminals themselves may be used in a half-duplex mode, they cannot be used with half-duplex modems.

Fortunately, special-purpose full-duplex dial network modems are now available for operation at 1200 bps, so the 'dumb terminal' user is not restricted to the 103's 300 bps. These modems, pioneered by Racal-Vadic with its Model 3400, include the Bell 212A. The international equivalent is CCITT V.22. However, none of these modems is compatible. All of them use synchronous modulation techniques, encoding the data into a narrow bandwidth line signal, narrow enough to fit both transmit and receive paths into a two-wire connection. They are relatively expensive, however, costing the equivalent of a 2400 bps modem.

The requirement for 'dumb terminals' to operate with full-duplex modems means that they are effectively limited to a transmission speed of 1200 bps on the dial network.

2.3.8 Terminal Groups in Remote Sites

Terminals located in groups at remote sites may be multiplexed to the computer site using a single 4-wire telephone line with synchronous modems and time-division multiplexors (TDM) or concentrators such as the MICOM Micro800 Data Concentrator. Connection of terminals to the remote concentrator may be by EIA RS-232 interface cable, private wires and line drivers, or telephone lines with local datasets or modems depending upon distance from the concentrator. The 'multiplexed' connection is explored in detail in Section 3.

Local Datasets have two major advantages compared with modems. First, they are substantially cheaper (sometimes as cheap as one month's rental of a long-haul modem) and they offer data transmission up to 9600 bps for the same price as 2400 bps. They are therefore particularly cost-effective at high-speed. Second, they can operate asynchronously, even up to 9600 bps. Therefore, they are ideally suited for use with 'dumb terminals'. Unfortunately, however, as already noted, the appropriate telephone circuits for this type of device are not always available. Even where the local phone company does offer metallic circuits, such circuits are generally not provided when the terminating points are served by different telephone company central offices. Two points which are physically close together may be served by different central offices so even physical proximity is not an indication that metallic circuits will be available.

2.3.7 Remote Terminals Connected Over Dial-Up Lines

Where a remote terminal requires connection to the computer only for a short time each day, the connection may be established via the dial-up telephone network using modems or 'acoustic couplers'. This is typically more cost-effective than leasing a telephone line at a fixed monthly charge, unless terminal utilization is heavy. It also allows many terminals to be supported by a computer with a limited number of computer ports, providing 'contention' on a first-come-first served basis.

2.3.7.1 Full-Duplex and Half-Duplex Modems Explained

A dial-up connection is a two-wire connection. A Bell 103, operating with FSK modulation up to 300 bps, requires so little of the available bandwidth that it can split the band into two halves and provide transmit and receive paths on the same telephone pair. It is thus able to operate full-duplex on a two-wire line. But the Bell 202, 201, 208 and 209, operating at speeds from 1200 to 9600 bps, need 4-wire lines to operate full-duplex, with one pair used for transmit and one for receive.

3. REDUCING LINE COSTS

Any computer system supporting multiple remote terminals does not need to be very large before the cost of the telephone lines and modems exceeds the cost of the computer and terminal equipment. Even in smaller systems, some attempt must be made to reduce the cost of the telephone lines by putting more than one terminal on each line.

Figure 3-1 shows the effect on a large-scale nationwide data communication system of installing a network of multiplexors and concentrators. In this example, the system had grown, over the years, in an unplanned manner, responding to crises, using point-to-point telephone lines to service new locations as the demands of the organization dictated. Before long, it was using more than 100,000 miles of leased telephone lines. With the intelligent use of multiplexors and concentrators, the line cost savings achieved were in excess of $50,000 per month.

This real-world example of what can be achieved is outrageous and exaggerated in many respects, but every minicomputer system supporting multiple remote terminals has the same potential to achieve major cost savings, albeit on a smaller scale, by applying the same techniques.

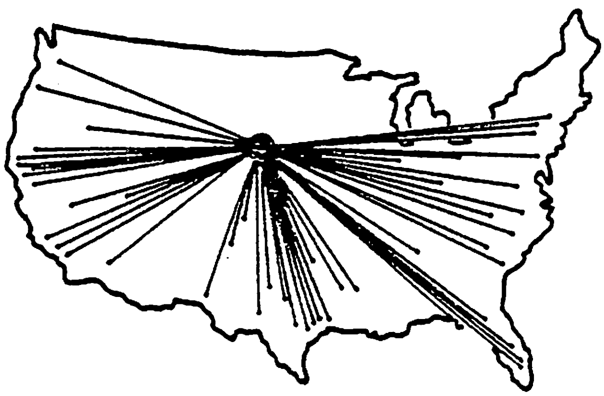
As discussed in Section 1, terminals supporting one of the established communication protocols can be multidropped and polled on a shared telephone line. But the 'protocol-less' terminals used with minicomputers cannot be multidropped. The only way several 'dumb terminals' can share one telephone line is by multiplexing.

3.1 What is a Multiplexor?

A multiplexor system provide a 'transparent' connection between modems and/or terminals located remotely from the computer and the computer ports to which these modems or terminals are normally directly attached.

As shown in Figure 3-2, a multiplexor system allows eight terminals in Denver to share a single telephone circuit to the computer center in Los Angeles where previously eight lines were required. It should be noted that multiplexors are normally required at both ends of the 'shared' telephone line, so that the eight channels multiplexed at one end may be demultiplexed back into eight channels at the other, thus providing the transparent connection between each of the eight computer ports and each of the eight terminals.

More specifically, the word 'transparent' means that the multiplexor system does not in any way interrupt the flow of data and EIA RS-232 interface signals which are normally passed across the cable between the computer port and the attached low-speed modems or terminal. Thus, as shown in the more detailed Figure 3-3, when the multiplexors are installed, neither the computer, nor the modems moved out to Denver, nor the terminals using the modems, know that the multiplexor system is being used, whether the telephone lines attaching the terminals to the multiplexor are leased or dial-up. As a result, neither terminal equipment nor computer software need be changed when a multiplexor system is installed to reduce telephone line rental or long-distance dial charges.


Nationwide network implemented without multiplexing.

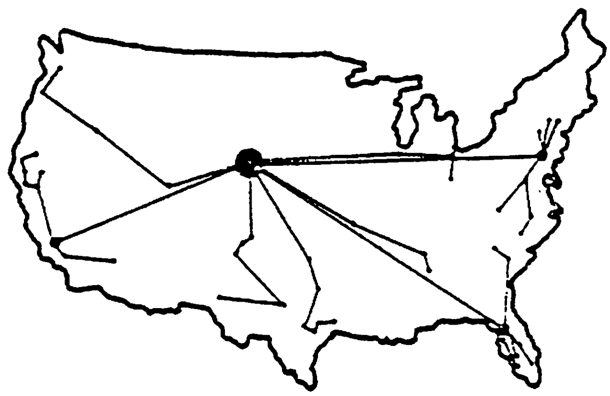

Line utilization improvement using multipoint multiplexor/concentrator system.

