THE UNIVERSITY OF MICHIGAN

Technical Report 20

Topics in Computer Communications Systems

David L. Mills

CONCOMP: Research in Conversational Use of Computers
F.H. Westervelt, Project Director
ORA Project 07449

supported by:

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

May 1969

Topics in Computer Communication Systems

ABSTRACT

This report surveys several aspects of data communications systems applicable to timesharing utilities. Topics presented include propagation characteristics of the telephone network and comparisons of the various modulation techniques suitable for use on this network. A description of several types of coding systems is presented and various aspects of these systems are studied, including those of synchronization and graphic/control-function interchange assignments. A detailed description of a proposal for a half-duplex processor-processor communication link protocol is presented to illustrate the problems in a practical design. Finally, a brief description of the operation of certain types of popular terminals and terminal control units is presented, along with a description of a special terminal control unit constructed by the University staff. An extensive bibliography is included.

This report was prepared using FORMAT, a computer program in MTS, the Michigan Terminal System. This program is described in: Berns, G.M., Description of FORMAT, a text-processing program. Comm. ACM 12, 3 (March 1969), pp. 141-146. The text was entered to this program partly in punched-card form and partly directly from a typewriter terminal and was printed on an IBM 1403 printer equipped with a TN print train.

Topics in Computer Communication Systems

TABLE OF CONTENTS

FIGURES

TABLES

1. INTRODUCTION

With the increasing number of timesharing systems now becoming operational there has been an increasing awareness of the problems of data communication between a centrally located computer complex and remotely located terminal equipment. The types of terminal equipment now available cover a wide spectrum of operating speeds and complexities and range from simple arrays of telephone-instrument pushbuttons through typewriter-like devices to complex visual display and data-gathering systems which are actually computer systems in themselves. Each of these devices, it seems, operates with an individually tailored transmission code and message protocol, and their connection to the parent timesharing system can be an extraordinarily complex operation. Furthermore, much of the commonly used terminal equipment has been designed originally for purposes other than computer communication; and the feeling persists in particular cases that the equipment is either too complicated for the average user to operate or is missing a vital feature for a particular use.

Discussions of this type might lead the system designer to suspect that insufficient effort has been made in the past to place in perspective those problems in the general communications environment which have peculiar import in timesharing systems operation. In this report a concerted effort will be made toward this goal. To illustrate the general problems involved, comparisons will be made between generally available communications systems and operating techniques. Most of the discussion will involve hardware-oriented topics such as transmission codes and message protocols and the architecture of equipment design. Related discussions in software-oriented topics can be found in a companion report [26].

In this report background will be drawn from several engineering and programming fields, as well as from manufacturers' specifications of generally available systems. The reader will be assumed to have at least a passing familiarity with the architecture of timesharing systems and the general nature of communications devices. In general, the discussions of the following sections will proceed from communications-theoretic models through their application in useful cases to examples drawn from currently operating systems. Some topics will be skimmed rather briefly for the purposes here, and in these cases references will be made to the bibliography at the end of this report.

2. THE COMMUNICATION NETWORK

Consider a typical data system, which happens to be in use at the University of Michigan, as illustrated in Figure 1. Here the remote equipment is a combination of several kinds of terminals operating at different transmission rates and feature environments. In very many cases the transmission links are provided by a common-carrier company on a dial-up basis using the same facilities as the common telephone although, in some cases, a transmission link may be a private leased telephone line or radio facility. Almost all of the discussion in this report will be concerned with common-carrier dial-up facilities.

A transmission link may be designed to propagate messages in both directions simultaneously (full-duplex), in one direction only (simplex), or alternately either one direction or the other (half-duplex). Full-duplex operation is of course preferred in any case involving conversational timesharing communications, but has the disadvantage of requiring either sacrifices in transmission rates or delicately-adjusted equipment. Common-carrier dial-up transmission equipment can be operated in a full-duplex mode up to about 300 bits-per-second (baud) using appropriate data sets and in a half-duplex mode up to 3600 baud using the newest data sets. Private wire and radio transmission links are almost always inherently half-duplex in operation and full-duplex operation is possible only by the addition of a separate but identical facility, operated so that each facility is dedicated to a single direction. Note that a specification of a half-duplex communications device to operate on a full-duplex transmission link is not incompatible; and in fact is a common (although wasteful) one.

At the terminus of each transmission link is presumably some kind of communications device or terminal or even another timesharing system. The common terminal equipment takes the form of a typewriter-like device operating at speeds appropriate for human operators. Appropriate devices are manufactured by several suppliers, some of which will be described in detail in a later section of this report. With the ready availability of low-cost display and batch-entry equipment, some of these terminals can become quite complex and provide message distribution and store-and-forward operations as well.

```
+----------+     +----------+
|          |     |          |
|   2703   |     |          |
|          +-----+          |
|  16 IBM  |     |          |
|  32 TTY  |     |          |
|          |     |          |
+----------+     |          |   Transmission
                 |          |   Facilities:
+----------+     |          |
|          |     | Data Set |
| (2) 2701 |     |          |   Switched:
|          +-----+  Equip:  |   78 lines Centrex
|  2 SDA   |     |          |
| IBM BSC  |     |          |
|          |     | 64 103E  |
+----------+     |          |   Leased:
                 |  6 201A  |   2 voice grade
+----------+     |          |
|          |     |  6 403E  |
|   DATA   |     |          |
|CONCENTRAT|     |  2 202C  |   Radio circuit:
|          |     |          |   HF SSB 175 W PEP
| 16 TTY/  |     |  2 801C  |   (Amateur exper.)
|.   IBM   |     |          |   TTY and voice
|          |     |          |
|  4 SDA   |     |          |
| IBM BSC  |     |          |
|          +-----+          |
|  4 SPEC  |     |          |
| CCI DSP  |     |          |
|          |     |          |
|  2 SPEC  |     |          |
|TOUCH TONE|     |          |
| DIG RCVR |     |          |
|          |     |          |
|  4 SPEC  |     |          |
|AUTO CALL |     |          |
|          |     |          |
+----------+     |          |
                 |          |
+----------+     |          |
|          |     |          |
|   7772   +-----+          |
|AUD RESP U|     |          |
|          |     |          |
+----------+     +----------+
```

Figure 1. Typical Timesharing Communication Network

2. The Communication Network

No matter what kind of communications device is used on a particular link, its interconnection with external equipment will be assumed on a serial character-by-character basis. That is, messages are assumed to be sequences of characters which are themselves fixed-length sequences of binary digits. The parameters which describe a particular communications device then include its functional interpretation of all the various character-code combinations and the speed of transmission of the serial bit stream within a message. Furthermore, a specification of the device behavior to control functions, especially in half-duplex modes, is required.

A connection via the telephone network is classed as exchange if it is between two points serviced by the same central office. A connection is classed as short-haul if it is between two points less than about 350-400 miles apart and as long-haul if it is at greater distances. Exchange connections are likely to be of comparatively high quality, and are inherently capable of data transmission at the highest rates. Short- and long-haul connections are likely to be made via wire and microwave radio carrier facilities. Signal degradation on such facilities occurs at the repeater stations where the signal is amplified and at the terminals where the signal is multiplexed along with others on the same physical circuit.

Most of the timesharing systems now operational on the switched network are concerned primarily with operation within a single exchange area, with only token use of short- and lond-haul circuits. Since the quality of exchange connections is higher than that of the other types, it should be possible to attain better performance. Unfortunately, all currently available data set equipment is designed for worst-case operation on the longer haul facilities and no attempt is made to take advantage of the better characteristics of the exchange plant in particular cases.

In the United States all common-carrier companies are privately owned, but are closely regulated by various governmental agencies. The various services offered and the rates charged are established by tariffs negotiated with the Federal Communications Commissicn and the several state equivalents. The various operating companies are organized on an intrastate basis, however, and can get away apparently with a good deal of mischief. Thus the rates charged for the same service may vary in the various states and the available equipment may not be identical.

Until recently the common-carrier companies have provided connection to the switched telephone network only by means of data set equipment of their own manufacture. Their announced reason for this requirement was that the special operating conditions of the telephone plant required protection and isolation between the user and the network for their mutual protection. However, late last year a court action found against one of the companies and required the common carriers to establish a tariff requiring only a simple coupling device to be used between the customer's equipment and the telephone line [16]. Such a tariff was allowed to go into effect early this year, providing for a so-called "Data Access Arrangement" [A3]. It can only be hoped that this "raw data" connection will foster the marketing of well-designed but inexpensive data set equipment available for this connection.

Some of the operating companies have been more agile than others in providing their own local options to this tariff, but Michigan has not been one of these. Equipment is apparently even available in Pennsylvania and Illinois for the connection of amateur radio "phone patch" equipment which has heretofore been in widespread, although illegal, use [20, 40].

3. THE COMMUNICATION CHANNEL

Propagation of the data signal itself between a central timesharing utility and its satellite terminals is most usually via a transmission link designed for speech-signal transmission. Thus, some kind of transmitter and receiver is required at each end of the link to adapt the serial bit-stream used by the terminal to the continuously variable signal used by the transmission link. These adapters, often called modems (for modulator-demodulator) are almost always furnished by the common-carrier company on a lease basis, although recent changes in the tariffs make it possible to use equipment not manufactured by the common carrier. Depending upon the type of modem and the transmission method involved, the actual signal transmitted on the link may take several forms and may achieve various degrees of optimality. In this section the parameters of the typical telephone transmission link and the characteristics of typical modem equipment will be summarized.

3.1 The Transmission Link

Since telephone circuits are most often used for the transmission of speech signals, the equipment design and performance parameters most often are expressed in units appropriate for the speech waveform. Unfortunately, the transmission of high-speed data over telephone circuits requires a different design emphasis using parameters not generally available. During several years of experience with data transmission on the switched telephone network, however, compromises in modem designs and the addition of certain circuit features within the telephone plant have become reasonably effective.

A telephone (or radio) circuit is usually characterized by its response to steady-state sine-wave signals at various frequencies of interest together with its inherent noise properties. The usual measured transmission parameters include the passband, or the useful frequency limits between which the attenuation is small, the attenuation and propagation delay at the various frequencies within the passband, the amplitude linearity, determined as a fraction of harmonically related distortion products and the intrinsic noise level, determined by a specified averaging process. A particular circuit is characterized when the values of these parameters are known. These parameters are not fixed with time, however, and depend for instance on such factors as switching paths through the telephone network and ionospheric conditions on a high-frequency radio circuit. Common telephone circuits have passbands from a few hundred to perhaps 3000 cycles-per-second (Hz) on short to medium-haul circuits typical of those useful in local or regional timesharing systems. Propagation delays of up to a third of a second are likely on long-haul circuits. Delay distortions of perhaps a millisecond and amplitude distortions of perhaps a few percent are not uncommon on these circuits. The noise process is substantially impulsive in nature, due to transients generated by the switching equipment. The effects of these characteristics are well known and devices which improve performance through artificial compensation on particular circuits are readily available. The characteristics of the switched telephone network which are of principal importance to data transmission are discussed at length in [1, 7, 15, 21, 28, 29, 30, 42].

One characteristic of transmission channels which is much more critical for data transmission than for speech transmission is the attenuation of the signal over relatively short time intervals. The human ear is relatively insensitive to changes in signal amplitude over periods longer than a large fraction of a second. In some telephone repeater equipment an amplitude limiting device (compander) compresses the dynamic range of the signal at the transmitting end of link and a corresponding device expands the range at the the receiving end [9]. The time dependency of this operation has a disastrous effect on any data signal which encodes the data as a series of amplitude changes. Thus, a useful data signal for such circuits must be encoded as a constant peak-amplitude signal.

In a half-duplex transmission link a very critical characteristic is the turnaround time necessary to flip the circuit end-for-end. In local and exchange telephone circuits the turnaround time is limited only by the circuitry of the modem itself; but in longer-haul circuits a special device called an echo suppressor is often inserted in the repeater equipment to avoid line reflections [6, 8, 10, 22]. These devices operate by sensing a speech (or data) signal originating at one end of the circuit and disabling transmission originating at the other end during the interval this signal is present. As a special concession to full-duplex data systems, most of this equipment is designed so that a continuous tone signal at a certain frequency will disable the echo suppressor operation, allowing transmission in both directions simultaneously. When the echo suppressor has been actuated so that transmission is allowed only in a particular direction, a short interval, equal at least to the round-trip signal propagation time, is necessary for it to change direction of transmission, and it is this figure which places the lower limit on turnaround times in long-haul telephone circuits.

3. The Communication Channel

3.2 Modulation and Demodulation - Data Set Equipment

Since the characteristics of the switched-telephone network are effectively fixed for the purposes here, the only real design freedom exists at the point where the serial bit stream is encoded into and decoded from the continuously variable transmission link signal. In this section techniques for accomplishing these processes, called modulation and demodulation, will be compared to each other and to theoretical upper bounds on performance.

The objective in modulator/demodulator design is to maintain the highest possible transmission rate together with the lowest error rate given a transmission channel of fixed characteristics. Sometimes, of course, other considerations enter into the design such as circuit complexity and economics and whether the link can operate in a full-duplex or half-duplex mode. A measure of ideality can be established for a given circuit against which the performance of the various modem techniques can be measured. If the channel itself can be assumed to have a sharply delimited passband and to operate in a white-guassian noise process, then the transmission rate can approach the Weiner-Shannon bound W log(1+S/N) baud, where W is the bandwidth, S the signal power and N the noise power over this bandwidth [41]. This bound can be defined as the channel capacity. Using typical figures W = 2000 Hz and S/N = 40 dB, the transmission rate can evidently approach 24,000 baud. Thus, with the best possible modem encoding in this idealized environment, the transmission rate could reach many times the bandwidth of the speech channel.

Now consider the end-to-end behavior of the channel including the modem equipment. From this viewpoint the transmission process is a simple serial transmission of a continuous bit-stream with errors introduced occasionally due to the additive noise. A famous theorem due to Shannon indicates that it is possible to encode in such a way as to realize a transmission rate arbitrarily close to (but not greater than) the channel capacity and with an arbitrarily low (but not zero) error rate. Therefore, without even looking at the design of the modem equipment itself, a coding technique with sufficient redundancy (but perhaps arbitrarily long encoding/decoding delays) can evidently be designed to realize the channel capacity W log(1+S/N) in this sense. Now, another famous theorem due to Nyquist indicates that binary data transmitted at a rate N baud over a strictly binary channel requires a bandwidth N/2 Hz for error-free transmission [31]. Evidently then a rate of 4000 baud could be sustained over a channel of bandwidth 2000 Hz.

In the following paragraphs refer to Figure 2, which illustrates the power spectra of the signal transmitted by the various data set equipment described in Table 1. In Figure 2 F is the carrier frequency and M is the highest modulation frequency. The Nyquist result applies to the unmodulated (baseband) signal. The simplest modulation technique is one in which a constant-frequency carrier signal is turned on and off according to whether the serial binary input is one or zero respectively. This has the effect of translating the passband of the signal to new values, so that the signal can be propagated via the telephone network facilities. However, this modulation process doubles the required bandwidth by creating two sidebands, each an image of the baseband spectrum but placed either side of the carrier.

If we amplitude-modulate the serial binary signal on a carrier frequency and filter out one of the sidebands the Nyquist results still apply. Now, assume the transmitter collects a sequence of n binary digits and codes each such sequence into 2**n different amplitude levels. If the receiver can distinguish this quantization without error then the transmission rate will be n times the Nyquist rate. This technique, called vestigal sideband modulation, is used in the Bell system 203 data sets for transmission rates to 3600 baud on the switched network and 5400 baud on leased lines [A8].

The problem with the vestigal sideband equipment when used on the switched network is that the peak amplitude of the transmitted signal is not necessarily constant with varying bit patterns as input. The 203 data set solves this problem by artificially scrambling the input sequence using an interesting shift-register circuit. The problem with this scrambler and all related techniques is that:

1) The encoding delay required (a matter of seconds) places a lower limit on the turnaround time.

2) Any scrambling sequence of finite length can be unscrambled and defeated by an appropriate data sequence, and the likelihood of almost any sequence using transparent data techniques is uncomfortably high.

```
 |
 +------+
 |      |
 |      |
-+------+- f
 0      M
```

Baseband

```
 |
 |  +------+------+
 |  |      |      |
 |  |      |      |
-+--+------+------+- f
 0 F-M     F     F+M
```

Amplitude Modulation
Phase Modulation (201, 301, 303 data sets)

```
 |
 |  +------+
 |  |      |
 |  |      |
-+--+------+- f
 0  F     F+M
```

Vestigal Sideband Modulation (203 data set)
(upper sideband)

```
 |
 |  +------+------+  +------+------+
 |  |      |      |  |      |      |
 |  |      |      |  |      |      |
-+--+------+------+--+------+------+- f
 0 F1-M    F1  F1+M  F2-M   F2   F2+M
```

Frequency Modulation (103, 202 data sets)

Figure 2. Modulation Spectra

3. The Communication Channel

Data Set	Used on	Rate	Type	Facil	Features
103A	S	to 300	FS	FDX	space disc
103F	4A	to 300	FS	FDX	
202C	S	to 1200	FS	HDX	rev chan
202D	4A	to 1800	FS	FDX	(5 baud)
201A	S	2000	4-PM	HDX	int or ext
201B	4A	2400	4-PM	FDX	clock;
					short COD
X203A	S	to 3600	VSB	HDX	rev chan
X203A	4A	to 5400	VSB	FDX	(150 baud)
					12 sec COD (!)
301B	Telpak	40,800	4-PM	spec	special
303-	Telpak	to 230,400	DSB/VSB	spec	special (very)
401-	S	to 20	MF-3/14	HDX	Touch Tone
402-	S	to 75	MF-8/8	HDX	Dataspeed
402-	4A	to 75	MF-8/8	HDX	
403-	S	to 10	MF-2/8	HDX	Touch Tone

Table 1. Comparison of Data Sets

3. The Communication Channel

Still another technique commonly used is frequency-shift modulation, in which one of two carrier frequencies is transmitted (see Figure 2), one for the binary one digit (F1), and the other for the binary zero digit (F2). Using the principle of superposition this can be analyzed as two separate amplitude-modulated carrier signals, one at each of the two operating frequencies. Although the receivers in such a system can make use of the redundancy in increasing the signal-to-noise ratio, the primary reason for adopting this technique is its simplicity and the fact that telephone repeater and echo-suppression equipment are well behaved using these signals. The effective transmission rate using frequency-shift modulation is, as can be seen, about one-fourth the Nyquist rate, or about 1000 band. The Bell System 202 data sets operate in this fashion at rates to 1200 baud on the switched network and 1800 baud on leased lines [A7].

Using a combination of bandpass filtering and directional couplers (hybrid circuits) any of the modulation techniques can be used in full-duplex systems. For practical reasons, however, only the frequency-shift technique is used in such service, and this at rates below 300 baud. The Bell System 103 data sets operate in this fashion [A4, A5]. These data sets are extensively used for manually operated keyboard/printer equipment and are designed to provide automatic-answer and disconnect when used in this service.

Still another technique used is one in which the phase of the transmitted carrier signal is varied depending upon the serial-binary input digits. In the simplest form the carrier phase is simply inverted or shifted 180 degrees depending upon the input digit. The Bell System 303 data sets operate in this simple fashion at transmission rates to 230,400 baud on special and exotic leased lines [A10].

Most commonly, however, a sequence of two input digits is accumulated and used to quantize the transmitted carrier phase into four discrete levels. Again using the principle of superposition, the spectrum of this signal can be analyzed as four amplitude-modulated carriers of the same frequency but different relative phases. The spectrum then is simply that of the usual amplitude-modulated signal as shown in Figure 2. The effective transmission rate of a four-phase modem is, as can be seen, about equal to the Nyquist rate, or about 4000 baud. For various reasons, the practical limit of this technique is about half of this.

The Bell System 201 data sets operate in this fashion at transmission rates to 2000 baud on the switched network and 2400 baud on leased lines [3, A6]. The Bell System 301B data sets use this technique at transmission rates to 40,800 baud on specially conditioned leased lines [A9]. The demodulation process used in a phase-modulation receiver is inherently capable of recovering the transmitter bit timing so that complicated external circuitry for this purpose is not necessary. Furthermore, since the envelope of the transmitted signal is constant as in frequency-shift modulation, no scrambling is required.

Finally, still another technique is used within the switched network itself to transmit dialing digits from the customer's telephone to the central office. This technique is not strictly a modulation process at all, but is rather a multiplexing technique. To each pushbutton on the Touch-Tone telephone, there corresponds a unique pair of carefully chosen frequencies which are transmitted when the pushbutton is depressed. At the central office these frequencies are extracted from the telephone line signal using rather complex circuitry to insure freedom from interference by coincident voice signals. Upon extraction these signals are used to actuate the dial-digit receiving equipment in the central office. Bell System has made available in data-set form a number of different dial-digit transmitters and receivers for general data-communication use [4, A11-A16]. Most of the equipment, such as the 401 and 403 series, operates using a restricted 2-out-of-8 or 3-out-of-14 code giving 16 or 99 different code combinations respectively. The operating speeds of this equipment are specified at between 10 and 75 codes-per-second.

The principal advantage of these multifrequency systems is that in very many cases no serialization/deserialization process is required and that the codes are inherently self-synchronizing. In addition the transmission equipment need be no more complex than a standard Touch-Tone telephone instrument, so that the system is ideally suited for manual interrogation and data entry systems using a large number of remote data entry stations. The principal disadvantage is the high cost of the complex signal extraction equipment and the low operating speeds.

3.3 Error Detection and Correction

A continuously valued waveform propagated over a transmission link such as a telephone circuit is polluted, expectedly, by various noise processes. Some of these noise processes are generated by thermal agitation in the various resistances in the circuitry and in shot-effects in the various amplifying devices. Other noise processes are generated by inductive interference from power equipment and transmission lines and by crosstalk and other interchannel coupling means in multiple-channel repeater systems. Signal degradations due to these processes are the primary determinants of performance in speech transmission systems which are characterized by the usual audio-fidelity criteria.

In data transmission systems designed for the propagation of encoded discrete-valued waveforms, noise processes of various types result in an increasing number of transmission errors. However, the design parameters leading to a well-behaved data transmission circuit are definitely not those leading to a well-behaved speech transmission circuit. For, in these cases, stepwise discontinuities in amplitude and phase transfer functions and impulse-noise processes highly degrade data transmission without materially affecting speech transmission. All of these parameters are affected by switching paths within the telephone network and can occur in the course of a particular dial-up connection due either to fading in microwave systems or to coupling processes within the telephone plant. In its most disruptive form, the impulsive noise process appears as a high-amplitude spike superimposed on the data signal at the receiver. The effect of such a spike, often called an "impulse-hit," is to disrupt a few bits in the received data stream, that is cause a burst error.

Comprehensive surveys [1, 17, 27, 38, 43] have indicated that the expected error rate is in the 10^{-5} range, that is one bit error per 10^5 transmitted bits. This figure depends of course upon the modulation method and transmission rate. In a study involving the 201A data set operating at 2000 baud, the error rate was 3.19×10^{-5} [43]. In a study involving a 202-type data set 55 percent of the calls placed at 1200 baud had error rates greater than this, but only 30 percent of the calls placed at 600 baud had error rates greater than this [1, 27].

Even when a carefully designed modem is used on a noise-corrupted transmission circuit, a considerable number of residual transmission errors can be expected, especially in high-performance systems. In order to diminish the effect of transmission errors the serial binary stream can be encoded with redundancy bits which allow the receiver to detect when an error has occurred and even to correct the error. Such encoding techniques decrease the net transmission rate of the link of course due to the inclusion of the redundancy bits. The efficiency of a particular encoding technique then can be defined as its power to detect and correct transmission errors per redundancy bit required. A well-known theorem due to Shannon postulates that it is possible to encode in such a way as to achieve an arbitrarily low number of errors at a net transmission rate arbitrarily close to the channel capacity as defined in the preceding section. Unfortunately, the theoretical limits have not even been approached in practical codes.

A great number of techniques have been developed in recent years with which messages can be encoded together with redundancy bits at the transmitter so that transmission errors can be detected and even corrected at the receiver. These techniques all reduce the transmission rate by the number of redundancy bits added and so a tradeoff between error rate and data rate is created. To illustrate this point, consider a closer look at the encoding/decoding process.

Let the length of a message incident at the transmitter be k bits, and the length of the message encoded on the link be n bits. In the coding process the transmiter has added n-k bits of redundancy information which the receiver can use to detect and optionally to correct the transmission errors. Now, each of the encoded messages, or codewords, is n bits long; and there are 2**k of these which can legitimately be transmitted. The receiver however could conceivably recognize as many as 2**n messages in the event of errors. The difference between the 2**n possible received messages and the 2**k possible transmitted codewords represents the number of error messages. The encoding and decoding scheme should obviously result in the detection of as many of these as possible.

The power of an encoding scheme to detect and correct errors is measured as the number of bits in one codeword that must be changed to convert it into another. The minimum of this quantity over all codewords in the code is called the distance of the code. If a code has a distance of one, then a single bit-error will convert one codeword into another; and the receiver will have no opportunity to detect this as an error. If a code has a distance of two, then any single bit-error can be detected by the receiver but two bit-errors might not. Finally, if a code has a distance of three, then any single bit-error can be corrected by the receiver, but two bit-errors cannot. Extrapolating, if the code is to detect b errors and correct m errors, then the code must have a distance at least 2m+b+1 [32].

The efficiency of a particular code n bits in length can be defined as the ratio of the distance d to the number of redundancy bits n-k required. A good deal of research into the mathematical theory of codes has led to the establishment of upper bounds on efficiencies and the development of practical codes useful in the design of transmission equipment [33].

3. The Communication Channel

One of the most useful of the recent developments has been the class of linear cyclic codes based on polynomials [32, 33]. Briefly, a linear code is a particular subset of the set of all those 2**n words indicated above. The codewords of this subset are required to satisfy certain combinatorial requirements so that the subset is formally a vector space. In addition, if given a particular codeword in this space, every cyclic shift of it is also a codeword, then the vector space is called a cyclic vector space, and the code is called a cyclic code. This construction vastly simplifies the encoding and decoding operations.

Given a message of k bits and a codeword of n bits, a burst error is a sequence of bit-errors. The length of a burst error is the number of bits between and including the first and last bit in error. A cyclic code can detect all burst errors of length n-k or less. In addition, the code can detect an overwhelming percentage of bursts of greater lengths. In [42] a study was made using a (31,21) cyclic code in a transmission link operating at 2000 baud. In this study 6.36×10^7 code words were transmitted and 29,731 of these were received in error. The cyclic code was effective in detecting all but two of these errors.

An interpretation more useful in equipment design assigns a polynomial p(x) in the undetermined variable x to each k-bit word as follows: each of the k coefficients in a polynomial of degree k-1 is associated with a bit of a word of length k, so if a word were {00010110} then the polynomial associated would be:

$$0x^0 + 0x^1 + 0x^2 + 1x^3 + 0x^4 + 1x^5 + 1x^6 + 0x^7.$$

Now consider the polynomial (x**n)-1. Each of the divisors of this polynomial can be shown to generate a cyclic code in the following fashion: if g(x) (called the generator polynomial) is a divisor of (x**n)-1 and has degree n-k, then the codewords are all those polynomials f(x) of degree less than n which are are divisible by g(x), and the code is an (n,k) cyclic code of n bits, k of which are information bits and n-k of which are redundancy bits. The implementation of a cyclic code is now clear: the transmitter adds n-k redundancy bits to the k information bits in such a way that the codeword f(x) is exactly divisible by g(x). At the receiver this division is performed; and, if the remainder is zero after division, either no error has occurred or an undetected error has occurred. Otherwise, a transmission error has occurred.

The transmitter performs two functions: first an automatic premultiplication of the incident message p(x) by x**(n-k) and second a division process. Thus, let p(x) be the message polynomial of degree k. After premultiplication by x**(n-k) and division by g(x), the result can be represented by:

$$p(x)x{*}{*}(n-k) = g(x)q(x) + r(x),$$

where q(x) is the quotient after division and r(x) is the remainder, which has degree less than n-k. The codeword is then

$$f(x) = g(x)q(x) = p(x)x{*}{*}(n-k) - r(x)$$

and is transmitted as first p(x), transmitted high-order bit first, and then r(x), transmitted high-order bit first. By construction here, f(x) is divisible by g(x) as required and consists of k information digits followed by n-k redundancy check digits.

The mechanization of an (n,k) cyclic code is very simple and can be constructed from an n-k stage shift register and a collection of modulc-two adders ("exclusive-or" circuits). The connections of these components can be determined by an inspection of the generator polynomial [32]. Circuits are shown in Figure 3. These codes are used by IBM in connection with Binary Synchronous Communication devices. These components are connected so that:

1) The k digits of the codeword are developed in real-time; that is in step with the message digits; and

2) The n-k check digits are transmitted only following the message digits, so that the transmitted codeword consists of exactly k bits of the message as presented and followed by n-k check digits.

Connected in this manner messages of indefinite length can be encoded and decoded without intermediate temporary storage. The formal properties of the coding system are not altered by such a connection.

The operation of these devices is as follows: before transmission the shift register is cleared. During transmission the serial bit stream is transmitted and also fed at the point shown into the device, which is performing a continuous division process such that the remainder is in the shift register at every step. Following transmission the feedback path is broken and the bits in the shift register are shifted onto the line as is. Meanwhile at the receiver a corresponding process is going on in which the codeword is being divided by the generator polynomial.

After transmission of the entire codeword, the remainder left in the shift register had better be zero or an error has occurred.

```
                     transmitted data --+
                                        |
   +--+--+    +--+--+  +--+--+    +--+  V
+->|0 |1 |->+-|2 |3 |..|13|14|->+-|15|->+
|  +--+--+  A +--+--+  +--+--+  A +--+  |
|           |                   |       |
+-----------+<------------------+<------+
```

CRC-16 Checksum Computer

Polynomial: $1 + x^2 + x^{15} + x^{16}$

Prime Factors: $1 + x$, $1 + x + x^{15}$

```
                                  transmitted data --+
                                                     |
   +--+    +--+    +--+    +--+--+  +--+--+    +--+  V
+->|0 |->+-|1 |->+-|2 |->+-|3 |4 |..|9 |10|->+-|11|->+
|  +--+  A +--+  A +--+  A +--+--+  +--+--+  A +--+  |
|        |       |       |                   |       |
+--------+<------+<------+<------------------+<------+
```

CRC-12 Checksum Computer

Polynomial: $1 + x + x^2 + x^3 + x^{11} + x^{12}$

Prime Factors: $1 + x$, $1 + x^2 + x^{11}$

Figure 3. Mechanization of Cyclic Codes

4. TRANSMISSION CODES

The discussion in preceding sections has centered around methods of describing and evaluating the performance of serial data systems, that is, systems that transmit a clocked sequence of binary digits from one point to another. To be useful in conventional computing systems, means must be provided to serialize a character stream at the transmitting terminal and deserialize the bit stream at the receiving terminal. The manner in which these operations are synchronized forms a basis for the classification of transmission codes as the asynchronous and synchronous codes.

4.1 Asynchronous (Start/Stop) Codes

The asynchronous or start/stop codes are most often used at low-to-medium transmission rates with manually keyed terminal equipment. The transmitter maintains the transmission link in an 'on' or mark state when no data is to be transmitted. The transmission of each character is a separate operation and begins when the transmitter switches to the 'off' or space state for a specified start interval, which the receiver recognizes as an indication to start its deserializer clock. Following this interval, the bits of the character are transmitted as required. When the last bit has been transmitted, the transmitter holds the link in the mark state for a specified stop interval to allow receiving equipment to complete mechanical operations. A start/stop code can then be described in terms of the transmission rate for the bits of the character itself, together with the start and stop intervals. In most such codes the start and stop intervals are an integral number of bit-times (units) so that the code can be described by the number of units for the character itself (levels), the total number of units including the start and stop intervals, and the unit transmission rate in bits-per-second (baud). The start/stop transmission code used in the Model 35 Teletypewriter would be described as an eleven-unit eight-level code at a transmission rate of 110 baud, for example.

Start/stop transmission codes were often used in early teletypewriter networks, where the stop units were used to carry additional information in the following manner: In a multi-point network each machine was connected in series with the others so that any keyboard could cause printing at all stations. If any station detected that the line failed to return to mark at the stop interval, then an error or break condition was indicated. This condition by convention caused all keyboards to lock and was purposely used by an operator to clear the network for an overriding priority message. If the line failed to return to mark after a longer period, say a second or so, then a line-break condition was recognized and the repair crew was dispatched.

In modern teletypewriter equipment using frequency-shift modems, this operation is simulated by means of a carrier which operates at one frequency for mark and another for space. The usual equipment can operate in a full-duplex mode using a separate set of carrier frequencies for each transmission direction. The equipment contains circuitry to generate and detect the break signals and to initiate and respond to the longer space-disconnect signal.

In all presently used start/stop codes, the start interval is equal to one unit and in most codes the stop interval is from one to two units. The duration of a unit element in typical codes ranges from about 20 milliseconds to about 7 milliseconds and the number of levels from five to eight. Figure 4 shows the most popular of these including the USASCII code, which is standard for most AT&T teletypewriter equipment available at this time, and the PTTC code, which is standard for most IBM equipment. The code-unit numbering in these codes is as per the manufacturer's arbitrary designation and will become important in comparison of graphic codes and control functions in a later section.

4. Transmission Codes

```
|<-------------- 67 mS ---------------->|
|                                 ->|6.7|<-
|                                   |mS |
+---+---+---+---+---+---+---+---+---+   +---+
    | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |   |
    +---+---+---+---+---+---+---+---+---+
            (8 = even parity)
```

150 WPM Teletype Models 37/38 (150 baud)

USA Standard Code for Information Interchange (USASCII)

```
|<----------- 67 mS --------------->|
|                             ->|7.4|<-
|                               |mS |
+---+---+---+---+---+---+---+---+   +---+
    | C | 1 | 2 | 4 | 8 | A | B |   |
    +---+---+---+---+---+---+---+---+
            (C = odd parity)
```

IBM Models 2740/2741 and 1050 (134.5 baud)

Paper-Tape Transmission Code (PTTC)

```
|<--------------- 100 mS ------------------>|
|                                     ->|9.1|<-
|                                       |mS |
+---+---+---+---+---+---+---+---+---+---+   +---+
        | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |   |
        +---+---+---+---+---+---+---+---+---+
                (8 = even parity)
```

100 WPM Teletype Models 33/35 (110 baud)

USA Standard Code for Information Interchange (USASCII)

Figure 4. Start/Stop Transmission Codes

4.1 Synchronous Codes

In a start/stop code characters can be transmitted at a rate from zero up to the rated capacity of the link, and each character provides its own synchronizing information. If, on the other hand, the transmitter agrees to transmit all characters in a message one after the other in head-to-tail fashion, then the start and stop intervals can be discarded and the net transmission rate correspondingly increased. In such a case the receiver must gain synchronization with the transmitter before the message begins and remain in step for the duration of the message. Techniques like these are used with phase-modulated modem equipment in which two or more bits are clocked off and encoded into the transmission-line signal. Note that these techniques can be used equally as well on frequency-shift modem equipment if sufficient information is available at the receiver to recover the timing of the individual bits within a single character. Most phase-modulation modem equipment produces a timing signal incidental to the demodulation of the received data. In frequency-shift modem equipment this signal must be derived from the bit-transitions in the data stream itself.

A synchronous code can be described completely by a specification of the number of code units (levels) per character, the unit transmission rate (baud) and the bit pattern used for synchronizing purposes. Often-used transmission links operating on the switched network operate at 1200 to 2000 baud using eight-level codes interpreted as per USASCII or EBCDIC conventions (see Section 5). Such links are most often operated in the half-duplex mode and rather complicated conventions (described in Section 6) become necessary for line-control and error-recovery procedures.

A synchronous transmission link synchronizes character phase each time the transmitter begins operation and occasionally within the message itself. Synchronization is accomplished by transmitting one or more special synchronizing codes which can be detected by the receiver without a knowledge of character phase. When a synchronous receiver is waiting for receipt of a message, it remains in a special synchronization-hunt state searching for the synchronization code. When this code is detected, proper character phase has been achieved and the receiver switches to the text state in which characters are clocked off at the appropriate periods. Sometimes this character-phase acquisition process is complicated by special redundancy-checking procedures to minimize the effects of line transients.

Obviously, not all bit patterns for synchronizing codes are equally effective in noisy environments. The effectiveness of each bit pattern can be determined by an inspection of its autocorrelation function and its crosscorrelation functions with other noise signals. Each of these functions takes the form of a convolution integral which is periodic in time.