Figure 3-1. Multiplexors/Concentrators Reduce Line Costs

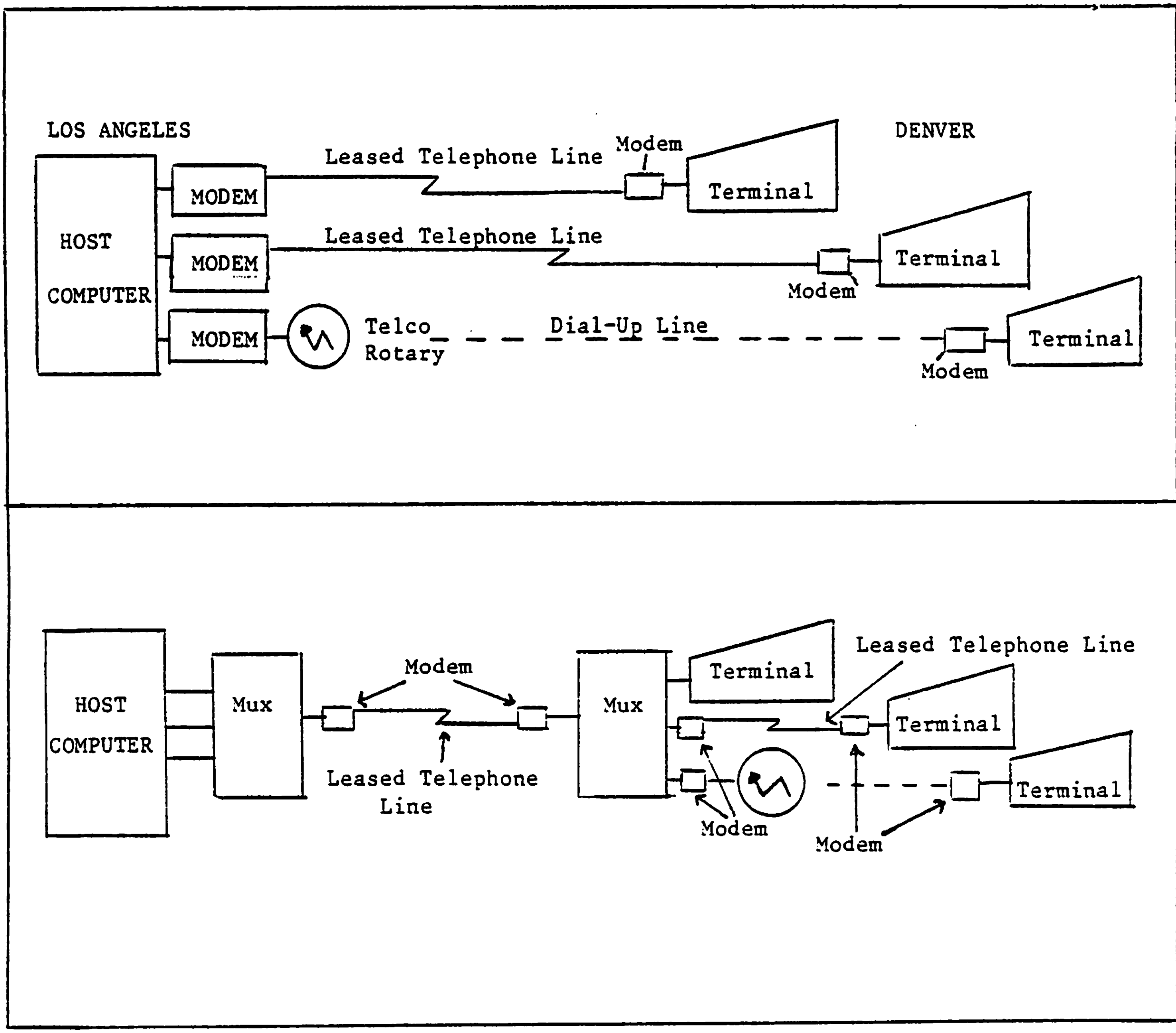


Figure 3-3. Multiplexor Transparency

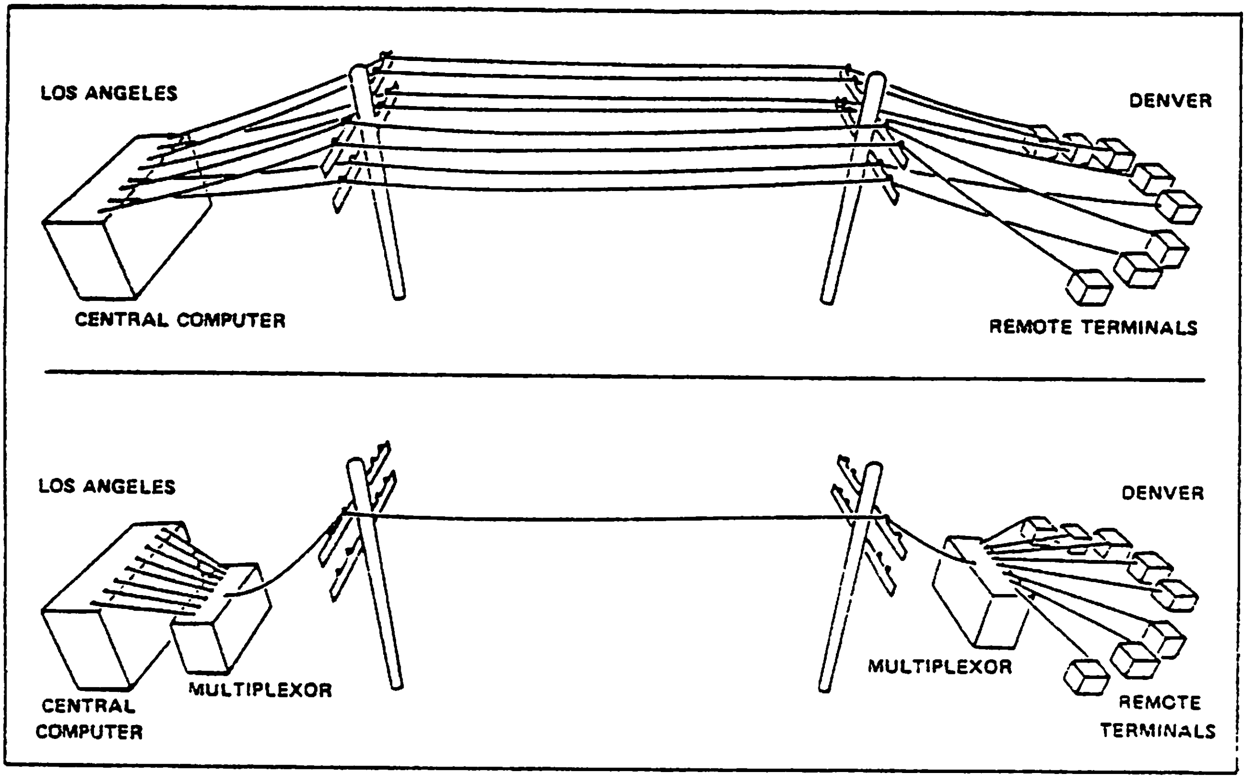


Figure 3-2. Typical Multiplexor Application

Comparison Factor	FDM	TDM	Stat Mux
Efficiency	Poor	Good	Excellent
Channel Capacity	Poor	Good	Excellent
High-Speed Channels	Very Poor	Poor	Excellent
Flexibility	Poor	Good	Excellent
Ease of Installation	Poor	Poor	Excellent
Retransmission-on-Error	N/A	N/A	Automatic
Multidrop Capability	Good	N/A	Possible

Figure 3-4. Multiplexing Techniques Compared

3.2 Multiplexing Techniques Compared

The three basic multiplexing techniques are:

- Frequency-division multiplexing (FDM)
- Time-division multiplexing (TDM)
- Statistical multiplexing

Frequency-division multiplexing is the oldest type of multiplexing. In Section 2.3.5.1 we discussed how a Bell 103 modem, using FSK modulation, splits the telephone line into two subchannels to provide a transmit and receive path simultaneously on the same telephone line pair. A frequency-division multiplexor is a group of FSK modems all sharing a single telephone line, with each operating on a part of the telephone line bandwidth separated from the next 'channel' by a 'guard-band' or unused portion of the bandwidth. The FDM takes advantage of the fact that the 3000 Hz bandwidth of a voice-grade telephone line is significantly greater than that required by a low-speed data channel, and by using a different frequency band for each channel it allows several to share the same line without interfering with each other.

Time-division multiplexing (TDM) is a digital technique. It interleaves bits (bit TDM) or characters (character TDM), one from each attached channel and transmits them at high speed down a telephone line equipped with high-speed synchronous modems. At the other end, the other multiplexor demultiplexes the 'bit train' or 'character frame' presenting one bit or character to each low-speed channel just as they originated.

Statistical multiplexing, sometimes referred to as intelligent time-division multiplexing (ITDM) or data concentration, is a relatively new technique. Like TDM it is a digital technique, but it exploits microcomputer technology to provide greater efficiency and automatic retransmission-on-error.

These three multiplexing techniques are compared against seven general criteria in Figure 3-4.