The most effective bit patterns are those with autocorrelation functions which take the form of a single sharply defined maximum value over the period and minimum values elsewhere. Furthermore, effective bit combinations will yield minimum-valued cross-correlation functions with all other bit combinations. Figure 5 shows these functions for several commonly used synchronizing codes, from which it is evident that the EBCDIC is the worst, at least by these criteria. A much better bit pattern than these can be formed from a pseudo-random sequence of period (2**n)-1 [33]. In such a sequence the synchronizing pattern would not necessarily be periodic at the same value as the number of levels per character, but this would complicate the hardware only slightly. The advantage of using such a sequence is that the correlation functions of a pseudo-random sequence are much better than those obtainable when the sequence is constrained to lengths other than (2**n)-1. Figure 5 shows the correlation functions for a pseudo-random sequence of length 7.

4. Transmission Codes

```
 |       *
 |      * *
 | *****   *****
 *               *
-+---+---+---+---+- t
-4  -2   0  +2  +4
```

USASCII code {00010110}
errors: 6(d=4), 1(d=6)

```
 |       *
 *      * *      *
 | *   *   *   *
 |   *       *
-+---+---+---+---+- t
-4  -2   0  +2  +4
```

EBCDIC code {00110010}
errors: 1(d=2), 2(d=4), 4(d=6)

```
 |       *
 |      * *
 ***   *   *   ***
 |   *       *
-+---+---+---+---+- t
-4  -2   0  +2  +4
```

IBM/STR code {00111001}
errors: 5(d=4), 2(d=6)

```
 |       *
 |      * *
 |     *   *
 |*****     *****
-+---+---+---+---+- t
-4  -2   0  +2  +4
```

Pseudo-Random code {0010111}
errors: 6(d=4)

Figure 5. Correlation Functions of Synchronizing Codes

4. Transmission Codes

5. INTERCHANGE CODES

An interchange code is an assignment or interpretation of a graphic symbol or control function to each code element of a transmission code. An interchange code differs from an error-detecting or error-correcting code in that transmission errors are not normally a factor in their design and differs from a transmission code in that synchronizing functions are not normally required. The design of an interchange code is usually predicated by a committee for the purpose of standardization and not for the purpose of transmission performance. In this section, the most popular interchange codes in current use will be described, together with some comments on expected frequency of occurrence and equivalences or mappings between them.

In discussing code structures a careful distinction should be drawn between the constructive aspects, that is, the number of levels, the parity conventions and so forth, and the interpretive aspects, that is, the printing graphic or control function assigned to a particular code element. Within any of the code structures described here, several different graphic interpretations may be assigned. These interpretations correspond to the choice of key-tops and printing elements in a particular terminal device, for example. Accordingly, given a certain code, that interpretation most appropriate for timesharing systems use will be emphasized most. It should also be understood that nothing in the code structure itself prohibits its use in either a stop/start or a synchronous environment, but that the use of a particular code in a start/stop environment requires a specification of the start and stop intervals and the use of a particular code in a synchronous environment requires a specification of the synchronizing character. Furthermore, a specification of a code need not, of course, necessarily indicate the transmission rate, although the most common use of a code may be at a specified rate. Finally, some conventions establish the numbering and ordering of the bits within the character reading in one direction and some in the other. IBM equipment is notoriously inconsistent in this regard. In this section all tables have been redrawn to the USASCII convention, which requires that bit transmission be "to the right."

In all important contemporary interchange codes the number of levels is either 6, 7 or 8. The graphic/control-function assigned to each of the code elements is shown in a table. Some interchange code specifications provide for a case-shift operation, and graphic/control-function assignments may differ depending upon which case-shift is in effect. The most popular interchange codes in contemporary timesharing systems use include the Extended Binary-Coded

Decimal Interchange Code (EBCDIC), the USA Society for Information Interchange (USASCII), and the Paper Tape Transmission Code (PTTC), although several others, including the Baudot, Binary Coded Decimal (BCD) and Synchronous Transmit/Receive (STR) codes are used occasionally. In following sections the EBCDIC, USASCII and PTTC codes will be described in some detail. The remaining codes are described in [I1].

5.1 The Extended Binary-Coded Decimal Code for Information Interchange (EBCDIC)

The EBCDIC [I1] is used extensively by IBM as the internal code of the System/360 machines and in almost all input/output devices attached directly to these machines. Communications equipment, however, most often operates with the USASCII or PTTC (described in subsequent sections). Since EBCDIC is an eight-level code and the others are of fewer levels, it should be possible to represent each of the others as a subset of EBCDIC; and IBM apparently made an effort to do this. In spite of the fact that the correspondences are inexact, EBCDIC appears the best common denominator in which the graphic/control-function assignments of the various interchange codes can be compared.

Figure 6 shows the most recent graphic/control-function assignments for EBCDIC. The 120 graphic assignments correspond to those of the TN print-train as used in the IBM 1403 line printer [I11]. The control functions are all indicated by a two-or-more character mnemonic. Some of these control functions are assigned to the EBCDIC-USASCII correspondence and some to the EBCDIC-PTTC correspondence. Table 2 lists each of these mnemonics separately, together with a short description of the control function. A more detailed description of these control functions is given separately by code in Sections 5.2 and 5.3 following.

The most common subset of EBCDIC used in System/360 operations is the 60-graphic subset indicated in Figure 7. This subset appears on the PN print train (among others), on the 029 card punch, and in other common equipment. Compare this subset with the graphic interpretations of other codes described subsequently.

0 1 2 3 4567	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
0000	NUL	DLE	DS		SP	&	-				-	^{0}				0
0001	SOH	DC1	SOS				/		a	j	°	^{1}	A	J		1
0010	STX	DC2	FS	SYN					b	k	s	^{2}	B	K	S	2
0011	ETX	TM							c	l	t	^{3}	C	L	T	3
0100	PF	RES	BYP	PN					d	m	u	^{4}	D	M	U	4
0101	HT	NL	LF	RS					e	n	v	^{5}	E	N	V	5
0110	LC	BS	ETB	UC					f	o	w	^{6}	F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x	^{7}	G	P	X	7
1000		CAN							h	q	y	^{8}	H	Q	Y	8
1001		EM							i	r	z	^{9}	I	R	Z	9
1010	SMM	CC	SM		¢	!		:								
1011	VT	CU1	CU2	CU3	.	$	,	#	{	}	└	┘				
1100	FF	IFS		DC4	<	*	%	@	≤	□	┌	┐				
1101	CR	IGS	ENQ	NAK	(	)	_	'	(	)	[	]				
1110	SO	IRS	ACK		+	;	>	=	+	±	≥	≠				
1111	SI	IUS	BEL	SUB	|	¬	?	"	┼	▪	•	—				

P	0	1	2	3	4	5	6	7

Code Structure
(P = odd parity)

Figure 6.

Extended Binary-Coded Decimal Interchange Code (EBCDIC)

5. Interchange Codes

CAT	EBCDIC	ASCII	TTY	PTTC	FUNCTION
	NUL	NUL	NULL		Null
CC	SOH	SOH	SOM		Start of Heading
CC	STX	STX	EOA	EOA (D)	Start of Text
CC	ETX	ETX	EOM		End of Text
DC	PF			PF	Punch Off
FE	HT	HT	H.TAB	TAB	Horizontal Tab
GR	LC			LC	Lower Case
	DEL	DEL	RUBOUT	DEL	Delete
	SMM				Start of Manual Message
FE	VT	VT	V.TAB		Vertical Tab
FE	FF	FF	FORM		Form Feed
FE	CR	CR	RETURN		Carriage Return
GR	SO	SO	SO		Shift Out
GR	SI	SI	SI		Shift In
CC	DLE	DLE	DC0		Data Link Escape
DC	DC1	DC1	X-ON		Device Control 1
DC	DC2	DC2	TAPE ON		Device Control 2
	TM				Tape Mark
DC	RES			RES	Restore
FE	NL			NL	New Line
FE	BS	BS		BS	Backspace
	IL			IL	Idle
	CAN	CAN	FEO	CAN	Cancel
	EM	EM	S1		End of Medium
	CC				Curser Control
CU	CU1				Customer Use 1
IS	IFS	FS	S4		Info. Field Separator
IS	IGS	GS	S5		Info. Group Separator
IS	IRS	RS	S6		Info. Record Separator
IS	IUS	US	S7		Info Unit Separator
ED	DS				Digit Select
ED	SOS				Start of Significance
ED	FS				Field Separator
DC	BYP			BYP	Bypass
FE	LF	LF	LF	LF	Line Feed
CC	ETB	ETB	LEM	EOB (B)	End of Text Block
	ESC	ESC	S3	PRE	Escape
	SM				Set Mode
CU	CU2				Customer Use 2
CC	ENQ	ENQ	WRU		Enquiry
CC	ACK	ACK	RU	(Y)	Acknowledge
	BEL	BEL	BELL		Bell
CC	SYN	SYN	SYNC		Synchronous Idle
DC	PN			PN	Punch On
DC	RS			RS	Reader Stop
GR	UC			UC	Upper Case
CC	EOT	EOT	EOT	EOT (C)	End of Transmission
CU	CU3				Customer Use 3

(continued)

DC	DC4	DC4	TAPE OFF	Device Control 4
CC	NAK	NAK	ERROR (N)	Negative Acknowledge
	SUB	SUB	S2	Substitute
DC		DC3	X-OFF	Device Control 3

Notes: the categories of characters are defined by the following:

CC (Communication Control). A functional character intended to control or facilitate transmission of information over communication networks.

FE (Format Effector). A functional character which controls the layout or positioning of information in printing or display devices.

IS (Information Separator). A character which is used to separate and qualify information in a logical sense. There is a group of four such characters, which are to be used in a hierarchical order.

DC (Device Control). A functional character used for the control of ancillary devices associated with data processing of telecommunication systems, more especially switching devices "on" and "off."

ED (Edit and Mark). A control character used by the System/360 Edit and Mark (EDMK) instruction for the formatting of alphanumeric fields.

GR (Graphic Control). A control character indicating that the code combinations which follow are to be interpreted in a particular code table, depending upon the particular control character.

CU (Customer Use). A character excluded from future assignment by IBM. These "protected" codes are intended for use by customer systems so that their use will not conflict with a possible future IBM use.

Table 2. Control Functions

0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
4567																
0000	NUL	DLE	DS		SP	&	-									0
0001	SOH	DC1	SOS				/						A	J		1
0010	STX	DC2	FS	SYN									B	K	S	2
0011	ETX	TM											C	L	T	3
0100	PF	RES	BYP	PN									D	M	U	4
0101	HT	NL	LF	RS									E	N	V	5
0110	LC	BS	ETB	UC									F	O	W	6
0111	DEL	IL	ESC	EOT									G	P	X	7
1000		CAN											H	Q	Y	8
1001		EM											I	R	Z	9
1010	SMM	CC	SM					:								
1011	VT	CU1	CU2	CU3	.	$	,	#								
1100	FF	IFS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	(	)	_	'								
1110	SO	IRS	ACK		+	;	>	=								
1111	SI	IUS	BEL	SUB	|	¬	?	"								

P	0	1	2	3	4	5	6	7

Code Structure
(P = odd parity)

Figure 7.

Extended Binary-Coded Decimal Interchange Code (EBCDIC)
(60-graphic subset)

5. Interchange Codes

5.2 The USA Society for Information Interchange Code (USASCII)

A case could quite certainly be made for the USASCII code to be the most widely used in modern man-machine communications systems. Perhaps the most cogent reason for this is the wide availability of relatively inexpensive terminal equipment available from AT&T for lease or purchase.

The USASCII [11, 34, 35, 36, 37, 44] is an eight-level code with the eighth level optionally assigned as an even-parity check bit. The most common use of this code is in an eleven-unit start/stop transmission code. In such uses, usually with terminal equipment manufactured by the Teletype Corporation, the start code element is one unit in length and the stop code element is two units in length. The resultant eleven-unit code is transmitted at 110 baud or 10 characters-per-second. This code is also widely used in synchronous systems.

The graphic/control-function assignments of the USASCII code has been well standardized [35] and is shown in Figure 8. Some of the older equipment does not conform exactly to this standard, in particular to the printing graphics for carat (^) and underscore (_), which were up-arrow (↑) and back-arrow (←) respectively on this equipment. Also, the names for some of the control characters have only recently become standard. Note that the full 94-character graphic assignments are implemented only on the latest equipment; only the 64-character subset assigned through the upper-case letters and the five special characters left bracket ([), right bracket (]), backslash (\), carat (^) and underscore (_) are available on older equipment.

The mnemonics assigned to certain control functions have been changed recently. The new USASCII assignments are shown under the "USASCII" column in Table 2, while the old assignments are shown under the "TTY" column.

Following are the standard interpretations of the USASCII control functions (from [35]):

NUL (Null). The all-zeros character which may serve to accomplish time fill and media fill.

SOH (Start of Heading). A communication control character used at the beginning of a sequence of characters which constitute a machine-sensible address or routing information. Such a sequence is referred to as the heading. An STX character has the effect of terminating a heading.

STX	(Start of Text). A communication control character which precedes a sequence of characters that is to be treated as an entity and transmitted through to the ultimate destination. Such a sequence is referred to as <u>text.</u> STX may be used to terminate a sequence of characters started by SOH.
ETX	(End of Text). A communication control character used to terminate a sequence of characters started with STX and transmitted as an entity.
EOT	(End of Transmission). A communication control character used to indicate the conclusion of a transmission, which may have contained one or more texts and any associated headings.
ENQ	(Enquiry). A communication control character used in data communication systems as a request for a response from a remote station. It may be used as a "Who Are You" (WRU) to obtain identification, or may be used to obtain station status, or both.
ACK	(Acknowledge). A communication control character transmitted by a receiver as an affirmative response to a sender.
BEL	(Bell). A character for use when there is a need to call for human attention. It may control alarm or attention devices.
BS	(Backspace). A format effector which controls the movement of the printing position one printing space backward on the same printing line (applicable also to display devices).
HT	(Horizontal Tabulation). A format effector which controls the movement of the printing position to the next in a series of predetermined positions along the printing line (applicable also to display devices and the skip function on punched cards).
LF	(Line Feed). A format effector which controls the movement of the printing position to the next printing line (also applicable to display devices).
VT	(Vertical Tabulation). A format effector which controls the movement of the printing position to the next in a series of predetermined printing lines (also applicable to display devices).
FF	(Form Feed). A format effector which controls the movement of the printing position to the first

predetermined printing line on the next form or page (also applicable to display devices).

CR (Carriage Return). A format effector which controls the movement of the printing position to the first printing position on the same printing line (also applicable to display devices).

SO (Shift Out). A control character indicating that the code combinations which follow shall be interpreted as outside of the character set of the standard code table until a Shift In character is reached.

SI (Shift In). A control character indicating that the code combinations which follow shall be interpreted according to the standard code table.

DLE (Data Link Escape). A communication control character which will change the meaning of a limited number of contiguously following characters. It is used exclusively to provide supplementary controls in data communication networks.

DC1, DC2, DC3, DC4 (Device Controls). Characters for the control of ancillary devices associated with data processing or telecommunication systems, more especially switching devices "on" and "off." (If a single "stop" control is required to interrupt or turn off ancillary devices, DC4 is the preferred assignment.)

NAK (Negative Acknowledge). A communication control character transmitted by a receiver as a negative response to a sender.

SYN (Synchronous Idle). A communication control character used by a synchronous transmission system in the absence of any other character to provide a signal from which synchronism may be achieved or retained.

ETB (End of Transmission Block). A communication control character used to indicate the end of a block of data for communication purposes. ETB is used for blocking data where the block structure is not necessarily related to the processing format.

CAN (Cancel). A control character used to indicate that the data with which it is sent is in error or is to be disregarded.

EM (End of Medium). A control character associated with the sent data which may be used to identify the physical end of the medium, or the end of the used, or wanted, portion of information recorded on a medium. (The position of this character does not necessarily correspond to the physical end of the medium.)

SS	(Start of Special Sequence). A control character used to indicate the start of a variable length sequence of characters which have special significance or which are to have special handling.
ESC	(Escape). A control character intended to provide code extension (supplementary characters) in general information interchange. The Escape character itself is a prefix affecting the interpretation of a limited number of contiguously following characters.
FS	(File Separator), GS (Group Separator), RS (Record Separator) and US (Unit Separator). These information separators may be used within data in optional fashion, except that the hierarchical relationship shall be: FS is the most inclusive, then GS, then RS, and US is least inclusive. (The content and length of a File, Group, Record or Unit are not specified.)
DEL	(Delete). This character is used primarily to "erase" or "obliterate" erroneous or unwanted characters in perforated tape. (In the strict sense, DEL is not a control character.)

5. Interchange Codes

7	0	0	0	0	1	1	1	1
6	0	0	1	1	0	0	1	1
5	0	1	0	1	0	1	0	1
4321								
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	¬
1111	SI	US	/	?	O	_	o	DEL

8	7	6	5	4	3	2	1	

Code Structure
(8 = even parity)

Figure 8.

USA Standard Code for Information Interchange (USASCII)

5. Interchange Codes

5.3 Paper Tape Transmission Code (PTTC)

The PTTC [I1, I6-I9] is used by IBM in typewriter terminals of the Selectric type, such as the 2741 Communications Terminal, and in data-collection and communications equipment, such as 1050 Data Communications System. The PTTC code is a seven-level nine-unit code including an odd-parity check bit and a single stop bit. It is most commonly used at 134.5 baud (14.8 characters-per-second), although speeds to 600 baud are used in some data collection equipment. All of the IBM terminal equipment uses case-shift codes, so that potentially 126 characters are available in this code, although many of these by convention have the same interpretation in either case.

Several interpretations of PTTC are common in connection with IBM computing systems of the 1400 and 7000 class. The one most useful in large-scale timesharing utilities is the EBCDIC-oriented variant, shown in Figure 9. Note that the full 60-graphic PN-train EBCDIC subset mentioned above is available, in addition to control functions for device addressing, polling, and selection. In Figure 9 the control characters have been renamed to be consistent with the EBCDIC names assigned in Table 2 and are described below:

PN	(Punch On). A device control character used to start the 1054 Paper Tape Punch when attached to the 1050 Data Communications System.
RES	(Restore). A device control character which causes the 1053 Printer when attached to the 1050 Data Communications System to resume printing following a BYP (Bypass) character.
BYP	(Bypass). A device control character which causes the 1053 Printer when attached to the 1050 Data Communications System to suspend printing until an RES (Restore) character is reached.
PF	(Punch Off). A device control character used to stop the 1055 Paper Tape Punch when attached to the 1050 Data Communications System.
UC	(Upper Case), LC (Lower Case). Control characters used to indicate that the code combinations which follow are to be interpreted according to selected portions of the code table (see Figure 9).
BS	(Backspace). A format effector which controls the movement of the printing position one printing space backward on the same printing line.

ETB	(End of Text Block - also called EOB, circle-B). A communication control character used in the 1050 Data Communication System to end a record containing text information. This character is always followed by the LRC (Longitudinal Redundancy Check) character.
RS	(Reader Stop). A device control character used to stop the 1056 Card Reader when attached to the 1050 Data Communications System.
NL	(New Line). A format effector which controls the movement of the printing position to the first printing position on the next printing line. (This character causes the same action as a combination CR (Carriage Return) and LF (Line Feed) sequence - see Section 5.2.)
LF	(Line Feed). A format effector which controls the movement of the printing position to the next printing line.
HT	(Horizontal Tabulation). A format effector which controls the movement of the printing position to the next in a series of predetermined positions along the printing line.
EOT	(End of Transmission - also called circle-C). A communication control character which causes the device and its control unit to be reset and to revert to the control-receive state following a message transmission sequence. This character is used in the 1050 Data Communications System before polling and addressing operations and in the 2740/2741 Communications Terminal following the NL (New Line) character at the end of each text record.
IL	(Idle). A character which serves to accomplish time fill and media fill. (This character is used in the same manner as the NUL character - see Section 5.2 - but because of an unfortunate choice of the bit combination used to represent the SP (Space) character, the all-zeros character is not available for this function.)
ESC	(Escape - also called PRE). A control character used by the various components of the 1050 Data Communications System to precede device control characters interpreted specially by the device for format-effecting and device control functions.
DEL	(Delete). This character is used primarily to "erase" or "obliterate" erroneous or unwanted characters in perforated tape. (In the strict sense, DEL is not a

control character.)

CAN	(Cancel). A control character used to indicate that the data with which it is sent is in error or is to be disregarded. (This character does not exist in the standard code table and is implemented as an even parity (rather than odd parity) code which the receiver recognizes by virtue of the parity error (!).)
STX	(Start of Text - also called EOA, circle-D). A communication control character used as an affirmative reply to a polling sequence. Transmission of this character indicates that the transmitter has more data to follow and that this data will be terminated with an ETB-LRC sequence.
ACK	(Acknowledge - also called circle-Y). A communication control character used in the 1050 Data Communications System in two ways: 1) As an affirmative reply to an addressing sequence indicating that the addressed device is ready to receive data, and 2) As an affirmative reply to an ETB-LRC sequence indicating that the preceding record was received without apparent error.
NAK	(Negative Acknowledge - also called circle-N). A communication control character used in the 1050 Data Communications System in two ways: 1) As a negative reply to a polling or addressing sequence indicating the polled or addressed device is not ready to receive or transmit data, and 2) As a negative reply to an ETB-LRC sequence indicating that the preceding record was received in error.

	UPPER CASE				LOWER CASE				CONTROL			
1	0	0	1	1	0	0	1	1	0	0	1	1
2	0	1	0	1	0	1	0	1	0	1	0	1
48AB												
0000	SP	2	1	3	SP	<	=	;				
0001	-	k	j	l	_	K	J	L	NAK			
0010	@	s	/	t	¢	S	?	T				
0011	&	b	a	c	+	B	A	C				
0100	8	0	9	#	*	)	(	"				STX
0101	q		r	$	Q		R	!				
0110	y		z	,	Y		Z	|				
0111	h		i	.	H		I	¬				ACK
1000	4	6	5	7	:	'	%	>				
1001	m	o	n	p	M	O	N	P				
1010	u	w	v	x	U	W	V	X				
1011	d	f	e	g	D	F	E	G				
1100	PN	UC	RS	EOT	PN	UC	RS	EOT				EOT
1101	RES	BS	NL	IL	RES	BS	NL	IL				
1110	BYP	ETB	LF	ESC	BYP	ETB	LF	ESC				
1111	PF	LC	HT	DEL	PF	LC	HT	DEL				

C	1	2	4	8	A	B	

Code Structure
(C = odd parity)

Figure 9.

Paper-Tape Transmission Code (PTTC)

5.4 Codes for Remote Job Entry

In cases where the speed of the transmission link is the limiting factor in terminal performance, a particular code such as USASCII or PTTC may need to be recoded for transmission. The most acute need for such techniques is in connection with Remote Job Entry (RJE) equipment where the volume of data is large and where the remote equipment is of a high-performance type. It is of interest here to gain some insight into the expected distribution of transmitted character frequencies in such an operation and to compare some simple recoding schemes.

Table 3 shows the expected incidence of blanks, letters, digits, special graphics and other characters typical in several categories of RJE traffic. Note the overwhelming incidence of blanks in the program source and object listing categories. As expected, the distribution is more uniform in object deck categories.

Table 3 also indicates the results of two simple recoding procedures when applied to traffic in each category. In the first procedure, runs of four or more characters of any type are deleted at a cost of three characters per run and two characters per line. In the second procedure, runs of three or more blanks are deleted at a cost of two characters per run and two characters per line. These recoding costs represent an upper bound and can be reduced with clever finagling. The conclusion here is that the former or blank-recoding technique is best.

The most obvious factor in RJE operation is the overwhelming incidence of letters and digits. It seems that a very strong case exists to transmit this material using a six-level code together with some kind of escape-character convention. The last column of Table 3 shows the effect of combining the blank-recoding technique with the escape-character technique. Using these techniques the average line length in RJE terminal-to-processor operations is equivalent to about 20 eight-bit characters; while the average line length in processor-to-RJE terminal operations is equivalent to about 36 eight-bit characters. Using a 2000-baud transmission link then, and neglecting turnaround and acknowledgment times, the RJE terminal performance can approach 750 cards-per-minute in processor-inbound operations and 416 lines-per-minute in processor-outbound operations.

Text	Distribution (percent) Bl	Let	Dig	Spec Graf	Oth	Line Length (char) No Cod	Arb Cod	Bl Cod	Esc Cod
360 asm sou	50	38	4	7	0	31	24	23	17
360 asm lst	69	12	16	2	<1	114	52	49	38
360 obj	20	15	6	2	56	80	67	72	121
PDP asm sou	53	37	4	5	0	35	25	24	18
PDP asm lst	57	17	22	3	0	70	46	43	32
PDP asm obj	<1	<1	<1	<1	99	183	182	185	277
PL-1 sou	40	41	2	16	0	38	29	29	22
PL-1 lst	64	17	13	4	1	103	54	52	39
SNOBOL sou	43	39	2	15	0	39	31	31	23

Table 3. Comparison of Remote Job Entry Codes

5. Interchange Codes

6. PROCESSOR-PROCESSOR COMMUNICATIONS

Message transmission between two processors involves a great many peculiar factors not found in message transmission between a processor and a conventional typewriter terminal. In processor-processor communication error control procedures must be able to recover from routine circuit faults without operator intervention. Also, transmission rates and turnaround times are usually limited by the modem equipment itself, rather than by human operator response times. Control functions in processor-processor communication systems are performed by special characters in the transmitted data stream; and special encoding conventions are necessary to allow transmission of all character codes in text messages. The basic premiss underlying all processor-processor communications systems is the necessity for an acknowledgment or an a assurance generated by the recipient in any transaction that the message transmission was in fact successful. If the transmission was successful then message traffic can continue; while, if not, the message must be retransmitted. It is convenient to describe a system of this nature in terms of a symmetric half-duplex system in which either terminal can transmit serial character streams to the other, but not both at the same time. In the following sections, several problem areas will be discussed with particular emphasis on those common to the timesharing environment. It will be apparent in this discussion that the problems are not restricted to message transmission using modems and telephone lines at all, but can be readily applied to any communication system in which records and status responses must be transmitted over a basically one-way-at-a-time medium. For instance, the communication between processor channels and control units are subject to those same problems.

A symmetric half-duplex system consists of at least two terminals connected by a bi-directional transmission medium. Only one terminal may transmit at any time and, if the message is to be acknowledged, the message must be directed to a particular terminal designated as the receiver. If any terminal begins to transmit then all others recognize the fact after a period of ambiguity equal to the propagation delay. If a message requiring an acknowledgment is directed to a particular receiver, then that terminal must transmit the acknowledgment immediately following reception of the message. With this exception, no terminal may transmit following a previous message within an interval equal to twice the propagation delay. No terminal may transmit if it is receiving a message, even if the message is directed to another receiver. Systems such as this are described in [2, 12, 14, 18, 24, 25, 37 and 44].

A system including only two terminals of this type is called a point-to-point system; a system including more than two terminals is called a multipoint system. A single terminal may accept a number of messages from different logical source devices and deliver a number of messages to different logical sink devices. A particular terminal may find all of its transmission facilities busy, in which case it must respond to any logical source device request with a busy sequence. This sequence must be distinguished from another which indicates that the logical sink device addressed by the message is busy but the terminal itself is not.

Control over the system operation can be assigned in various ways. If a single master station is assigned to coordinate message traffic between the various terminals on a polling/addressing basis, then the operation is described as centralized. If permission to transmit is assigned to a terminal by a bidding or tie-breaking procedure, then the operation is described as contention with bidding. If each terminal assumes permission to transmit as necessary but conditioned by certain rules designed to minimize interference, then the operation is described as contention without bidding. The systems considered herein all operate as contention systems, both with and without bidding. Those systems restricted to bidding operation behave very much like other proposed contemporary systems [14, 37]. The most interesting systems, however, are those not restricted to bidding operation, and these will be the subject of the closest scrutiny here.

Now we can interpret this general description in several ways. A collection of communication devices and terminals in a complex store-and-forward network forms the most obvious model. Here the connecting half-duplex lines represent the transmission links and the communications devices represent the terminals. In particular, devices such as card readers and punches and line printers may be attached to a terminal, and these may be capable of overlapped operation. Furthermore, the same device may be reached via two or more half-duplex circuits. In an alternate interpretation, consider a collection of selector channels, control units and input/output devices. Under this interpretation the channels represent the transmission links, the control units represent the terminals, and the devices represent the logical source and sink devices. Note that this description provides for pathfinding mechanisms where a single device may be reached by a number of channels and control units. It is clear that a single analysis should suffice for either of these cases.

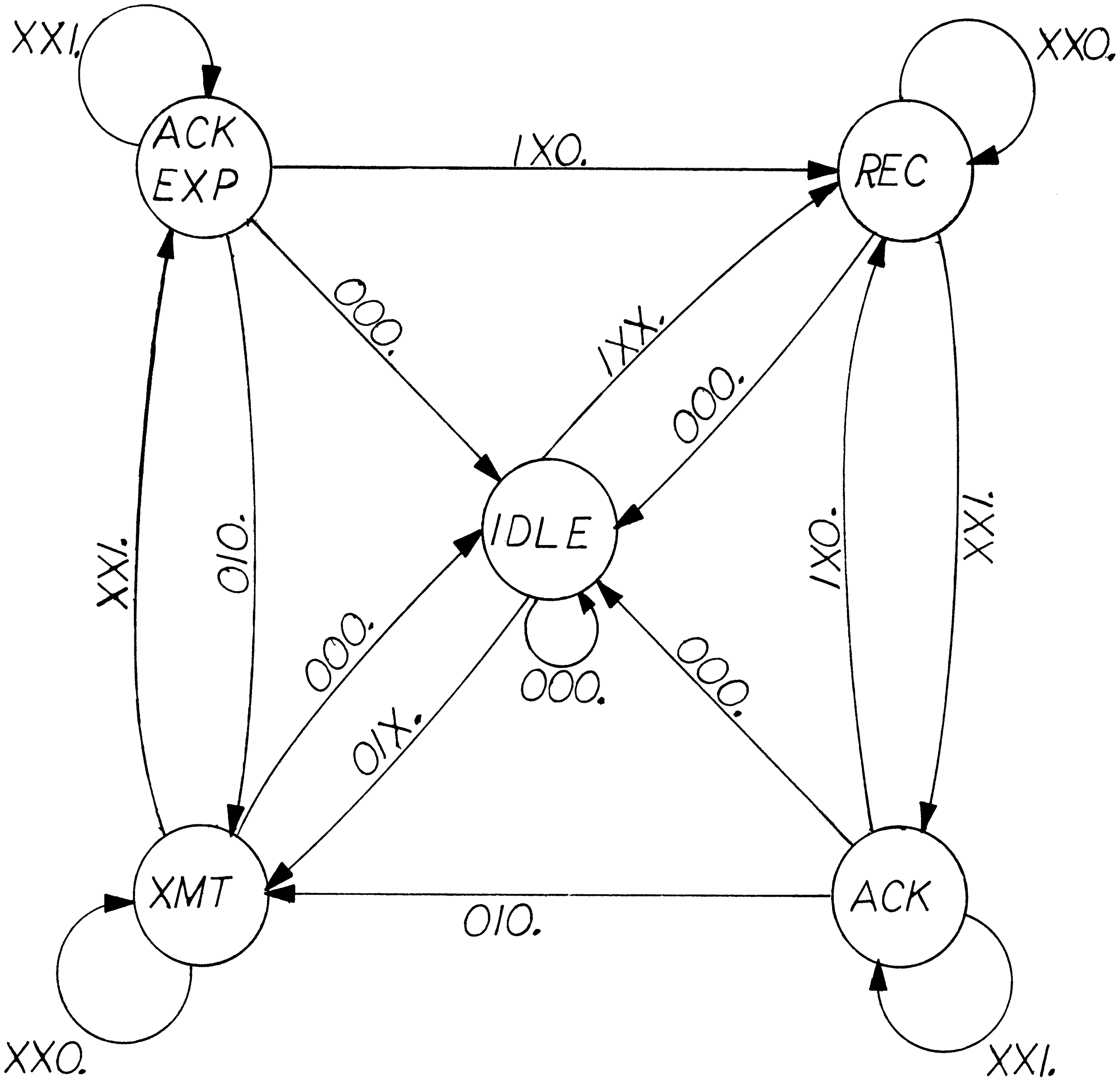
6. Processor-Processor Communications

6.1 Modeling the Half-Duplex System

In this section, a simple model will be developed from which many of the properties of the half-duplex system are evident. In this model such a system will be assumed to consist of two terminals of identical characteristics and arbitrarily designated A and B. Each logical source attached to a terminal is presumed to generate in a stochastic fashion messages of varying length intended for a logical sink attached to the conjugal terminal. It will not be important to establish blocking conventions, checksum and control-character codes and other message-dependent factors at this time.

The simplest useful half-duplex terminal contains a single data buffer, which may be used for transmitted records or received records but not both at the same time. The various control characters used for checksum, acknowledgment and enquiry functions are assumed to be processed by the transmission hardware or software directly and are not necessarily involved in buffer-transfer operations. This type of organization is typical of hard-logic terminal systems, such as the IBM 2780 Data Transmission Terminal [I10]. We will be interested here primarily in a slightly more complex system in which every logical source and sink attached to a terminal can be assigned a buffer dynamically as required. Each of these buffers may contain a single message, which is defined as a sequence of characters delimited in a suitable fashion. Several messages from different logical sources may be concatenated into a single record, which is defined as a sequence of messages transmitted in a single operation.

The behavior of such a system can be visualized with the aid of Figure 10, which shows the states that each terminal can be in at any particular time. Two types of transmission records can be recognized: those that require an acknowledgment to be returned from the receiving terminal and those that do not. Typically, but not necessarily, messages involving text are ended with a checksum sequence which requires a positive or negative acknowledgment. Messages not involving text, such as certain control functions and the acknowledgments themselves, do not require acknowledgment. These two classes are not exclusive; and it may happen that a message requiring an acknowledgment is included in a record with a message that does not.



Input digit code: abc.
    a REC Receive status
    b XMT Transmit request
    c ACK EXP Acknowledgment expected

Figure 10. States of the Half-Duplex System

In Figure 10 three signals may be identified as driving functions during state transitions:

REC (Receive Status). A signal determined from an inspection of the data set control leads and certain timeout criteria (see below and Section 7.4). This signal is unambiguously determined only in certain states. If the data set is in receive status then the signal has value 1, otherwise it has value 0.

XMT (Transmit Request). A signal determined from the status of a queue of messages waiting to be transmitted. If this queue is nonempty then the signal has value 1, otherwise it has value 0.

ACK EXP (Acknowledgment Expected). A signal determined from the nature of the transmitted or received message by means of some format criterion. Once the end of a record containing a message to be acknowledgd is detected, the signal has value 1; and, once the acknowledgement is transmitted, the signal has value 0.

In general, while a terminal is transmitting a message which will require an acknowledgment, it is in the transmit state, and, following transmission, is in the acknowledgment-expected state. In the acknowledgment-expected state, the terminal can transmit no messages at all except the enquiry message (see below). When the acknowledgment is received, the terminal may begin immediately to transmit its next record, if one is pending, or to go to the idle state. If a message is received at any time, in either the idle state or the acknowledgment-expected state, the terminal switches to the receive state. Following the received message the terminal switches to the acknowledge state and transmits the acknowledgment. Following the acknowledgment state, if the terminal has traffic to transmit, it may transmit it at this time; if not, the terminal enters the idle state. With this organization, if traffic is always waiting to be transmitted by either terminal, then both terminals are always in a non-idle state and the transmission direction changes in a flip-flop-like manner.

Now, once either terminal enters the idle state because of a temporary lull in traffic, a condition can occur in which both terminals may attempt to transmit simultaneously, in which case either or both transmissions may be lost by the intended receiver. Such situations, commonly called line-contention situations, can be expected routinely although infrequently in well-designed systems. Transmission protocols discussed here are designed to provide recovery procedures in these and other situations.