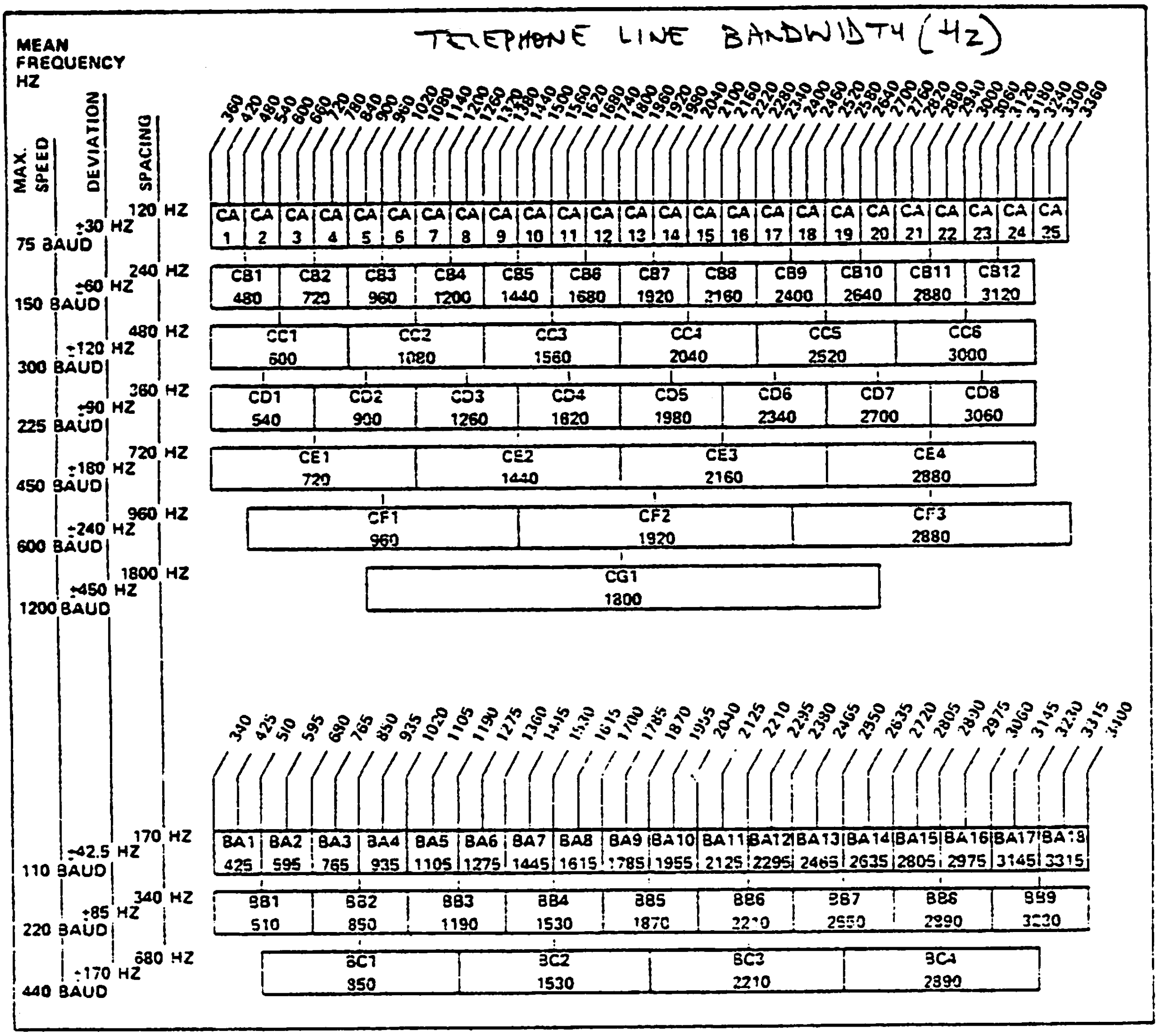


Figure 3-5. FDM Frequency Allocation Chart

3.2.1 Frequency Division Multiplexing (FDM)

Using Figure 3-4's comparison criteria, frequency-division multiplexing does not score very highly because it is not a digital technique. It is inefficient, because the channels have to be separated across the band to prevent cross-talk and this, of course, wastes bandwidth. Referring to the Frequency Allocation Chart, Figure 3-5, it will be seen, for example, that a 300 baud channel requires 480 Hz bandwidth to provide the recommended channel spacing. Channel capacity is physically limited by the telephone channel's 3000 Hz bandwidth resulting in a capacity of only six channels at 300 bps. And since the FDM is not even capable of multiplexing two channels at 1200 bps, it scores very badly in its ability to handle high-speed channels.

FDM is also inflexible because a change in channel speed or number of channels may demand that the center frequencies of all other channels must be redefined. The need to configure each channel for the correct data rate and the correct separation from the next channel also complicates installation.

The FDM has no ability to perform automatic retransmission-on-error. One strong point of the FDM, however, is that since each channel uses a different frequency within the bandwidth, individual channels can be dropped and inserted at different points along the same telephone line. This permits multiplexing one or two terminals in one city or location and extending the line to the next city or location to pick up one or two more terminals. There can be as many drops or points along the line as there are channels within the capacity of the FDM system (see Figure 3-6). Thus, the FDM provides a unique multidrop capability for simple asynchronous terminals without imposing any requirement that they be 'addressable' or use any type of communication protocol.

The character buffering does introduce a delay through the system of 1.5 ± 0.5 character times. However, in most applications this is insignificant, particularly with higher speed terminals. The delay is only noticeable with printing terminals operating full-duplex (echoplex) at speeds of 300 bps and below.

The simplicity of the TDM technique results from the fact that the multiplexors communicate with each other by transmitting a constant stream of bits (bit TDM) or characters (character TDM) with a regularly recurring SYNC character and a constant number of 'time-slots' between SYNC characters, each 'time-slot' containing a predefined number of bits or characters for a specific channel (see Figure 3-7). It is a fundamental principle of time-division multiplexing that the 'frame' (the data block following each SYNC character) is fixed in length, and that frames are transmitted continuously and contiguously, since the receiving multiplexor only knows which bits or characters belong to each channel as a function of their time relationship to the SYNC character. The number of bits or characters in each 'time-slot' allows it to handle its assigned channel without losing data even when the channel is operating at maximum speed. But since the number of bits or characters is fixed, the TDM must transmit them even if the channel is operating below its maximum speed or not operating at all. And since transmission is continuous, there is no possibility of retransmission even if the receiving multiplexor does detect errors.

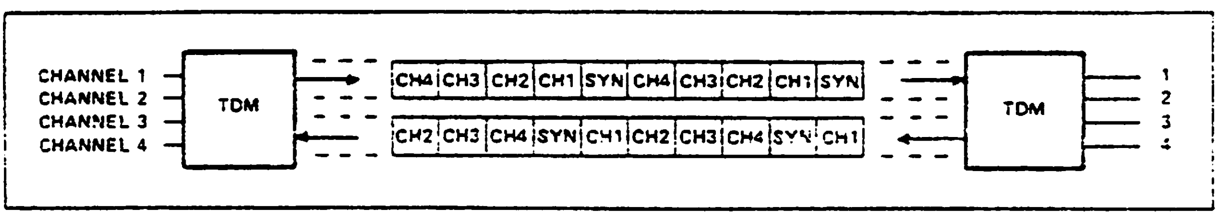


Figure 3-7. TDM Transmission Protocol

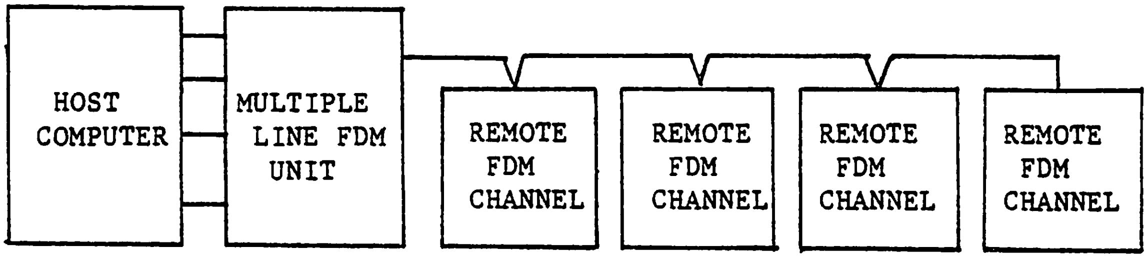


Figure 3-6. FDM Multidrop Capability

3.2.2 Time-Division Multiplexing (TDM)

Since TDM is a digital system, it can be used with the more complex synchronous modems capable of transmitting data at speeds up to 9600 bps on voice-grade telephone lines. TDM's may also be used at much higher speeds on the newer DDS (Dataphone Digital Service) circuits or satellite channels, operating at speeds of 56,000 bps or higher. Hence, channel capability is much greater than with FDM. TDM multiplexors also allow very considerable flexibility to change the number and/or the speed of channels as the requirements of the data network change. This too is possible because the TDM is a digital system independent of the data modem attaching it to the high-speed line. Unfortunately, however, most TDM's incorporate literally thousands of strap option permutations, making installation by a qualified technician mandatory.

The character TDM offers greater efficiency than the FDM because it buffers a complete character before transmitting it down the high-speed line. As a result, it is possible to remove the start and stop bits from each asynchronous character before transmission, adding the start and stop bits during demultiplexing at the other end. Thus, in the case of a teletype, it is only necessary to transmit eight bits for every eleven bits (one start and two stop) received.

3.2.3 The Statistical Multiplexor or Data Concentrator

Since it is very rare for any terminal to transmit continuously at its maximum operating speed, any multiplexor design which is intelligent enough to take advantage of this fact will achieve much greater terminal capacity on a given phone line. The advent of the microcomputer made possible the low-cost implementation of a new generation of intelligent TDM or Statistical Multiplexor, designed to handle efficiently the statistically determined data traffic, with the ability to temporarily store data in its memory during periods of peak traffic activity.

MICOM calls this type of device a Data Concentrator or Computer Concentrator, since the new jargon phrase 'Statistical Multiplexor' implies a new multiplexing concept. In fact, this type of product has been in service for years in the largest data communication networks, with minicomputers being used for implementation. The Computer Concentrator is programmed to buffer data from individual terminals, to transmit variable length data blocks according to the loading on individual channels, and to check data blocks received on the high-speed line and request retransmission in the event of errors. Without a computer, and computer control of a large memory, such functions are exceedingly complex to implement. But with a computer, the functions can be readily programmed. The new Statistical Multiplexors or Data Concentrators represent no new concept, but a breakthrough in low-cost <u>implementation</u> made possible by the advent of the microcomputer.

Figure 3-8 shows a typical concentrator protocol. Each block starts with a Start of Header character marking the beginning of the control information relating to the block contents. The control information will typically contain the sequence number of the block, and also the block sequence number of the last block received correctly in the other direction; in the event the last block was received in error, a NAK (non-acknowledgement) flag will replace the sequence number of the last block received; the block header also includes

The 'idle' character used to fill a 'time-slot' when no data character is ready for transmission is used to transmit channel interface condition information, the 'break' condition, remote loopback commands, diagnostic status information, and terminal speed and code information needed to support adaptive speed multiplexor configurations where the multiplexor channel speed is established by analysis of the first character received from a dial-up terminal. Thus, even the 'idle' characters are not totally redundant, but the frequency of change in the control information is normally sufficiently low that a large element of redundant information is transmitted.

For simplicity, Figure 3-7 shows a TDM configuration with fixed sequential scanning for interleaving bytes into the high-speed serial stream. Fixed scanning samples a bit or character from each channel in order; channel 1 then 2, 3, etc. This results in all channels occupying the same amount of bandwidth on the high-speed line. Thus, if the highest speed terminal attached to the multiplexor is 300 bps, all channel allocations on the high-speed line will be at 300 bps regardless of the terminal speed (which may be lower, such as 110, 134.5 or 150 bps). Most TDM's on the market today use variable channel address scanning.

A variable scan multiplexor has the capability to make three samples for each 300 bps line compared to one for 110 bps or two for 134.5 bps terminals. This permits much more efficient use of high speed capacity. But, with or without variable channel scanning, each channel's time-slot is fixed and capable of supporting the channel operating at maximum speed continuously.

In practice, it is very rare for any terminal to transmit continuously at its maximum operating speed. This is particularly true of the interactive 'dumb terminals' used with minicomputer systems which typically transmit only a line or perhaps a screen-full of data before pausing.

For the minicomputer user, this type of device represents a major breakthrough, allowing efficient support of terminal clusters without the need to poll terminals. Compared with the TDM possibilities, the Data Concentrator also permits much faster terminal operation without needing very high-speed modems and without suffering line errors. Typical prices range from $1200 for a 2-channel unit to $1700 for a 4-channel and $2700 for a 8-channel unit.

Another major advantage of the Data Concentrator is that it can be designed for use in multidrop configurations. As shown in Figure 3-9, a central site 'Master' Concentrator can be programmed to poll multiple remote 'Node' Concentrators, with the polling performed automatically and quite transparently to the terminals and the host computer. In this configuration, the 'Multidrop Concentrator' serves the same function as the FDM, while permitting operation with much higher speed 'dumb terminals'.

The point-to-point Data Concentrator permits support of terminal clusters without the requirement for polling protocol in the computer and terminals; the multipoint Multidrop Concentrator permits support of multidropped terminals and/or terminal clusters, also without any requirement for polling protocol. Both products represent 'black boxes' which provide 'add-on protocol' for the dumb terminals used with minicomputers.

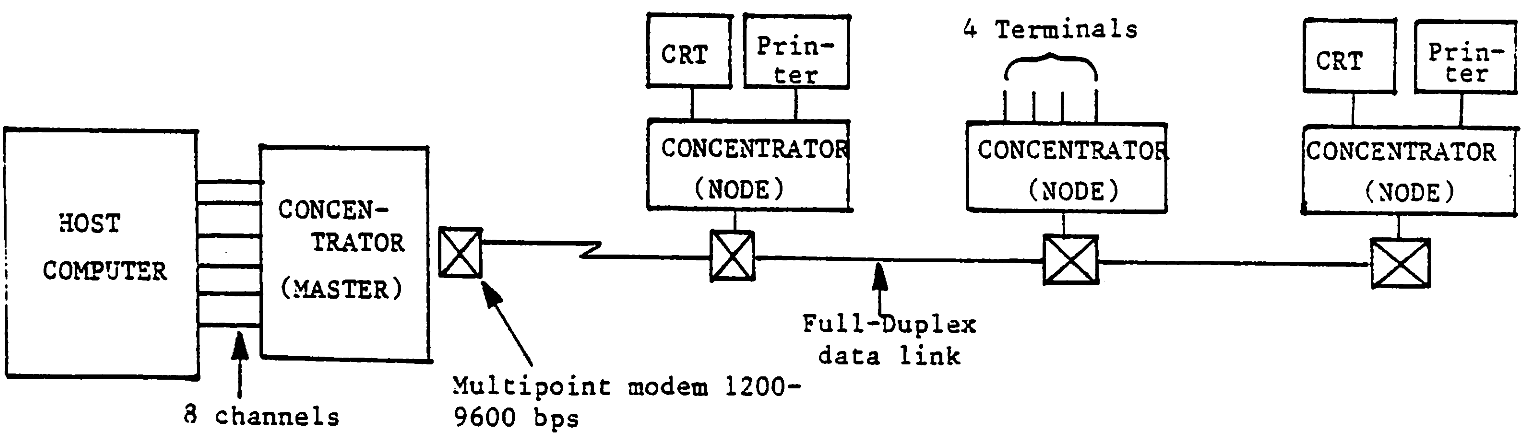


Figure 3-9. Multidrop Concentrator Configuration

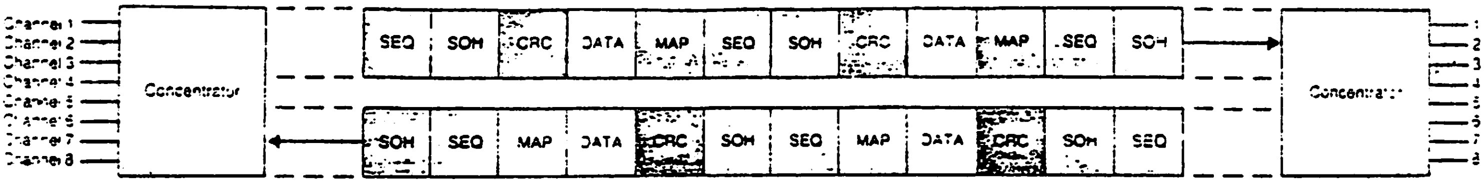


Figure 3-8. Typical Concentrator Line Protocol

'mapping' information to indicate the channels present in the block and the number of data characters for each channel. Following the control information is the data portion of the block which is variable in length, depending on the number of active channels and the activity rate of each channel. Finally, a Cyclic Redundancy Check (CRC) character terminates the block; the CRC is recalculated at the receiving end to ensure that the data block was received correctly. The CRC is the 16-bit result of a polynomial calculation performed on the bits in the block. It provides only a 1 in 10^{12} probability that a CRC will check out correctly even with a block in error. Thus, it reduces, almost to zero, the possibility that multiple bit errors will be self-cancelling and avoid detection.