6. Processor-Processor Communications

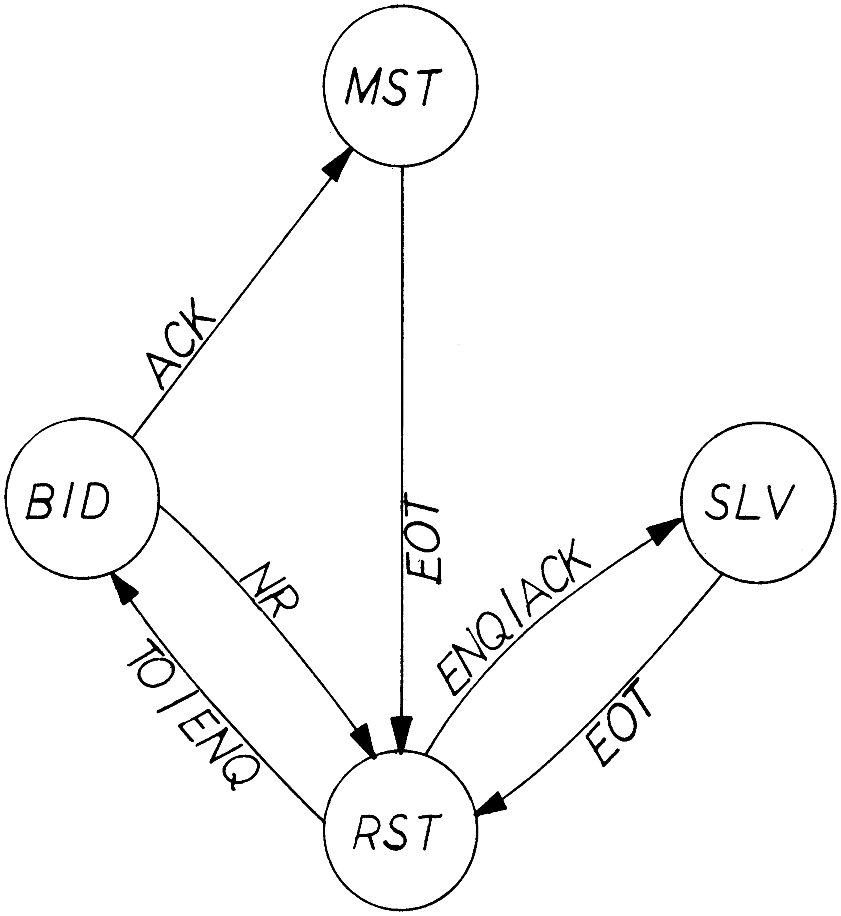
The next section presents an analysis of the expected incidence of contention conditions.

Permission to begin transmission of a message can be assumed in either of two ways. If permission is assumed as in Figure 10 with no additional interlocks, the operation is as a contention-without-bidding system. Alternatively, a specific bidding procedure can be executed which assigns to only one terminal permission to transmit. The bidding procedure uses a tie-breaking sequence such as shown in Figure 11. Assuming both terminals start in the RST state, then one or both generate an ENQ character, which is interpreted by the intended receiver as a request to assume slave status. The receiver, upon assuming slave status, generates an ACK character as a reply. The transmitter, upon receiving the ACK character, assumes master status. These states are maintained until either terminal indicates as part of its message or acknowledgment traffic an EOT character, which causes both terminals to revert to the RST state.

A terminal assuming either master or slave status can transmit traffic as indicated in Figure 10 with the following exception: a slave terminal cannot take the transition from the IDLE state to the XMT state. Thus the only terminal that can begin to transmit a message, once both terminals have entered the IDLE state, is the master.

Half-duplex operation requires terminal equipment control over whether the attached data set is transmitting or receiving. Control over these modes is indicated by the terminal using the Request to Send (REQ SND) control lead which when raised causes the data set to begin transmission. It is important to realize that, once a transmission has been initiated on a half-duplex data set, any transmissions made by the opposite terminal are lost. Some data sets provide a "local copy" at the receiver when transmitting, but this is simply a monitor of the local transmitter data stream and reflects the data exactly as sent for verification purposes.

6. Processor-Processor Communications



Note:
NR: Anything except ACK, including TO.
TO: Timeout.

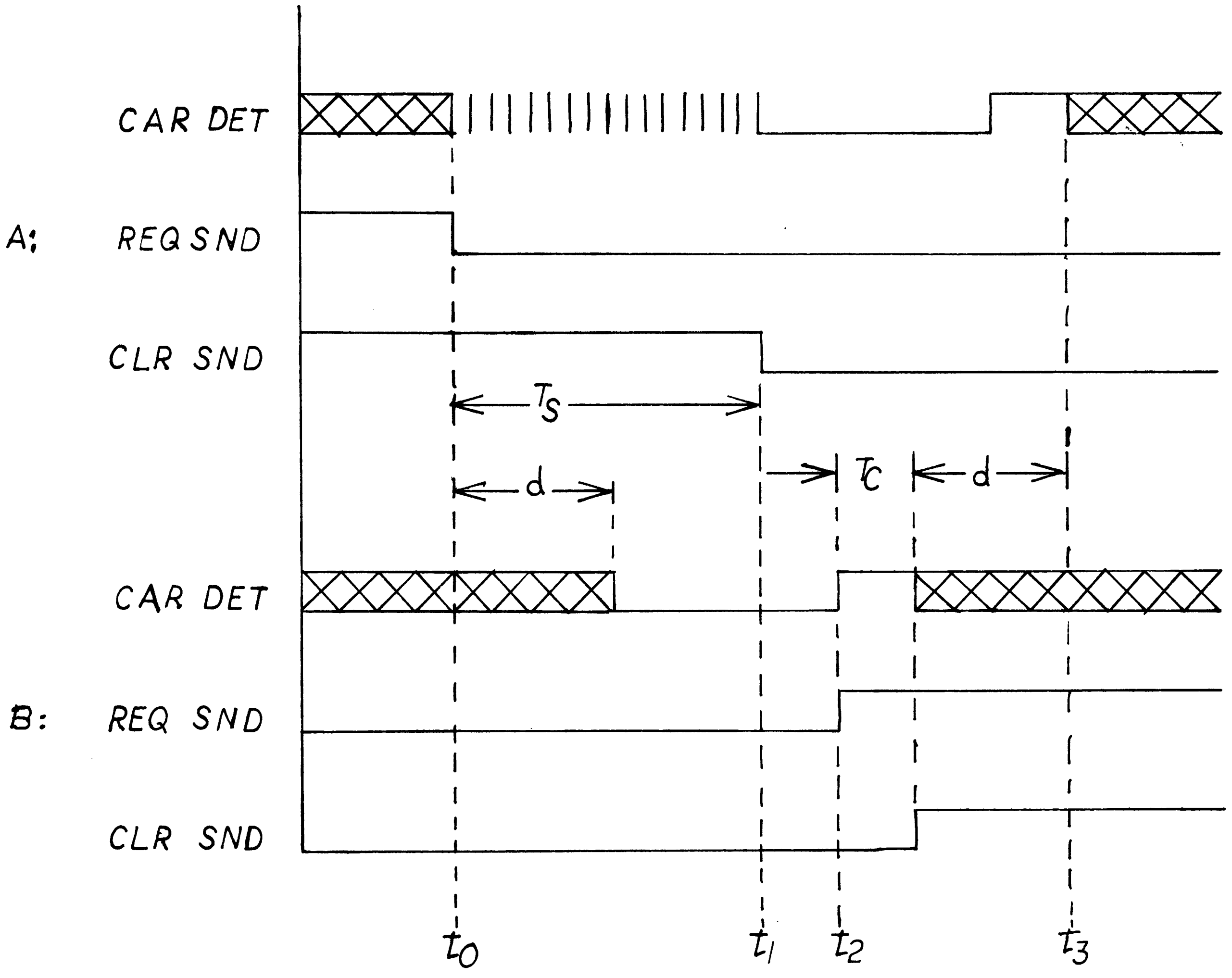
Figure 11. Bidding Operations

Figure 12 shows the signal sequences involved in line turnaround or change of direction. Once the REQ SND lead is raised at the data set an internal timeout is set within the data set so that, after a fixed interval, the Clear to Send (CLR SND) lead is raised. This is interpreted as permission to begin data transmission. This interval, usually a couple of hundred milliseconds, is provided to allow for echo-suppressing equipment within the switched network to change direction. The time-delay mechanism is entirely within the data set itself, however, and does not condition any transmission or receiving components. Therefore, it can be interpreted as advisory only and, if it is known that no echo-suppressing equipment is incorporated in a particular call, the CLR SND signal can be ignored. Instead a logical "Clear to Send" can be developed in the following fashion:

In Figure 12 terminal A begins a line turnaround sequence when it drops the REQ SND lead at t(0), causing the transmitter carrier oscillator to be quenched. Terminal B recognizes the fall of the Carrier Detector (CAR DET) signal following a propagation delay d (about 100 microseconds-per-mile); however, reflections of the immediately preceding carrier signal may persist for a period estimated at twice the propagation delay. In order to mask itself against reception of its own echoes, terminal A initiates a squelch delay T(s), as shown, during which the echoes are quenched and echo-suppressing equipment within the telephone network changes state. Terminal B begins to transmit carrier at t(2) when it raises its REQ SND lead. Data transmission must be delayed for a clamp delay T(d) however, to allow terminal A to synchronize its receiver circuits and for compander equipment within the telephone network to stabilize.

From this elementary analysis it is apparent that the startup time for terminal B's transmitter can be overlapped with the shutdown time for terminal A's transmitter so that the minimum turnaround time referred to terminal A should be 2d+T(c), provided that A's squelch delay is carefully adjusted. This condition is satisfied if t(3)>t(1) in Figure 12.

It is possible to compute these delays if the length of the circuit is known and it is known whether or not echo-suppression equipment is in use on a particular call. Experience on calls up to a couple of hundred miles within Michigan indicates that a short-turnaround delay of perhaps a few milliseconds is sufficient to allow the echoes to be quenched and the circuitry to stabilize.



d	Propagation time
T_C	Clamp time
T_S	Squelch time
☒	Data
|||	Echo

Figure 12. Line Turnaround Sequence

It is possible in this manner to determine the state of the half-duplex system with respect to a particular terminal through an inspection of the data set control-lead signals. However, with respect to the transmission network, an ambiguity exists at certain times when one terminal is unsure of the state of the other. The period of this ambiguity extends from the time that one terminal raises its REQ SND lead and enters the transmit state until the other terminal recognizes the rise of the CAR DET lead and enters the receive state. The following rules will obviously minimize the contention incidence:

1) A terminal will not attempt to enter the transmit state if its CAR DET lead is raised;

2) Upon entering the idle state a terminal will not attempt to transmit until a timeout equal to twice the propagation delay has expired.

6.2 Error Recovery in Half-Duplex Systems

Up to this point no mention has been made of the format of the transmitted message or the nature of the replies transmitted by the receiver. In this section, a specific protocol will be presented which has the capability of recovering from lost-record and bad-checksum situations. In this discussion, frequent reference will be made to communications control characters such as ETX and NAK. These should be interpreted as logical functions only for the purposes of the present discussion; the exact codes and sequences chosen to represent these functions will be discussed in a later section.

There are two tools which provide mechanisms for error recovery: the use of a cyclic-redundancy checksum and the use of the ENQ function. The checksum is primarily used to distinguish line errors which occur during transmission of message text, although it is also useful in certain cases involving contention. The ENQ character is used in conjunction with a timeout function primarily in cases involving contention, but is also useful in certain cases involving line errors during the synchronizing phase.

The transmission conventions can be concisely summarized thus: each message including text is terminated with a communications control character (ETX, ETB, etc.) which indicates that a two-character checksum follows. Following transmission of such a message, the transmitting terminal (A) initiates a timeout of perhaps a second or so in duration. Meanwhile, the receiving terminal (B) checks the transmitted checksum against its own calculations and transmits:

1) A positive acknowledgment (ACK0 ACK1) if the checksums agree, or

2) A NAK if they do not agree.

The choice of which of the two positive acknowledgments to transmit is determined from a count of the received records as kept by the B receiver. If this count is odd, then ACK0 is transmitted; if even, then ACK1 is transmitted. Remember that B may fail altogether to receive the message due to failure to synchronize or to line contention. Now if A, which is maintaining a count of its transmitted records in step with B's count, receives the appropriate positive acknowledgment, then it proceeds with the next operation normally. If A receives either the wrong acknowledgment or the NAK character, then it retransmits the message exactly as before. If the timeout started by A expires, then A must assume that either of two cases prevail:

1) The message itself was lost and no acknowledgment was sent by B, or

2) The message was received by B but the acknowledgment itself was lost by A.

To resolve these two cases, A sends the ENQ character to B which, by convention, responds with the last acknowledgment character it transmitted. Terminal A now enters the same state as when it transmitted the original message and checksum; that is, it initiates its timeout and acts upon the received acknowledgment as defined. Refinements to protect against infinite loops under hard-error conditions are simple and obvious.

Several minor points should be made here before proceeding to the formal analysis of the reliability and scope of this procedure. First, this procedure must be used in every case that either terminal is in an idle condition, that is, not expecting an acknowledgment. Therefore, if a delayed go-ahead convention is used to prevent receiver buffer overflow, then the go-ahead message (or character) must be treated as a separately acknowledged message. Second, we assume that neither terminal will attempt to transmit if its CAR DET lead is raised (see Section 7.4). This requirement is not a necessary one, but it reduces the incidence of contention and, of course, the number of time-consuming recovery procedures.

Consider the two finite automata of Figure 13, one designated the transmitter (A) and the other the receiver (B). Assume both automata start in state 0 (a symmetrical argument holds if both start in state 1). We will assume that the two automata operate asynchronously but that the output symbols of A will be the input symbols of B and vise-versa.

A message transmission is completed when the ETX is transmitted by A. As indicated B may interpret this properly, as a bad-checksum record, possibly as another communications-control character, or may fail to synchronize and miss the record entirely. Now, because of the structure of the record and the checksum, it is very unlikely that B will interpret the ETX as another communications-control character, so that the only real choices left to B are:

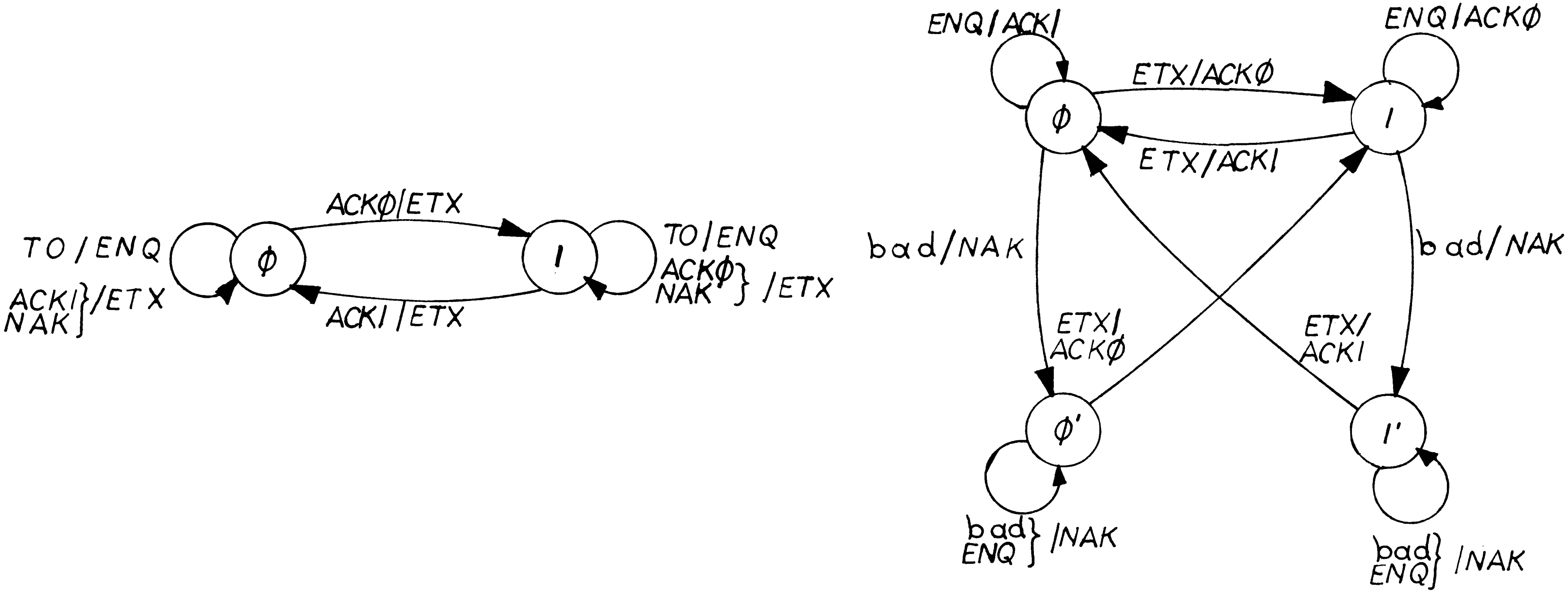
a) properly received,

b) bad checksum,

c) no reception at all.

Note that only in the case of (a) will the receiver transition to state 1 and emit the ACK0 character. Case (b) will cause the receiver to emit a NAK character and return to state 0, while case (c) causes, of course, no action at all.

While all this is going on, the transmitter is sitting in state 0 with its timeout running. If it receives an ACK0 from B as above, then it disconnects the timeout and transitions to its own state 1. During this transition, it may transmit another message to B (which is now in its own state 1) and the transmission-acknowledgment cycle can continue as before with both automata in state 1. In this manner, with no errors, both automata remain in step flip-flopping between states 0 and 1.



Legend: received/transmit

TO	Timeout
ENQ	Enquiry
ACK0 ACK1 NAK	Acknowledgments
ETX	Message good parity
bad	Message bad parity

Figure 13. Model for Error Recovery

Returning, however, to the first case, assume that the transmitter receives a NAK as an acknowledgment from B. Assuming correct reception of this character, A can assume that B is still in state 0 and can retransmit its record, remaining itself in state 0. If the timeout expires, then A does not really know whether B received the record correctly but A did not receive B's acknowledgment, or whether B failed to receive any record at all. Transmitter A, therefore, sends an ENQ character to B, returns to state 0, and restarts its timeout. Again assuming correct reception by A, if A understands B's last acknowledgment was ACK0, then the message was, in fact, properly received by B and A can continue as in the error-free case. If any other acknowledgment is understood by A, then A can assume that B has lost the message entirely and A can retransmit the message. As an exercise, the reader can verify that most second-order errors (errors in the acknowledgment characters themselves) still do not compromise the eventual successful recovery of this procedure. The only requirement is that line garbling does not change one type of acknowledgment into another. An analysis similar to that presented here can be found in [2] and [24]. The use of the ENQ timeout function and the even/odd acknowledgment convention is proposed in [14] and [37].