At first sight it would appear that the concentrator protocol has considerably more overhead (non-data) characters than the TDM protocol. But the characters in the data portion of the concentrator protocol are real data from active channels, whereas many TDM time-slots are actually filled with pad characters or idles. It should be noted, however, that if every channel was operating continuously at its maximum speed, and hence all the TDM time-slots were filled with real data, then the TDM protocol would be more efficient. But this point is only of academic interest.

The transmission of variable length data blocks according to the loading on individual channels allows the Data Concentrator to waste no high-speed line time with inactive channels. It needs a high-speed line capable of dealing only with <u>average</u> traffic loading since it can make use of its buffer storage to accommodate temporary <u>peak</u> loading on individual channels or groups of channels. The 'concentration' achieved by the Concentrator's transmission scheme and 'backup' buffer storage means that its channel capacity and efficiency are superior to TDM. If the total throughput of the channels is significantly below their maximum data rate, the greater is the superiority of the Data Concentrator relative to the TDM.

4. ADD-ON DATA LINK CONTROL (ADLC™)

We have already discussed the significance of communication protocol. Several new protocols have been introduced recently in an attempt to establish a generally acceptable standard: IBM has developed Synchronous Data Link Control (SDLC); an international standard now exists known as the High-Level Data Link Control (HDLC); and there is now a Federal Standard called Advanced Data Communication Control Procedures (ADCCP). But all these protocols are inapplicable to the 'dumb terminal' user, and hence to most minicomputer users. The benefits which they provide, however, may be obtained by using external, standalone 'black boxes' to provide Add-On Data Link Control (ADLC ™).

The ADLC™ concept was pioneered by MICOM SYSTEMS, INC. with a range of products which provide add-on protocol to minicomputer-based 'dumb terminal' configurations without requiring any changes to existing hardware and software.

4.1 The Error Controller

The most basic element of the ADLC™ family is the Error Controller.

Retransmission-on-error is particularly required at higher data transmission speeds. Although error rates with 4800 and 9600 bps modems on leased telephone lines are often as good as 1 bit in 10^6, the telephone company will provide assurances of no better than about 1 bit in 100,000. At 4800 bps this is equivalent to one error every 20 seconds. In reality, phone line errors occur more often as random bursts of noise, rather than very frequent single bit errors, but data transmission at 2400 bps and above without retransmission-on-error is risky at best.

In addition, as has already been noted, data transmission at 2400 bps and above requires synchronous modems, and 'dumb terminals' are asynchronous.

The Error Controller is a low-cost 'black box' designed to provide automatic retransmission-on-error as well as the mode conversion necessary to allow an asynchronous terminal to operate with synchronous modems at 2400 bps and above. The Error Controller is installed in pairs, one at each end of the leased telephone line, as shown in Figure 4-1.

The Error Controller is also sometimes referred to as an intelligent async-to-sync converter. Special dial-up versions are available which permit the full-duplex, asynchronous 'dumb terminal' to be used error-free with synchronous dial-up modems such as the Bell 201C and 208B which operate half-duplex at 2400 or 4800 bps. These dial-up models must also serve as half-duplex-to-full-duplex protocol converters. A typical configuration is shown in Figure 4-2.

In operation, the Error Controller receives data character-by-character from the terminal or computer port and transmits it in a block format, with block length determined by the number of characters received since the last block was transmitted. Blocks are transmitted continuously to minimize transmission delay. Each block starts with a 'Header' containing control information. This consists of the sequence number of the block, and also the block sequence number of the last block received correctly in the other direction. In the event the last block was received in error, a NAK (non-acknowledgement) block will be sent. The block Header also includes a count to indicate the number of data characters in the block. Finally, a Cyclic Redundancy Check (CRC) character terminates the block. The CRC is recalculated at the receiving end to ensure that the data block was received correctly. The CRC is the 16-bit result of a polynomial calculation performed on the bits in the block. It provides only a 1 in 10^{12} probability that a CRC will check out correctly even with a block in error. Thus, it reduces, almost to zero, the possibility that multiple bit errors will be 'self-cancelling' and avoid detection. Any block received in error is automatically retransmitted. This is the same error control technique used in SDLC, HDLC, or ADCCP.