6.3 Busy Conditions and Wait Interlocks

In typical cases involving remote job entry and display equipment, situations may occur in which a logical sink device attached to a terminal has a basic operating speed much below the operating speed of the transmission link. In such cases the device will appear busy if an attempt is made to transmit a message to it before it has finished processing a previous message. With the foregoing transmission protocol in mind it is obvious that a positive interlock is available if the receiver of a message destined for such a device can withold the acknowledgment until the device is in fact ready to process the next message. However, if the waiting time is longer than the ENQ timeout period of the transmitter, then some reply conventions are needed to prevent spurious recovery attempts possibly leading to abandonment and disconnect by the transmitter.

What is needed here is an acknowledgment convention which allows the verification of a correctly received message and the indication of readiness to receive the next message to be separately indicated in the acknowledgments. We invent two new control functions for this purpose, called GA and WABT, where GA indicates readiness to receive the next following record and WABT denies this condition. Furthermore, we establish that the ACK and NAK functions be restricted to indicate the affirmation or denial of a correct message reception, as indicated by the checksum comparison.

Now the transmitter can establish a strategy to maximize the throughput of the system. We must remain within the constraints of the error recovery environment, however, which evidently requires the following behavior:

1) Once a terminal has transmitted a record containing a message to a device it cannot transmit any traffic whatsoever except an ENQ, and that only after a timeout period equal to at least twice the propagation delay.

2) If the acknowledgment record received by the transmitting terminal contains an ACK for the message in question, then the transmitter can reclaim the buffer and proceed to the next operation. If the acknowledgment record contains a NAK for the message, then the transmitter marks this message for retransmission and does not proceed to the next operation. However, the ACK or NAK in themselves do not imply that the transmitter can send the next message if one becomes pending.

3) If the acknowledgment record received by the transmitting terminal contains a WABT for the message in question, then the transmitter sets a wait latch which prevents further traffic altogether. In particular, no ENQ operations are permitted and the status of the last transmitted message is not indicated. Transmission of the WABT implies that the terminal transmitting the WABT has accepted responsibility for generating and verifying reception of a GA message at some future time. Furthermore, the WABT applies only to the message and its connected device in question, and traffic may proceed normally for other devices which may be attached to either terminal.

Note that the ACK-NAK and WABT-GA functions are logically independent, so that various operations can be indicated by combining these functions in a single acknowledgment.

Now the receiver's strategy may be quite complex, and depend upon a number of delay conditions likely to occur. Consider a state of the system in which B has just indicated that A's last message has been correctly received but has not indicated whether or not it can accept further traffic.

Now, we assume that either terminal can attempt to use the link for transmission of control functions. Let A transmit its next message anyway. However, we assume that B's receive buffer may not be empty at this time and it may happen that B has no place to put A's next following message. In such a case, a buffer overrun condition can exist at B and part or all of the message may be lost. The following strategies are possible:

1) If B just happens to have room in its receive buffer at this time, then the record can be acknowledged appropriately and transmission may continue normally.

2) If a buffer overrun condition exists at B and, furthermore, if B expects that its receive buffer will be empty within a short interval comparable to the message retransmission time, then B can delete the partially received record and acknowledge with a NAK. This will cause A to immediately retransmit its record, which should then creep into B's receive buffer at about the same rate that the previous message is creeping out of it. This creepy behavior is possible in RAMP-like systems [25, 26].

3) If B knows that its receive buffer will be tied up for a period greater than the retransmission time but less than the timeout interval for ENQ operations, then B can wait until the buffer is empty and zap a NAK back to A, which then retransmits the record. (Note that this behavior is distinct from (2), since, if B has traffic to be transmitted to A, B would like to get rid of it as soon as possible. Therefore, if B can assume that its receive operations are proceeding snappily, then it can take a chance and sneak a transmit record including the NAK back to A.)

4) If B knows that its receive operations are slow, then it can sit tight and wait for the ENQ function following A's timeout. At this time B has an interesting set of possibilities. It can continue to sit tight and let A retransmit ENQ's; it can bounce a NAK off A, causing it to retransmit its last record; or it can transmit a WABT to A, which tells it to shut up and await a GA from B. Note that the GA message in this case must be transmitted as a separately acknowledged message, since the WABT gave A explicit permission to send acknowledgments and messages for other devices. This is the only way that B can punch

through traffic for A without acknowledging A's last record.

It is desirable here to allow several acknowledgments to be concatenated in the same record, so that the transmitter can release its previous message buffer concurrently with transmission of the next message. Also, note that at least one acknowledgment character must be received by a terminal in the acknowledgment-expected state before any further transactions can occur. Finally, note that the GA is treated like any outbound message from B and must have the even/odd acknowledgment convention applied.

In the operation of half-duplex systems in highly interactive conversational environments there is a need for a special message which can be used to interrupt the execution of a program or the transmission of data. This message is commonly called an asynchronous interrupt [25, 26] and is designed so that delivery is assured with high probability in the event of outstanding device hangups and transmission errors. It is proposed that a new message called ATTN be invented for this purpose. It will have much the same characteristics as the GA message in that it contains no text but will be separately acknowledged. This message may be transmitted at any time in which the intended receiver is not waiting for a positive or negative acknowledgment.

There is an interesting, not altogether accidental, interpretation of these procedures in connection with channel and control-unit operation internally to a time-shared system such as the System/360, and most particularly to a duplex system involving several control units. In such a network a message is a channel-to-control unit transmission and is terminated by an ending condition either by the channel or by the control unit, following which a status cycle (device end) is initiated by the control unit. If no errors are found then an ACK can be assumed. Otherwise unit-check is included in the status byte and NAK is assumed. Now, if a channel attempts to select a control unit which is attached to another channel by a different path, then a control-unit-busy (WABT) condition is returned to the channel which, by convention, awaits a control-unit-end indication (GA) when the control unit again becomes available.

6.4 Multiplexing Several Parallel Paths

Now, using the machinery developed above to describe the basic point-to-point half-duplex system, we show how a system of queues can be constructed so that several parallel logical message paths can be sustained using several parallel half-duplex links. There are two reasons why this is desirable: first, an obvious speed improvement can be achieved by using a number of parallel transmission paths operating within a fixed maximum rate, and second, the use of parallel logical paths allows the transmitter and receiver to concatenate messages and acknowledgments along each path and to minimize the number of turnaround delays. This procedure implies, of course, a delay while a message waits in a queue for its turn to be transmitted. We assume that this delay can be tolerated in the interest of higher effective transmission rates.

The notion of parallel logical paths here is not necessarily assumed to be restricted to separate real-time operations, but could be applied to serial processes in which the several buffers are used sequentially by a single real-time operation. As before, we assume the method of scattering records among the transmission paths to be a strategic rather than a procedural operation. The system, given an incidence of message requests, attempts to propagate each to the corresponding receiver buffer with equal priority along any path available at the moment.

Now, consider that each terminal in this network maintains a queue of messages to be transmitted together with the logical destinations of these messages. We can assume that the distribution of the messages queued at the transmitter follows a meaningful algorithm and that each terminal maintains a connection matrix showing how any logical sink device can be reached from any terminal. Furthermore, we assume that the cost of transmission along any path can be precomputed by the terminal based on its information as to network loading.

Given all of these constraints, the solution of which represents an interesting problem in itself, a set of protocols can be developed which provide for point-to-point transmission operations within the network and for conventions in error recovery. In the most basic analysis, the behavior of the network is dependent upon that of the individual transmission link, and this has already been described.

At the receiver, incoming messages may be routed to relatively slow devices. For each of these devices an input buffer is assumed to exist at any time in which an input message may be received by that device. If the buffer does not exist then a device-busy condition is assumed. Once the message has been stored in a buffer, however, there may be some delay before the message is propagated from the buffer to the device or is processed by the system.

6. Processor-Processor Communications

The transmission operation in the system can then be visualized with the aid of Figure 14, in which a queue of message buffers awaiting transmission is maintained at each terminal and a corresponding set of buffers is maintained at the receiver. The strategy of message protocol is obviously then to provide the highest message rate consistent with the smallest queueing delay. Each message stored in the transmit buffer at one terminal is to be delivered to the receive buffer of the other. Each transmission is to be acknowledged in some fashion so that the transmitter may either retransmit the message or reclaim the buffer.

The states of the system can be assigned as described previously, but with the understanding that each buffer may be in a particular state independent of all the others. Once a message has been stored in a transmit buffer the buffer will be in the transmit-pending state. When transmission begins, the buffer will be in the transmit state, and, following transmission, will be in the acknowledgment-expected state. When the acknowledgment is received, the buffer will return to the transmit state or be released, depending on whether the message was positively or negatively acknowledged. If, when in the idle state a message is received, a buffer goes into the receive state, and following an end-of-message indication may revert to either the idle state or the acknowledge state, depending on whether an acknowledgment was required or not.

Now consider that each transmitter maintains a queue of all buffers in the transmit-pending state. When a terminal can enter the transmit state it transmits all buffers on this queue and marks each buffer accordingly as in the acknowledgment-expected state. Transmission continues until the transmit queue is empty, after which a timeout is started separately for each buffer. At the receiver a collection of buffers is maintained, each presumably attached to some device. We assume that each transmit buffer is marked as to which receiver buffer should pick up the message. At the receiver, then, the incoming messages are each checked for validity and parked in the appropriate buffer. Following the transmit operation, then, some of the receive buffers are full and in the acknowledgment state. Where known, the acknowledgments are posted with the receiver buffer. All of these acknowledgments are concatenated and transmitted back to the transmitter, which then deletes from its queue all those buffers which have been successfully transmitted, and marks the others to be retransmitted. Note that all of the control functions ACK, NAK and ENQ must be coded in such a way that they can be associated with the logical buffer to which they belong.

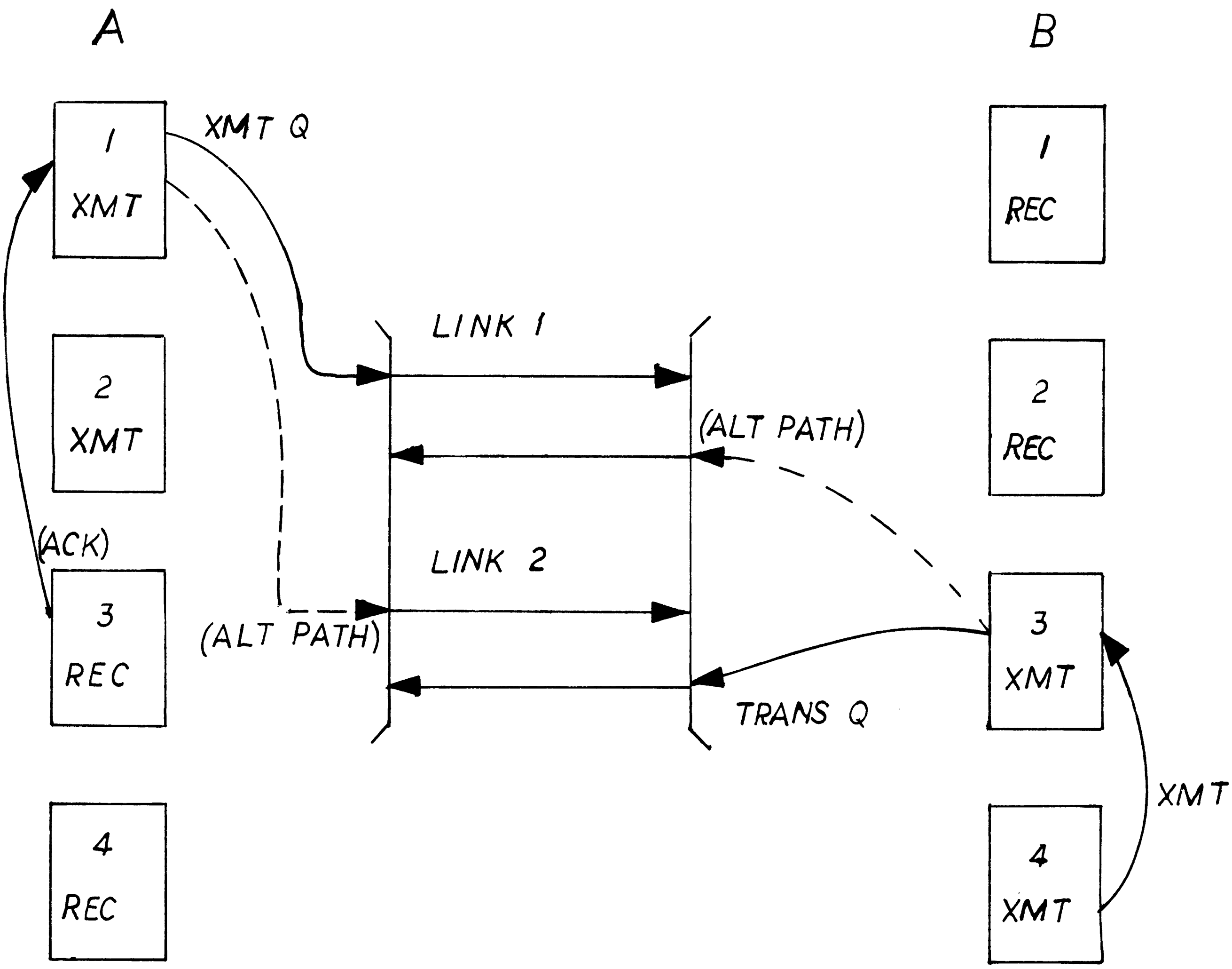


Figure 14. Multiplex Queue Organization

6.5 Coding Messages

So far, the following eight control functions have been identified: ETX, ACK0, ACK1, NAK, ENQ, WABT, GA and ATTN. We have also indicated that, in multiple-path operations, a means is required to code the logical buffer to which these apply. Furthermore, a need exists to encode special sequences so that these control codes can be transmitted as part of text information. In this section a coding procedure will be described which is based on formal analysis. This approach provides not only a concise definition of the coding procedure but also defines in an automaton-theoretic sense the hardware/software processor which performs the encoding/decoding operations.

We begin by defining a number of classes which effectively partition the set of all characters which may be propagated on the system. Some of these classes will be reserved for control functions; but the remainder can be used in the message text itself. This partition does not mean that the full set of characters is not available for use in messages, but only that a control character appearing in a message must be recoded in a special manner to avoid confusion with the control characters used by the system itself. The recoding process is often called a transparent binary procedure [14]. Following is a functional summary of the control-function classes adapted from [14] and [37]. By implication, all of the remaining characters belong in the text class:

STX (Start of Text). A character used to signal the start of a text message. May be preceded by a selection sequence (see below). A message beginning with an STX character always requires an acknowledgment. This character causes the checksum to be reset in certain cases.

SOH (Start of Heading). A character used to signal the start of a control message intended for a particular terminal on a store-and-forward system. May be preceded by a selection sequence and followed by a selection character which identifies the particular terminal to which the message is directed. For the purposes of data transmission, however, no distinction is made between the SOH and STX characters. Note that this is a more restricted interpretation than specified in [37]. A message beginning with an SOH character always requires an acknowledgment. This character causes the checksum to be reset in certain cases.

ETX (End of Text) and ETB (End of Text Block). Characters which end a message begun by either STX or SOH. No distinction is made herein between these two characters (but see [37]). These characters are always followed by a two-character checksum sequence and a line turnaround for acknowledgment purposes.

IUS (Information Unit Separator - also called ITB). A character used in identically the same fashion as the ETX/ETB characters but does not imply line turnaround for acknowledgment purposes. In links using IBM terminals the IUS-checksum sequence must be followed by at least two SYN characters to allow the equipment to resynchronize.

SYN (Synchronous Idle). A character used both to synchronize the receiver deserializer process and to fill otherwise "dead spots" in the transmitted character stream. Must be inserted every second or so in links using IBM terminals for equipment reasons. May be preceded by a DLE character in transparent-binary operations. Also, in links using IBM terminals, at least two SYN characters must be transmitted after certain checksum sequences involving the ITB character and certain acknowledgment characters. The SYN character never participates in the checksum accumulation.

DLE (Data Link Escape). A character used in code-extension sequences for two purposes:

1) To differentiate the use of communications control characters as text information from their use as control information in transparent binary sequences.

2) To provide a mechanism in which extended control functions can be added to the set assigned the USASCII communications control characters (see Section 5.2). Examples are: ACK0 (DLE-0), ACK1 (DLE-1), WABT (DLE-?), GA (DLE-:) and ATTN (DLE-;).

A DLE sequence of either type begins with a DLE character, continues with an optional sequence of intermediate characters of a specified type, and concludes with a final character of a specified type.

ACK0 and ACK1 (Positive Acknowledgments). A pair of two-character sequences coded as DLE-0 and DLE-1 respectively and used to signal acceptance of a message requiring an acknowledgment. May be preceded by a selection sequence. The choice of which of these two sequences to use in any case is determined by a count of the received records as described in Section 6.2. In transmission links involving IBM equipment these sequences may cause the receiver to end the current record and abandon synchronization.

NAK (Negative Acknowledgment). A character used to signal rejection of a message requiring an acknowledgment. May be preceded by a selection sequence. In transmission links involving IBM equipment this character may cause the receiver to end the current record and abandon synchronization.

WABT (Wait Before Transmit). A two-character sequence coded DLE-? And used to explicitly request the transmitter to suspend message traffic until the transmission of a GA sequence as described in Section 6.3. May be preceded by a selection sequence. Upon transmission of this sequence the transmitter agrees to take responsibility for the delivery and verification of the GA sequence. In transmission links involving IBM equipment this sequence may cause the receiver to end the current record and abandon synchronization.

GA (Go Ahead). A two-character sequence coded DLE-! And used to explicitly request that transmission be resumed following a WABT sequence. May be preceded by a selection sequence. This character is only generated by a receiving terminal if it has previously generated a WABT message for the purpose of interlocking the transmitter against further transmission while the receiver completes some time-dependent operation. This sequence is considered a message and must be acknowledged in the usual fashion. Note that this sequence does not appear in either [14] or [37].

ENQ (Enquiry). A character used in error recovery operations as indicated in Section 6.2. May be preceded by a selection sequence. In transmission links involving IBM equipment this character may cause the receiver to end the current record and abandon synchronization.

ATTN (Attention). A two-character sequence coded DLE-; and used to explicitly request an asynchronous interrupt to the operating system of the connected terminal or computing system. May be preceded by a selection sequence. This sequence is considered a message and must be acknowledged in the usual fashion. Note that this sequence does not appear in either [14] or [37].