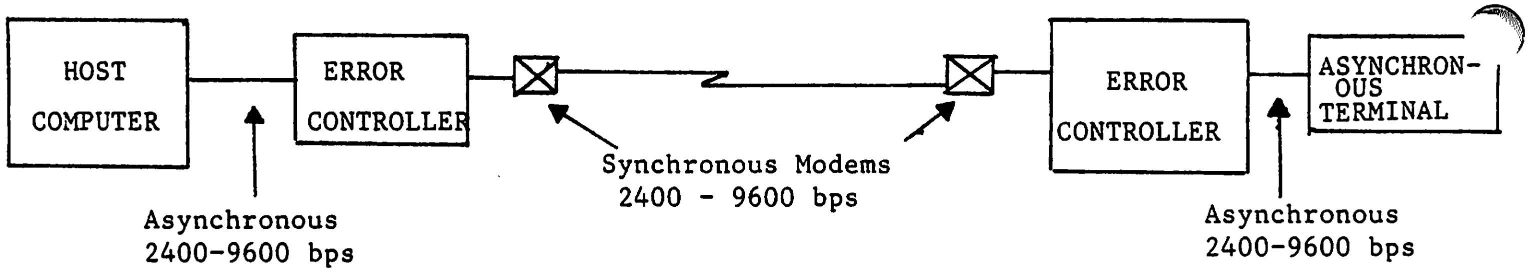


Figure 4-1. Typical Error Controller Configuration

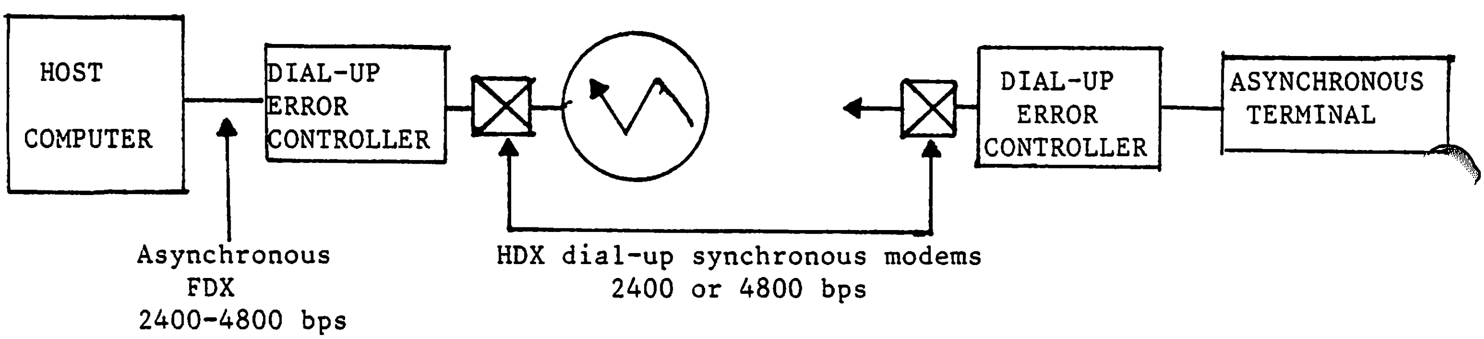


Figure 4-2. Typical Dial-Up Error Controller Configuration

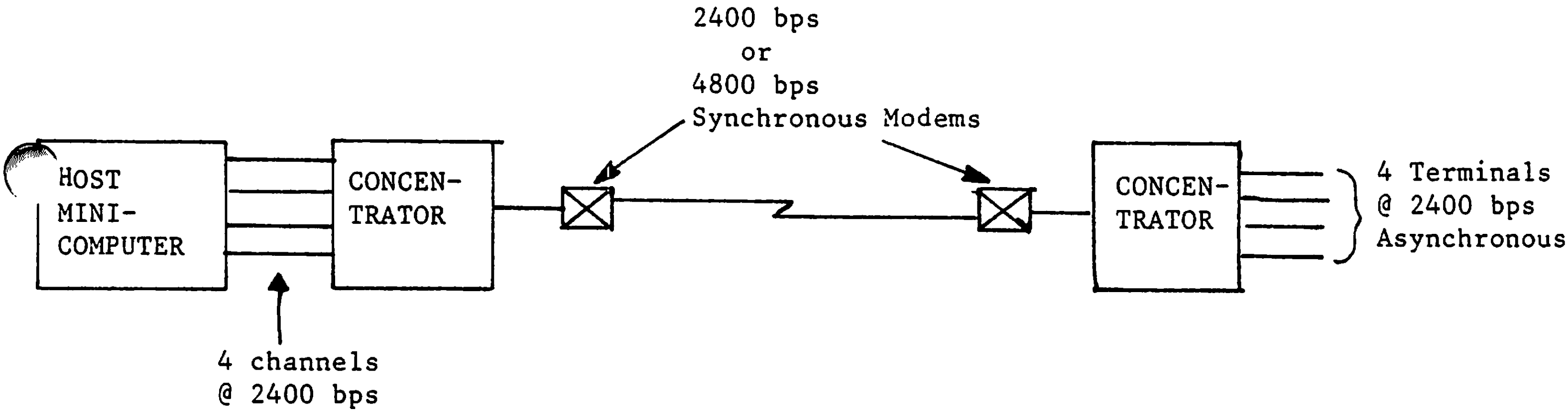


Figure 4-3. Typical Data Concentrator Configuration

In operation, the concentrator assembles characters from each active channel, building variable length data blocks for transmission down the shared high-speed line. Transmitted block length is a function of the amount of data accumulated since the last block was transmitted. All data blocks received from the high-speed line are checked for errors, and retransmissions are requested for any block received incorrectly.

4.3 The Multidrop Concentrator

The third product in the ADLC™ family is the Multidrop Concentrator.

The Multidrop Concentrator uses the same statistical multiplexing techniques as the Data Concentrator, but since it can be configured in a multipoint configuration it allows individual dumb terminals and dumb terminal clusters to be multidropped at different locations to share a single multipoint telephone line.

The typical Multidrop Concentrator system links multiple regional offices with a central computer. In the example shown in Figure 4-4, the master Concentrator unit at the computer site is configured with 8 channels, each of which provides to the computer an apparently point-to-point transparent connection to a terminal in one of the regional offices. Regional offices are equipped with node Concentrator units, configured with 1, 2, or 4 channels.

While a retransmission is taking place, incoming data from the terminal or computer is temporarily stored in the Error Controller's buffer memory. Despite the fact that several hundred characters of memory are provided, buffer overflow may occur, either because of excessive retransmissions or a prolonged line outage.

However, options may be provided for DEC, Hewlett-Packard, Data General and other minicomputer users which take advantage of the fact that these minicomputer systems will suspend transmission either on receipt of a special control character (XOFF) or on the dropping of the Clear-to-Send interface control signal. Transmission will resume when the system receives the XON control character or when Clear-to-Send is raised.

4.2 The Data Concentrator

The second product in the ADLC™ family is the Data Concentrator.

The Data Concentrator uses the statistical multiplexing techniques discussed in Section 3 to allow several 'dumb terminals' to share a single telephone line, with one concentrator unit installed each end of the line (see Figure 4-3). The device operates extremely efficiently with interactive CRT terminals, since it allocates the shared telephone line to each terminal dynamically, as needed, rather than on a predefined fixed basis. As a result, for example, it can allow four CRT's each operating at 2400 bps to share a single 2400 bps line. In addition, the protocol used between the concentrators incorporates the same retransmission-on-error facility provided for single terminals by the Error Controller. Thus, the Data Concentrator acts as both an error controller and cluster controller for as few as two terminals, without requiring any changes to the existing 'dumb terminal' hardware and minicomputer software. Incidentally, it also allows the asynchronous terminals to be used with high-speed synchronous modems.

The benefit of the true communication protocol, supported in software on the host computer, is that it allows multiple terminals to be polled using a single computer port as well as a single telephone line. But unless special software resides in the host computer, it is not possible to support multiple terminals on a single port. The purpose of the Port Concentrator is to simplify the special software required.

The Port Concentrator is used in conjunction with a Data Concentrator as shown in Figure 4-5.

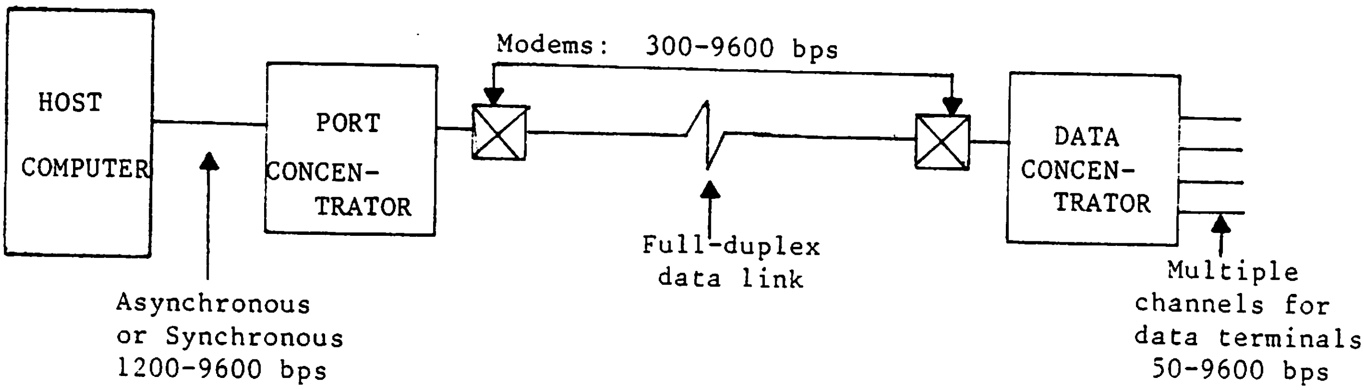


Figure 4-5. Typical Port Concentrator Configuration

The Port Concentrator functions as a single channel master Data Concentrator, allowing one computer port to communicate with multiple channels attached to a remote Data Concentrator using a simple asynchronous or synchronous protocol. It allows transmissions to and from the host computer to consist of simple lines of text, preceded by a terminal address.

The Port Concentrator performs more than half the work necessary to communicate directly with a remote Data Concentrator. As previously discussed, all data flow between Data Concentrators is transmitted in numbered blocks, each terminated by a 16-bit Cyclic Redundancy Check character. The Port Concentrator assumes full responsibility for the complex task of error control and only transfers to the host computer error-free data blocks. In addition, data

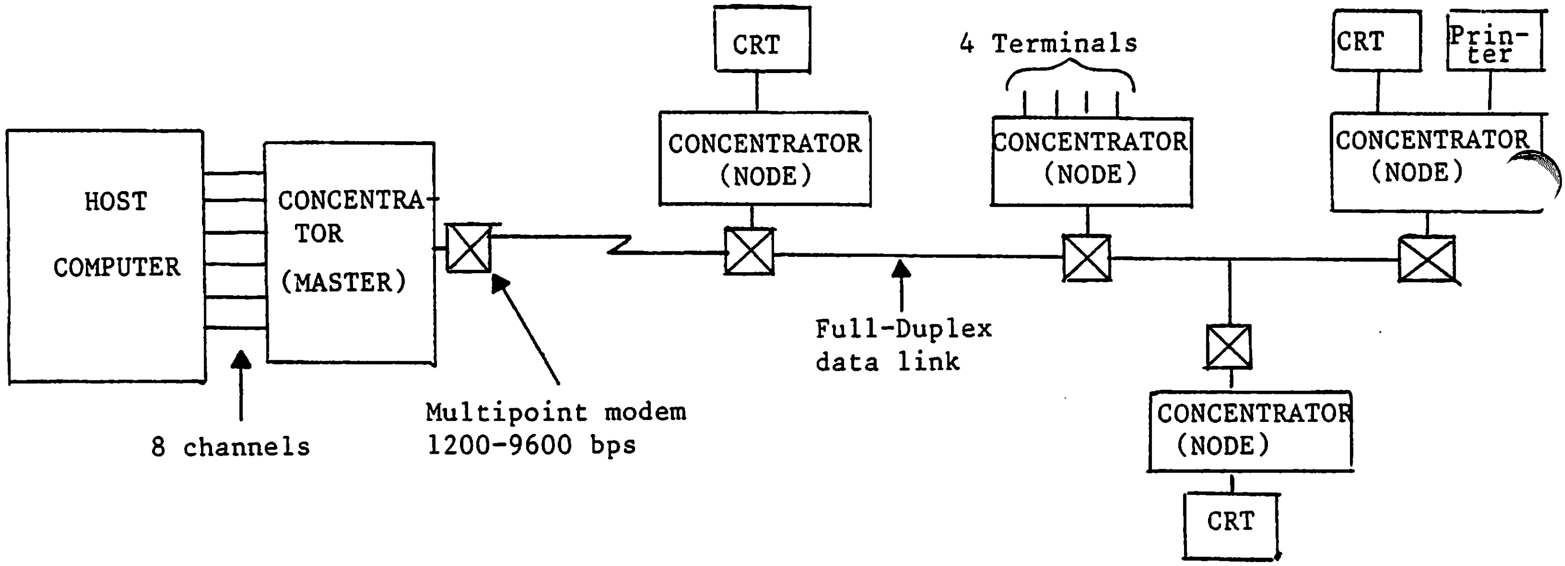


Figure 4-4. Typical Multidrop Concentrator Configuration

In operation, the master Concentrator polls the remote node units at high-speed, accepting multiplexed inbound data from terminals at each site and delivering multiplexed outbound data addressed to individual terminals at each site. The multipoint data link operates synchronously at speeds to 9600 bps, but speeds of 2400 or 4800 bps are most typical. The individual 'dumb terminals' multiplexed by the node Concentrator at each site may operate at virtually any asynchronous data rate to 4800 bps.

The combination of multidrop capability, terminal cluster control and automatic retransmission-on-error, all without the need to modify existing 'dumb terminal' hardware and minicomputer software make the Multidrop Concentrator the most sophisticated of the ADLC™ products.

4.4 The Port Concentrator

The last of the ADLC™ products is the Port Concentrator.

All the other ADLC™ products require no changes to existing hardware and software. They preserve the one-to-one relationship between each terminal and its associated port on the minicomputer. But in some configurations ports may be limited. Some type of port-sharing device may be desirable.

5. REDUCING PORT COSTS AND OPTIMIZING MULTIPLE COMPUTER ACCESS

As discussed in Section 4, the cost of computer ports can be reduced by allowing several terminals to share a single port. The Port Concentrator offers one solution, but requires special software support. But applications which involve a 'session', where a terminal need only be connected to the computer for a relatively short period of time, offer a simpler solution.

5.1 The Advantages of Dial-Up Access

Port sharing can be achieved by using dial-up access to connect terminals to the computer. Dial-up access provides two significant benefits.

5.1.1 Contention

Dial-up access allows a large number of terminal users to contend for a smaller number of computer ports on a first-come-first-served basis with the telephone rotary providing a busy signal when all ports are in use. The contention provided by the telephone rotary typically allows support of up to four terminals per computer port in a typical time-sharing application. This contention results in considerable savings in computer port hardware and much better utilization of the computer ports installed.

5.1.2 Port Selection

Dial-up access also allows terminal users to gain access to ports connected to different computer systems. Figure 5-1 shows a computer facility with two HP 3000 systems used for business applications and one DEC system supporting time-sharing. Each of the three computers is equipped with 16 ports. Three separate telephone rotary groups are assigned and any terminal may access any of the three services by dialing the appropriate telephone number. Thus, dial-up access may be used to provide a port selection facility.

buffering requirements caused by backup of data owing to retransmissions or temporary line outages on the link to the remote Data Concentrator are handled automatically by the Port Concentrator.

The Port Concentrator also assumes responsibility for synchronization with the remote Data Concentrator. This is necessary even when there is no data activity in order to ensure rapid response and minimum delay whenever there is data to transmit. This need for constant synchronization might otherwise impose a significant burden on the host computer. The Port Concentrator assumes all of this burden.

As a device to economise on computer ports, the Port Concentrator has only one disadvantage: unlike other ADLC™ products, it does require special software support. This is unavoidable, however, if all terminals sharing the port need to be on-line at all times. But for applications which involve a 'session', where a terminal need only be on-line for a period of a few hours at one time, other solutions are available which permit port sharing without requiring any software changes. These are discussed in Section 5.

5.2.2 The PABX Burden

The costs resulting from Single Message-Unit Rate Timing do not, of course, apply to users with in-house PABX or Centrex service. But the long duration of terminal-computer conversations which has caused the telephone companies to seek a new tariff also causes bottlenecks in PABX systems whose design criteria also are based on many calls of short duration rather than a few calls of long duration.

Use of the PABX for computer access may not be as 'free' as it appears: it may result in the need for a larger PABX long before the voice traffic warrants it.

5.2.3 The Cost of Modems

Dial-up access demands the use of modems, even for terminals which are at the computer site and within a short distance of the computer. Direct terminal connection, even if the distance involved requires the use of Line Drivers or Local Datasets, can involve substantial cost savings through the elimination of modems whether rented from the telephone company or purchased directly from an independent modem manufacturer.

5.2.4 Transmission Speed Restrictions

As discussed in Section 2, a dial-up connection is a two-wire connection, and the maximum speed for full-duplex data transmission over 2-wire circuits is 300 bits per second, or 1200 bits per second with the more expensive Racal-Vadic VA3400 or Bell 212A. Since dumb CRT terminals operate much more effectively at speeds higher than 1200 bps, the dial-up speed limitation is significant.

5.2.5 Peak Period Panic Dialing

Dial-up access normally implies that some callers will receive the 'busy' signal at peak periods. Unfortunately, the dial network gives the caller no opportunity to wait on the line until the 'busy' condition clears. He must hang up and re-dial. No matter how long a caller has been waiting and no matter how many times he has tried to get through, the first available line will be connected to the caller fortunate enough to be the first to dial in after

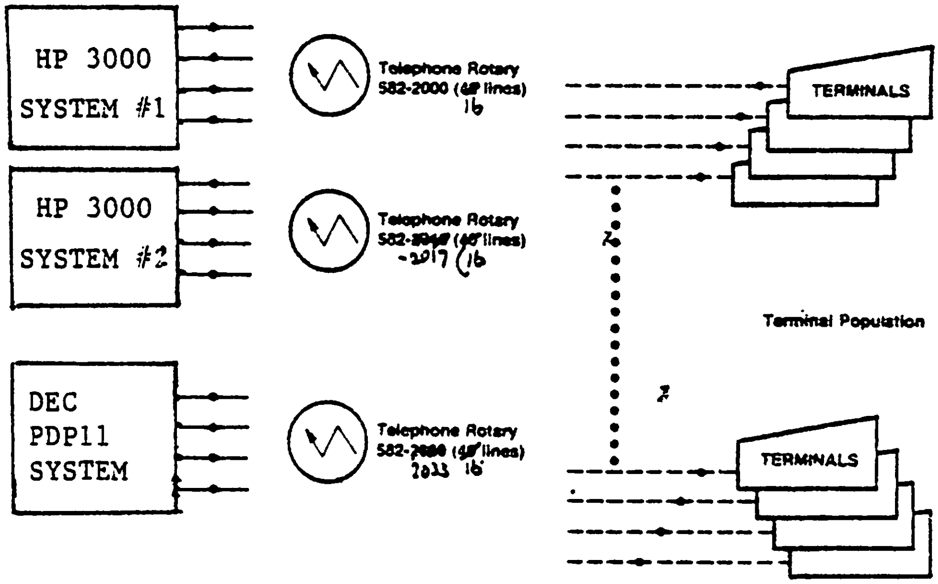


Figure 5-1. Dial-Up Access, Providing Contention and Port Selection

5.2 The Disadvantages of Dial-Up Access

The disadvantages of dial-up access are all a result of the fact that the dial-up telephone network was designed for telephones, not data terminals. Dial-up access involves both cost factors and performance limitations which may be intolerable.