EOT (End of Transmission). A character used as a general reset function to clear the circuit of outstanding error hangup conditions. In transmission links involving IBM equipment this character may cause the receiver to end the current record and abandon synchronization. The following uses are explicitly recognized:

If the EOT character is the first (and only) non-SYN character in the record then a general reset function is performed along all logical paths, so that outstanding acknowledgment and busy conditions can be cleared. This operation is performed prior to a bidding operation in the contention-with-bidding mode and in certain error recovery situations.

2) If the EOT character is preceded by a selsection sequence, then the reset function applies only to the logical path selected and the record is not ended.

3) If the EOT character is not the first non-SYN character in the record and is not preceded by a selection sequence, then the only operation is to end the current record. The EOT character is used in this manner to end a record consisting entirely of acknowledgments or of messages ended with an IUS-checksum sequence.

The enumeration above defines a partition of all those characters which are available in a transmission code. For the purposes here all of these characters are given the interpretation and character-code assignments according to the USASCII conventions established in Section 5.2.

The syntax of a record structure which meets these requirements can be effectively described as a language-generating device which generates (or recognizes) a set of sequences of symbols in a deterministic manner. This device may be thought of as a model for the logic design of a hard-logic terminal or as a model for a program construction in a programmable terminal. A language-generating device of the type useful here can be described completely either by a state-transition diagram of an automaton or by a system of rules of a formal grammar. Either description defines a device which can generate (or equivalently can recognize) a set of sequences of symbols. Each of these sequences is called a sentence and the set of all sentences generated (or recognized) by the device is called its language.

Figure 19 shows the state-transition diagram of an automaton which describes such a language-generating device. The language of this automaton is in fact a subset of that used in IBM Binary Synchronous Communications [14] and is also a subset of that proposed as a USA standard [37]. This particular syntax does not, however, provide for either transparent binary or for multiplexing operations such as proposed in Section 6.4.

Under the intended interpretation here the language generated (or recognized) by the automaton is the set of all valid records, consisting of messages and control functions, which can be transmitted (or received) by a terminal. The generation (or recognition) process begins when the automaton is started in a particular state ("PHA" in Figure 15), continues as the input characters are presented in turn, and concludes in a particular final state ("EOT" in Figure 15) if the record is accepted. If the automaton ever receives an input which is not assigned in a particular state then the automaton is said to be blocked and the transmission is in error.

In the alternate description as a formal grammar the language-generating device can be described by a set of rules of the form A -> aB and A -> a, where a is a terminal symbol, that is a symbol which might occur in a record, and A and B are nonterminal symbols, that is symbols introduced for the purpose of assigning a structural description to the record. In fact the nonterminal symbols in these rules will correspond exactly to the labeled states in the state-transition diagram and, in this simple case, the rules will bear a simple relationship to the transition functions of the familiar state-transition table.

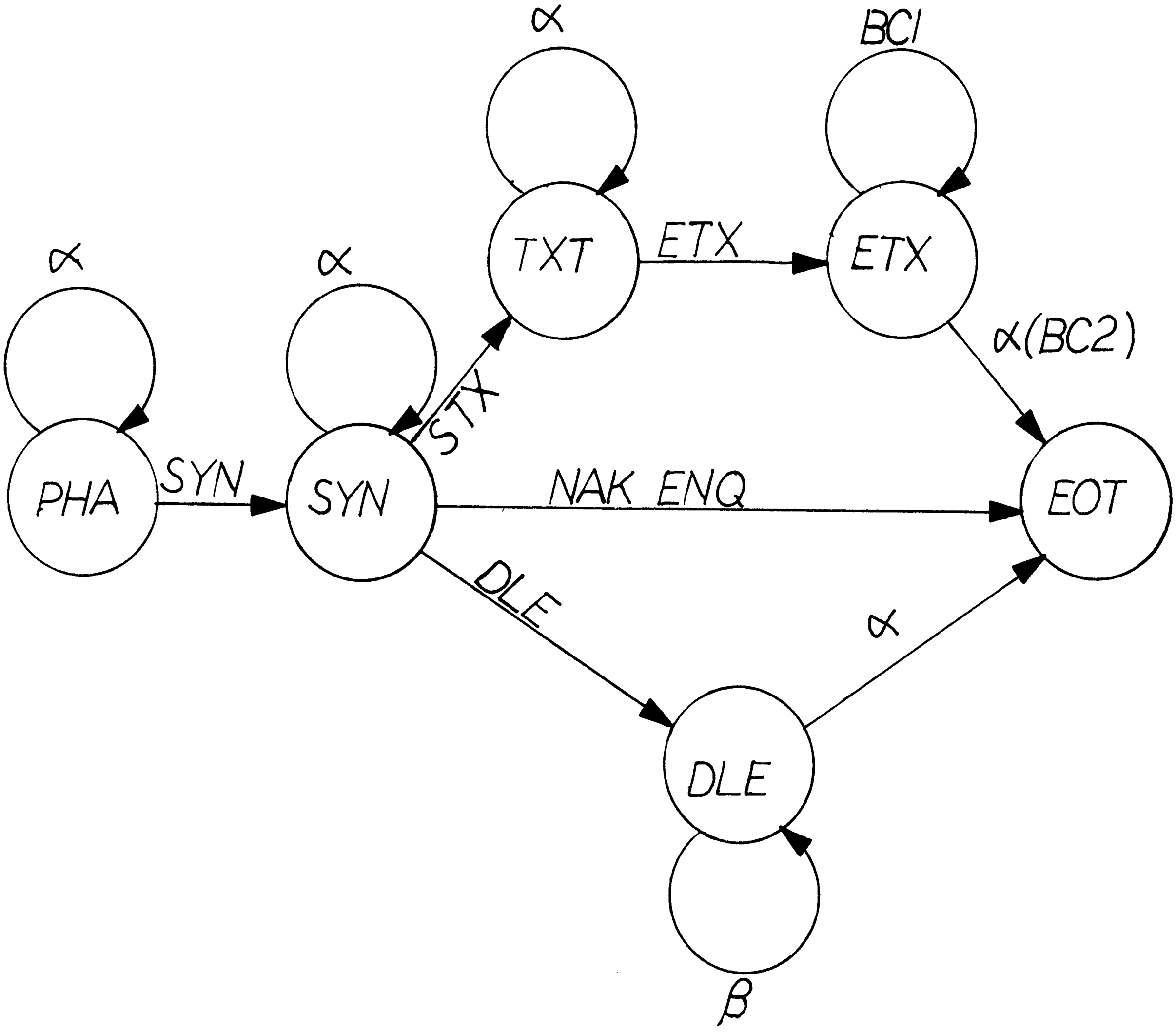
Now the rules of the formal grammar for the language described by the automaton of Figure 15 can be written out in the conventional Backus-Nauer form (the justification for this procedure is beyond the scope of this report, but an inspection will reveal the general technique). In the following V represents the set of all terminal symbols (the vocabulary). If a set notation is used to represent a class of terminal symbols, then the rule is assumed to be replicated once for each element in the class.

```
<PHA> =:: {V-SYN} | <PHA> {V-SYN}
<SYN> =:: <PHA> SYN | <SYN> {V-{STX+NAK+ENQ}}
<TXT> =:: <SYN> STX | <TXT> {V-ETX}
<BCC> =:: <TXT> ETX | <BCC> BC1
<DLE> =:: <SYN> DLE | <DLE> BETA
<EOT> =:: <SYN> {NAK+ENQ} | <BCC> {V-BC1} | <DLE> {V-
     BETA}
```

The operation of a formal grammar language-recognition device can be described in the following crude fashion: starting with a string (the record), an attempt is made to match the right-hand side of a rule to a substring of the string (starting from the beginning of the string). When a match is found the substring is replaced by the nonterminal symbol on the left of the matching rule. This process is continued until either no rule applies, in which case the wrong substitution might have been made and the process must be "backed up" to try an alternate matching rule, or the string is reduced to the single starting symbol (<EOT> in the rules here). If it is possible to perform the reduction operations as described, then the string is in fact a valid sentence (or record), otherwise it is not.

It should be clear from the above descriptions that the two alternate descriptions of language-generating devices are in fact equivalent and that both describe in a deterministic manner the syntax of a record structure as required in the previous sections. Note in passing that a much more complex syntax can be described using these same approaches, and such devices may have practical application in more complex store-and-forward systems than are imagined here. For the present discussion only the theories of finite-state automata and regular grammars are pertinent.

6. Processor-Processor Communications



Legend:
β A set of intermediate chars [37]
α Any char not mentioned

Figure 15. Simple Transmission Syntax

6.6 A Proposal for a Transmission Protocol

Now, with the foregoing tutorial in mind, let us develop an automaton-theoretic model sufficiently complex to implement the functions called for in Sections 6.1 through 6.4. The same lexical symbols will be used, as described in the beginning cf Section 6.5. Where possible the conventions will coincide with those established in [14] and [37]. The following guidelines will be pertinent:

1) All messages involving text will be separately checksummed using the 16-bit two-character checksum computed as shown in Figure 3 of Section 3.3. All character codes may be included in the text, so that transparent binary conventions will be necessary. All messages involving text will be acknowledged. If such a message is received erroneously it must be retransmitted.

2) Control-function sequences need not be separately checksummed since each character is assigned an odd-parity check bit. If a control-function sequence is received in error then it is disregarded. Only the GA and ATTN control-function sequences are acknowledged, since they are in reality messages. All other control-function sequences are never acknowledged.

3) Each message involving text and each control-function sequence may be preceded by a selection sequence which identifies the logical source and sink path as required in Section 6.4. The structure of the selection sequence is not important except that is should be short, of fixed length, and designed so that transmission errors will not likely change a valid selection sequence into another. If a transmission error is detected in a selection sequence then the following message or control-function sequence will be disregarded.

4) Messages involving text can be included in the same record as control-function sequences, but in such cases the control-function sequences must occur between the messages (i.e., not embedded within the message text itself).

5) It must be assumed that transmission line errors and equipment faults will be a routine, although infrequent, occasion in the system. Therefore the syntactic specification should be exhaustive and the automaton should not block at other than designated error-recovery states.

6) Compatibility with [14] and [37] shall be maintained in the following manner: each terminal implementing this protocol shall operate in either of two modes: a) an unrestricted mode in which the full set of procedures outlined here are applicable, and b) a restricted mode in which:

1) All acknowledgments, including all of the DLE sequences, end the record of which they are a part. Therefore multiplexing in the sense of Section 6.4 is not possible and selection sequences are not meaningful.

2) Following an IUS-checksum sequence the checksum will not be reset at the next STX or SOH character and will be allowed to accumulate at the next character following the checksum.

3) Operation is confined to the contention mode, in which a terminal must bid for the transmission circuit before message transmission can begin (see [14] and [37]).

4) The WABT, GA and ATTN sequences are never acknowledged. The GA and ATTN sequences are not defined and will not be transmitted.

5) In response to a record containing an IUS-checksum sequence the acknowledgment applies to all messages in the record; that is the positive acknowledgment will be transmitted if all checksums compare properly and the negative acknowledgment will be transmitted if any do not.

Figure 16 shows a syntax which satisfies these guidelines. The figure shows a ten-state automaton which generates (or recognizes) a record which might be transmitted (or received) by a terminal. The operation of this automaton can be explicated as a function of each of its ten states as follows:

PHA (Phase Synchronization). The receiver always starts in this state when searching for the correct deserializer synchronization. In this state all character codes are ignored except SYN, which causes a transition to the SYN (Synchronous Idle) state.

SYN (Synchronous Idle). This state is entered following deserializer synchronization, following each message terminated by an IUS-checksum sequence and following each control-function sequence. This state is also entered if a cancel sequence (ENQ or DLE-ENQ) is detected in a text state or if a selection sequence is detected in error. An EOT character detected in this state causes a transition to the EOT (End of Transmission) state, which causes in turn a line turnaround (see Figure 10). In the restricted mode of operation the control-function characters ACK, NAK and ENQ also cause this action.

SEL (Select). This state is entered only from the SYN (Synchronous Idle) state when the first character of a selection sequence is detected (most likely a letter). It is presumed in Figure 16 that all of the operations necessary to identify a selection sequence are implied in the single state shown. A selection sequence can terminate only in a STX, SOH or DLE character, and in these cases the automaton takes the paths shown. A selection sequence terminated by any other character is assumed ill-formed and operation continues in the SYN state.

TXT (Text). This state is entered either from the SYN (Synchronous Idle) or the SEL (Select) states when an STX or SOH character is detected. Upon entry to this state the checksum is reset. In the restricted mode of operation the checksum is reset only if the STX or SOH character begins the first message in the record. Operation continues in this state as text characters are accumulated until either an ETB, ETX or IUS character is detected, which cause the text message to be ended. A SYN character is not considered part of the text information and does not participate in the checksum accumulation. An ENQ character has the effect of canceling the message and in this case operation continues in the SYN state.

DLE (Data Link Escape). This state is entered when a DLE character is detected in either the SYN (Synchronous Idle) or SEL (Select) states. According to the conventions established above, the DLE initiates a code-extension sequence consisting of a single DLE character followed by a number of intermediate characters and terminated by a final character. Only a few of these DLE sequences are defined for the purposes here and none involve an intermediate character. As required by [37] however, intermediate characters are properly processed as shown, although the sequence is marked invalid. An STX or SOH final character following a DLE character causes a transition to the TTXT (Transparent Text) state as shown. All other final characters following a DLE character cause a transition to the SYN state. In the restricted mode of operation the arrow leading from the DLE state to the

SYN state is diverted instead to the EOT state.

TTXT (Transparent Text). This state is entered from the DLE (Data Link Escape) state when an STX or SOH character is detected and in other cases as shown. Upon entry to this state the checksum is reset. In the restricted mode of operation the checksum is reset only if the STX or SOH character begins the first message in the record. Operation continues in this state as text characters are accumulated until a DLE character is detected. The DLE character itself does not participate in the checksum accumulation.

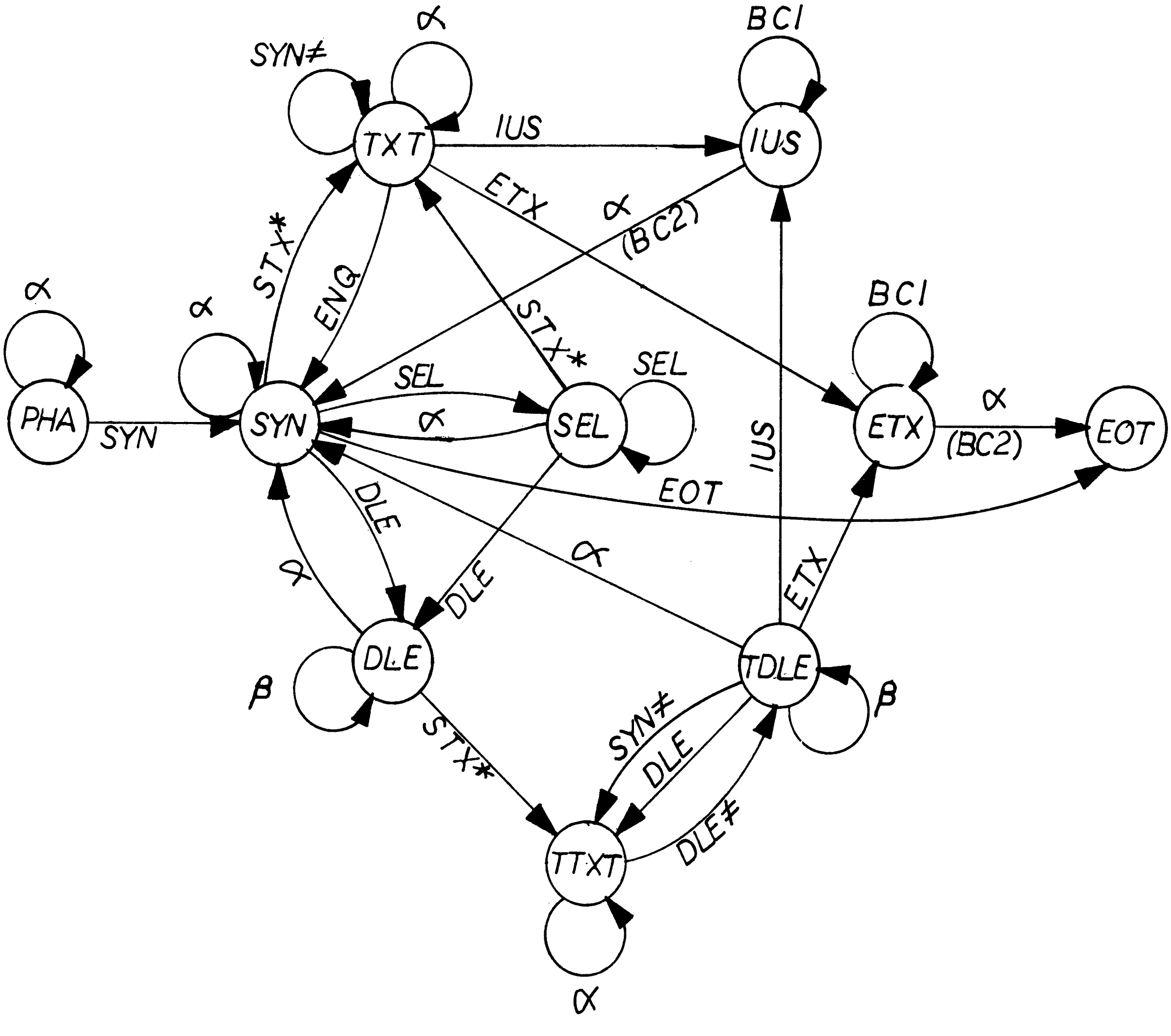
TDLE (Transparent DLE). This state is entered only from the TTXT (Transparent Text) state when a DLE character is detected. The DLE character initiates a code-extension sequence as described under the DLE (Data Link Escape) state. An ETB, ETX or IUS character following the DLE character causes a termination of the text message as shown. All other characters cause a transition to the SYN (Synchronous Idle) state, although only some of these are valid. In the restricted mode of operation the arrow leading from the TTXT state to the SYN state is diverted instead to the EOT state.

IUS (Information Unit Separator). This state is entered from a text state when an IUS character (also called ITB) is detected. It is presumed in Figure 15 that all of the operations necessary to generate the two-character checksum are implied in the single state shown. As shown, a two-character checksum sequence is generated and exit is made to the SYN (Synchronous Idle) state.

ETX (End of Text). This state is entered from a text state when either an ETB or ETX character is detected. It is presumed in Figure 15 that all of the operations necessary to generate the two-character checksum are embedded in the single state shown. As shown, the checksum sequence is generated and exit is made to the EOT (End of Transmission) state.