5.2.1 The Increasing Cost of Dial-Up Telephone Calls

In the past, dial-up access was a very cost effective method of terminal-computer connection. Originally, the telephone companies based their tariffs on voice usage. A typical telephone call lasted from 3-5 minutes and billing was based on the number of completed calls. With the growing use of the switched telephone network by data terminals, the length of a typical call has become much longer. Many terminal users dial the computer in the morning and remain connected throughout the business day. The telephone companies decided to penalize terminal users by implementing a new tariff, under which local service by business users would be measured and calls timed in increments of five minutes or less with a single message-unit charged for each increment. The conversion to Single Message-Unit Rate Timing will eventually affect all parts of the United States.

the line has become free, perhaps even on his first attempt. This gives rise to 'panic dialing' at peak periods, as frenzied callers repeatedly attempt to assert their 'right' to the next available line.

5.3 What is a Port Selector?

The role of the Port Selector is to control and coordinate terminal access to a computer facility, integrating dedicated terminal connection with dial-up access if required and providing the advantages of both without the disadvantages of either.

The Port Selector is installed between the computer and the terminals as shown in Figure 5-2. Like the telephone rotary, the Port Selector provides first-come-first-served contention between a larger number of terminals for a smaller number of computer ports. But unlike the telephone rotary this facility is offered to all terminals whether access is dial-up or dedicated.

In this simplest type of application, all terminals are in contention for all ports . Alternatively, the ports may be partitioned into 'class' groups, providing contention for each computer system in the same manner as the configuration shown in Figure 5-1 using multiple telephone rotaries. When multiple port 'classes' are defined to the Port Selector, port selection from the terminal is achieved by entering the desired 'class' (i.e. computer system) from the terminal keyboard, rather than having to dial a different telephone number for each system.

Thus, the Port Selector provides the same Contention and Port Selection facilities provided by the dial-up telephone rotary, without requiring dial-up access.

5.4 How a Port Selector Works

The heart of a Port Selector such as MICOM's Micro600 is a time-division switch, a solid-state electronic version of the electro-mechanical crossbar switch used in most conventional telephone

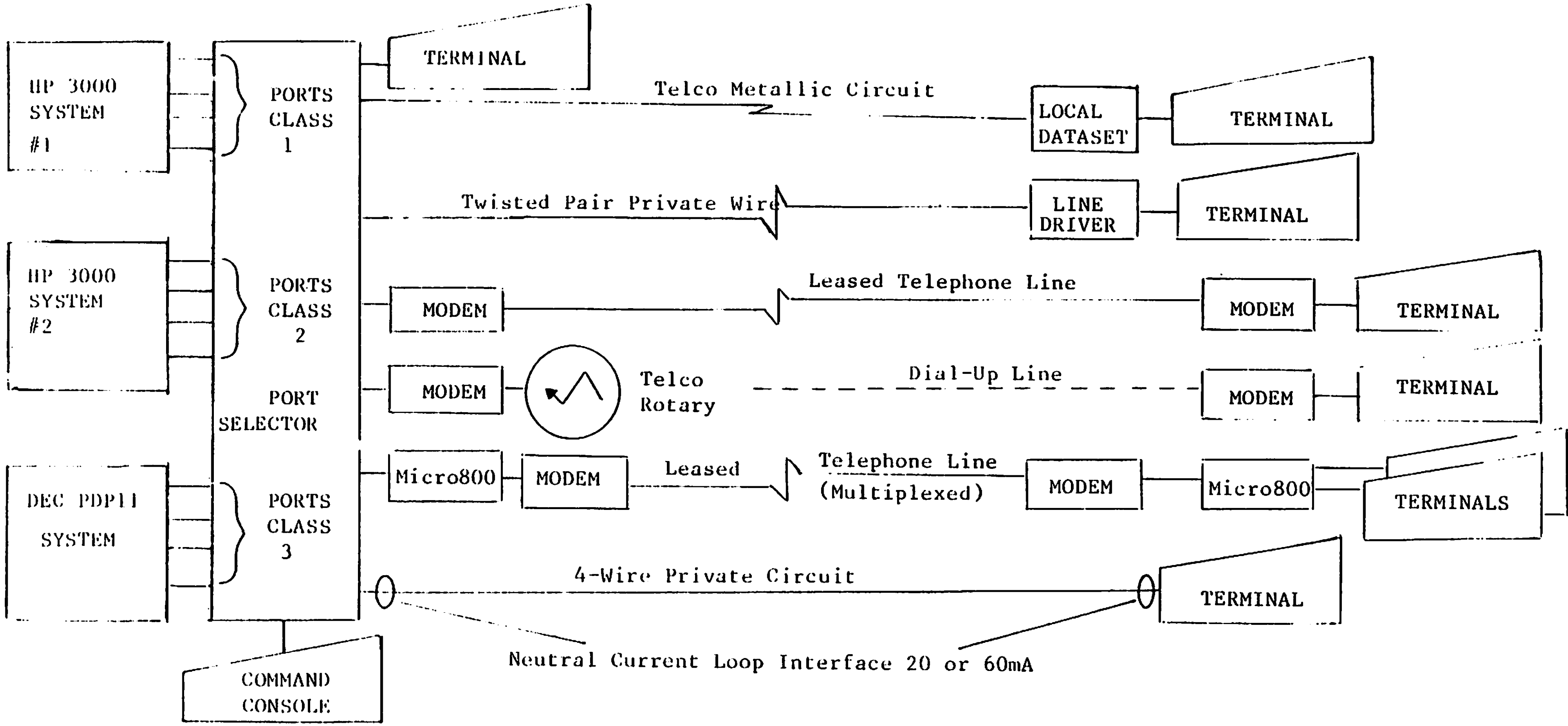


Figure 5-2. Typical Port Selector Configuration

5.5 Port Savings and Access Management

The Contention facility provided by the Port Selector can be used to reduce the number of computer ports required to support a given number of terminals without the need for dial-up access. The Port Selection facility can be used to permit multiple computer access for dedicated 'dumb terminals'. In addition, the Port Selector can be used to restrict access to certain computer ports from certain terminals. It also operates independent of the computer(s) to transmit special messages to terminal users to advise of system problems and scheduled restoral of service after down-time.

With a Port Selector, the terminal connection can be made in the most cost-effective manner appropriate to each individual terminal, either by direct interface cable, limited-distance line driver or local data set, or modems, dial-up or dedicated, without restricting the freedom of any terminal to access the computer port of its choice.

In addition, the Port Selector can maintain usage statistics, allowing the computer manager, for the first time, to monitor usage of each group of computer ports. The Port Selector thus allows proper Access Management and significantly improves the ability of the computer manager to ensure an optimum level of service to all terminal users, while keeping computer port costs to a minimum.

exchanges. The time-division switch operates under control of a microcomputer which controls all connections and disconnections. Once a connection is established, operation is totally transparent, the time-division switch transfers data directly from terminal to port at very high speed, and the microcomputer controller is only activated when the connection needs to be broken.

To the computer port, the Micro600 may appear either as a dedicated terminal or as a modem emulating the full answering sequence of a Bell 103 modem. Each 'port' interface on the Micro600 has a 'class' defined in the Micro600's control memory. This 'class' may be modified from the Command Console of the Micro600 at any time.

Terminals connected to the Micro600 may be directly cabled or connected by line drivers, local datasets, or modems, dial-up or leased line. The terminal operator requests connection by depressing any key on his keyboard. The Micro600 responds with the prompt message "CLASS=". The terminal operator must then key the desired class number (I through 127). If the Micro600 can make a connection to a port of the desired class, it will do so, transmitting "GO" to the terminal. If unsuccessful, it will transmit "BUSY", "UNAVAILABLE", "UNASSIGNED", "UNAUTHORIZED" or "WRONG SPEED" as appropriate. If ports are busy, the operator may elect to 'camp on', or wait in line; he is automatically told how many terminals are ahead of him and receives a "GO" message as soon as it is his turn to be connected.

Terminals are disconnected either when the Micro600 'port' interface sees that the computer port has dropped the Data Terminal Ready interface signal or on detection of 'Break' from the terminal or after a period of inactivity of predefined length.

6. SUMMARY

The subject of data communications is extremely broad. In this General Introduction, we have addressed those aspects which we believe to be most significant to the minicomputer user.

Those who wish to study the subject in greater depth may choose from a wide range of available text books. Several are particularly recommended as follows:

Basic Techniques in Data Communications, Ralph Glasgal
Advanced Techniques in Data Communications, Ralph Glasgal
Introduction to Teleprocessing, James Martin
Telecommunications and the Computer, James Martin
Teleprocessing Network Organization, James Martin
Technical Aspects of Data Communication, John E. McNamara