EOT (End of Transmission). This state is entered either from the SYN (Synchronous Idle) state when an EOT character is detected or from the ETX (End of Text) state when the two-character checksum sequence has been generated. This is the only valid final state the automaton can enter following record transmission (or reception). Entry into this state implies a line turnaround (if necessary) in the fashion shown in Figure 10.

6. Processor-Processor Communications



Legend:
- * Reset checksum
- ≠ Do not include in checksum
- β A set of intermediate chars [37]
- α Any char not mentioned

Figure 16. Practical Transmission Syntax

The operation of transmission links utilizing this protocol is illustrated in Figure 17. The transactions shown at the top of the figure correspond to traffic flow from terminal A to terminal B. Note the even/odd acknowledgments and the use of the EOT character to terminate a record consisting only of acknowledgments. The transactions shown at the middle of the figure correspond to bidirectional traffic flow between both terminals. Note the inclusion of acknowledgments in the same record as text. The transactions shown at the bottom of the figure correspond to traffic flow from terminal A to terminal B but with terminal B generating a WABT-GA sequence as described in Section 6.3. Note the acknowledgment for the GA record in A's last transmission. Also note that traffic from B to A could be included in the same record as the WABT and GA control-function sequences. In such a case A is free to transmit acknowledgments to B.

Figure 18 shows the operation of this protocol in a multiplexing environment. In the top line (1) terminal A sends two transparent binary text messages to B using selection sequences B0 and B1. In the next line (2) B replies to A's messages and transmits a nontransparent binary text message to A using selection sequence A0. This message happens to be terminated in an IUS-checksum sequence followed by an EOT character. In the next line (3) A has detected a checksum error on B's message and replies with a NAK character. Terminal A has understood B's reply to the message identified by selection sequence B1 and transmits the next message along this path. However, a transmission error has occurred so that A has missed the acknowledgment for the message identified by B0. In the next line (4) B replies to A's last transmission and retransmits the message in error. In the next line (5) A receives B's message correctly, but by this time the ENQ timeout on path B0 has expired, so it generates an ENQ along this path. In the next line (6) B responds with its last acknowledgment for path B0 and, in the last line (7), A continues with traffic along the B0 path. Continuing in this fashion, it is clear that bidirectional traffic flow for several logical message paths can proceed so that error recovery procedures for one logical path do not materially affect traffic flow along other logical paths.

```
        SS      EBB    SS      EBB
TERM A  YT text TCC    YT text TCC
        NX      X12    NX      X12

                   SD E           SD E
TERM B             YL00           YL10
                   NE T           NE T

            A has traffic, B does not.


        SS      EBB             SD S     EBB
TERM A  YT text TCC             YL0T text TCC
        NX      X12             NE X      X12

                   SD S      EBB
TERM B             YL0T text TCC
                   NE X      X12

            Both A and B have traffic.


        SS      EBB            SD S     EBB
TERM A  YT text TCC            YL0T text TCC
        NX      X12            NE X      X12

                   SD D E  SD E
TERM B             YL0L?0  YL:C
                   NE E T  NE T

      A has traffic.  B transmits ACK0-WABT
         record followed by a GA record.
```

Figure 17. Transmission Operations

```
                S  DS         DIBB  DS         DEBB
(1)  TERM A     YB0LT  text  LUCCB1LT  text  LTCC
                N  EX         ES12  EX         EX12

                S  D    D   DS        DIBBE
(2)  TERM B     YB0L0B1L0A0LT  text  IUCCO
                N  E    E   EX        ES12T

                S  N  S         EBB
(3)  TERM A     YA0AB1T  text  TCC
                N  K  X         X12

                S  D   DS        DEBB
(4)  TERM B     YB1L1A0LT  text  LTCC
                N  E   EX        EX12

                S  D   D E
(5)  TERM A     YA0L0B0L?O
                N  E   E T

                S  D E
(6)  TERM B     YB0L0O
                N  E T

                S  DS        DEBB
(7)  TERM A     YB0LT  text  LTCC
                N  EX        EX12
```

Figure 18. Multiplex Transmission Operations

7. TERMINAL DEVICES

It is highly likely in a vigorous timesharing system that some quantity of several different kinds of terminals will claim service at any particular time, including those manufactured by IBM, AT&T and several others. Notwithstanding policy decisions as to the economy or usefulness of such heterogenous connections, it seems likely that the use of miscellaneous and "odd" terminals will flourish, at least in timesharing utilities which provide connection to the switched network and do not provide terminal equipment as part of the service.

By far the most common terminals, however, are particular models manufactured by IBM and AT&T. Three of the most common of these will be described in the following sections. Most of the presently available "odd" typewriter-like equipment is designed to appear to the processor as one or the other of these devices. In Section 8.3 a terminal control device for use in the central equipment complex is described which can service these devices as well as any other likely to claim service on the switched network using conventional data set equipment.

<u>7.1 AT&T Models 33/35 Teletypewriter</u>

Teletype Corporation, a wholly owned subsidiary of AT&T, has for many years manufactured a rugged although unbeautiful line of electric typewriters designed for use in the TWX, Telex and press services. The most ubiquitous of this line, the Models 12 and 15, still see heavy usage in commercial wire services. The later models of this line, the Models 33 and 35, have been extensively used in recent years on the TWX and Telex networks and may be reasonably expected to supplant the older models eventually. The recently announced Models 37 and 38 are the latest in this line and offer higher operating speeds and additional features. Apparently AT&T is having trouble deploying these latter machines in the field however; at least their availability in Michigan is in doubt. Inevitably, of course, these machines have found extensive application as inexpensive yet rugged terminal devices and have enjoyed considerable popularity in the OEM market as console devices for inexpensive machines.

The Models 33 and 35 Teletypewriters and their optional paper tape attachments operate at a speed of 100 words-per-minute (10 characters-per-second) with the USASCII eight-level eleven-unit transmission code. This code is interpreted as shown in Section 5.2 for printing graphics and control characters. The basic Model 33 machine interprets the control functions CR (carriage return), LF (line feed), SP (space), BEL (bell) and ENQ (enquiry). An answerback drum is provided to generate a unique character sequence identifying the terminal, and is triggered by a received ENQ character. The Model 33 may be equipped with a sprocked-feed paper drive system, in which case the control function FF (form feed) is interpreted. This machine can be equipped with a paper tape reader and punch, in which case it is designated 33ASR.

The Model 35 responds identically to the Model 33, but in addition interprets the control functions HT (horizontal tab) and VT (vertical tab). The Model 35 when equipped with a paper tape reader/punch is designated the Model 35ASR. When so equipped the DC1 (Device Control 1) character starts the reader and the DC3 (Device Control 3) character stops it. Many variations in the control functions are possible in this machine and can be controlled using a "stunt box," which provides a series of cams which are actuated by the various control codes and can be connected to almost any circuit.

In all Teletype equipment the operation of the keyboard is independent of that of the printer; and completely separate message transactions can occur. Thus some external mechanism is necessary to return echo or print each character as typed on the keyboard. Machines wired for "half-duplex" operation perform the return echo function locally within the machine itself. Machines wired for "full-duplex" operation expect this function to be performed by the terminal control unit. In either case, nevertheless, the data set operation itself is on a full-duplex basis.

7.2 IBM 2741 Communications Terminal

The IBM 2740 and 2741 Communications Terminals [I8, I9] are combination typewriter-control-unit devices designed for connection to the AT&T 103-series data sets used on either dial-up or leased facilities. The typewriter itself is a ruggedized version of the Selectric office typewriter and uses a "golf-ball" interchangeable printing element. The control unit differs between the two models. The 2740 control unit contains features to detect bad parity records and to indicate to the processor and the terminal operator when invalid data have been transmitted. The simpler 2741 control unit does not contain these features, although each character transmitted to the processor does include an odd-parity check bit.

Both of these models operate using the PTTC seven-level nine-unit start/stop transmission code. Operational speed is nominally 14.8 characters-per-second at 134.5 baud transmission rate. The structure of this code and the interpretation of the code elements are discussed in Section 5.3.

One particularly useful arrangement of the special-feature group for the 2741 is used with TSS and MTS operation. This arrangement includes the basic 2741, together with the 2741 break feature. This feature allows the processor to lock the keyboard and transmit a message at any time, even if the keyboard had been unlocked prior to the operation. This prevents the possibility of a hangup when the system has a message pending for the terminal while the terminal operator is entering but has not ended an input line, and can occur in sophisticated operational protocol such as that programmed in the Data Concentrator (see Section 8.3).

The operation of this terminal can be illustrated with the aid of Figure 19, which shows a contrived operating session. The session begins when the terminal becomes connected to the central processor and sends the STX character, following which the 2741 keyboard is unlocked. When the terminal operator presses RETURN on the keyboard, the two-character sequence NL followed by EOT is transmitted to the central processor and the keyboard is locked. The central processor now sends a message preceded by the STX character, following which the 2741 keyboard is again unlocked. Continuing in this manner, first the processor then the terminal transmit messages to each other framed by the STX and EOT characters.

```
               S      E        S      E
PROC           T text O        T text O
               X      T        X      T

        S      E        S      E
2741    T text O        T text O
        X      T        X      T
```

Figure 19. IBM 2741 Communications Terminal Operations

The implied flip-flop-like behavior can be broken at any time by either the processor or the terminal by transmitting the BREAK signal while the other is transmitting. This signal transmitted to the 2741 causes the keyboard to be locked preparatory to receiving a message from the processor. This signal transmitted to the processor causes a hardware indication to the operating system which can then terminate its outbound message with an EOT and prepare to receive a message from the terminal.

7.3 IBM 1050 Data Communications System

The IBM 1050 Data Communications System, together with auxiliary paper tape and card equipment [I6, I7], provides a complete remote communications facility for data entry via keyboard, paper tape and punched cards, and data output via one or more line printers, paper tape and punched cards. The system can operate on either dial-up or leased facilities, and in either a multipoint or point-to-point mode. The basic system consists of a typewriter-like keyboard and line-printer together with a control unit. Additional paper tape and punched card equipment can be attached to the control unit and the system can be operated as a stand-alone system or in connection with a remote processor.

All system components operate using the same PTTC seven-level nine-unit start/stop transmission code as do the 2740 and 2741 Communications Terminals (see above). The system operating protocol is considerably more complex, however, due to the necessity of selecting the particular system component for transmission during any one operation. This is done using a combination of procedures involving polling, to determine if a particular component is ready to transmit, and addressing, to determine if a particular component is ready to receive. Additional procedures are necessary to generate and process parity information and to recover from error conditions.

The operation of this system can be illustrated with the aid of Figure 20, which shows the control characters and state transitions involved under typical point-to-point operating conditions. Initially, the system is assumed in a control-receive state in which the system is sensitive to polling and addressing operations. Each terminal on a circuit (in this case only one) recognizes a unique two-character selection sequence which identifies the system and a component attached to the system. If the system receives this sequence in the control-receive state it generates one reply if the component is ready to transmit (STX), another if the component is ready to receive (ACK), and yet another if it is neither (NAK). The processor recognizes either of those replies or no reply at all, in which case the entire system is assumed not-operational. In Figure 20 the sequence A1 identifies the station printer and A5 identifies the station keyboard.

Following an STX character, the system either transmits its record or waits for a record from the processor, depending on the transmission direction. In either case the record is ended with an ETB character followed by a checksum character (LRC). Upon reception of the ETB-LRC sequence by either the processor or the system, the receiver generates one reply if the checksum agrees (ACK) or another if it does not (NAK), giving the transmitter an opportunity to retransmit in case of error. In certain cases, the receiver can transmit a message in lieu of the ACK character, which is then processed in the normal way. Operation in this manner continues until the transmitter signals EOT, which again returns the system to the control-receive state.

```
      E           A        A        A        A
PROC  OA5         C        C        C        C
      T           K        K        K        K

         S      EL       EL       EL       EL
1050     T text TR  text TR  text TR  text TR
         X      BC       BC       BC       BC
```

1050-Processor Operations

```
      E   S      EL       EL       EL       EL
PROC  OA1 T text TR  text TR  text TR  text TR
      T   X      BC       BC       BC       BC

         A         A        A        A        A
1050     C         C        C        C        C
         K         K        K        K        K
```

Processor-1050 Operations

```
      E   S      EL E           E   S      EL
PROC  OA1 T text TR OA5         OA1 T text TR
      T   X      BC T           T   X      BC

         A         A   S      EL   A         A
1050     C         C   T text TR   C         C
         K         K   X      BC   K         K
```

Conversational Operations (the hard way)

Figure 20. IBM 1050 Data Communications System Operations

This system can be equipped with the auto-EOB feature, which provides an automatic ETB-LRC sequence following the new-line character when the RETURN key is struck, and break features similar to those described for the 2741 Communications Terminal. These features make it practical to support both the 2741 and the 1050 with only minimal changes in the processor control program.

7.4 Data Set Control

Almost all data set equipment adapted by AT&T for use on the switched telephone network is available with a common control interface to the terminal control unit or terminal device. This interface provides control over unattended answering and disconnection of calls and monitoring of data set conditions. The electrical specifications of this common interface have been standardized [13], but are certainly beyond the scope of this report. On the other hand, the logical description of certain control functions will effectively illustrate the procedures necessary for timesharing system operation on the switched network.

Following is a summary of these control leads and their functions (from [13]):

Circuit CA-Request to Send (REQ SND). Signals on this circuit are generated by the data processing terminal equipment to condition the local data set to transmit. For example, if the data set contains a modulator, the carrier signal shall be transmitted during the ON condition of Circuit CA.

Circuit CB-Clear to Send (CLR SND). Signals on this circuit are generated by the transmitting data set to indicate that it is prepared to transmit data. The ON condition is a response to the ON condition on Circuit CA (Request to Send), delayed as may be appropriate to the data communication equipment for establishing a communication channel to a remote data processing terminal. When Circuit CA is turned OFF, Circuit CB shall also be turned OFF.

Circuit CC-Data Set Ready (SET RDY). Signals on this circuit are generated by the local data set to indicate that it is ready to operate. The OFF condition shall be used to indicate either:

1) Any abnormal condition which disables or impairs any normal function associated with the class of service being furnished.

2) That the communication channel is switched to an alternate means of communication (e.g., alternate voice telephone).

3) That the local data set is not connected to the communication channel (i.e., the data set is "on hook").

The ON condition shall appear at all other times.

Circuit CD-Data Terminal Ready (TRM RDY). Signals on this circuit are used to control switching of the signal converter to the communication channel. The ON condition causes the signal converter to be connected to the communication channel. However, if the station is equipped only for call origination by means external to the interface (e.g., manually or an automatic call origination unit), then the ON condition serves only to maintain the connection established by these external means. When the station is equipped for automatic answering of received calls, connection to the line may be arranged to occur only in response to a ringing signal. The OFF condition removes the signal converter from the communication channel.

Circuit CE-Ring Indicator (RING). Signals on this circuit indicate that a ringing signal is being received from a remote station. This circuit may be required for automatic answering of received calls. The ON condition indicates that a ringing signal is being received. The OFF condition shall be maintained at all other times.

Circuit CF-Data Carrier Detector (CAR DET). Signals on this circuit are used to provide an indication that the data carrier is being received. When the data carrier is lost because the transmitting signal converter is turned OFF or because of a fault condition, the OFF condition follows after an appropriate guard time delay. In half-duplex service where the signal converter is arranged for local copy, Circuit CF may respond to carrier signals from either the local or remote transmitting signal converter. The ON condition indicates reception of the data carrier. The OFF condition provides an indication of the end of present transmission activity or a fault condition.

Figure 21 shows an interpretation of these signals. In brief, if a ringing signal is detected at any time by the data set then the RING lead is raised. If the TRM RDY lead is raised by the terminal equipment then the data set will cause the call to be answered. Following answer, and depending upon the particular data set type, a particular handshaking procedure will be executed, following which the SET RDY (also called Interlock) lead will be raised by the data set. During transmitting or receving operations the

CAR DET lead is raised. During receiving operations this signal is an indication that the remote data set is in fact transmitting; while, during transmitting operations, this signal is an indication that the local transmitter carrier is in fact being generated. In full-duplex systems this signal is always raised.

	RING	TRM RDY	SET RDY	CAR DET	REQ SND	CLR SND
on-hook	0	0	0	x	x	x
ring	1	0	0	x	x	x
connect	x	1	0	x	x	x
req xmt	x	1	1	x	1	0
xmt	x	1	1	1	1	1
req rec	x	1	1	x	0	1
rec	x	1	1	1	0	0
idle	x	1	1	0	0	0
discon	x	0	1	x	x	x

Figure 21. Data Set Interface States

Control over whether the data set is transmitting or not is exercised via the REQ SND lead. When the terminal equipment raises this lead the data set begins transmitting carrier and certain echo-suppression equipment within the telephone network is activated. After a delay in the 200 millisecond range, the CLR SND lead is raised, indicating that data transmission can ensue. When the REQ SND lead is dropped, the data set immediately stops sending carrier and returns to the receive mode. Following such a transition echoes of the immediately preceding message bits may appear at the receiver for a period up to a significant fraction of a second.

If the TRM RDY lead is dropped by the computing equipment the data set goes on-hook, terminating the call.

7. Terminal Devices

Shortly after this lead is dropped the data set will drop the SET RDY lead, indicating that calls will no longer be automatically answered. Some full-duplex data set equipment will automatically terminate a call if no carrier is received from the remote data set for a period exceeding ten seconds or so. Also, some start/stop data sets can be equipped with a feature which will automatically drop the call if a continuous space signal is transmitted for a period exceeding three seconds or so. Start/stop data sets can be equipped with a feature which causes an automatic disconnect in the event of receiving a continuous space signal for a period exceeding a second or two. The common carrier companies stress that transmission protocol, especially in regard to half-duplex facilities, be arranged so that either terminal will disconnect following an appropriate message on the part of the other terminal. If not, however, the telephone central office equipment will drop the call after a few seconds if the originating party goes on-hook.

7. Terminal Devices

8. TERMINAL CONTROL EQUIPMENT

Terminal control equipment for use at the central processor must usually service a moderate to large number of terminal devices of very heterogenous characteristics. At the University of Michigan, equipment with "standard" operating characteristics is connected to the IBM terminal control units, which have been configured at installation to match the expected mix of terminals of each type. Equipment with unusual operating characteristics is connected to the Data Concentrator, a special-purpose terminal control unit which was constructed by the University staff. This device has been programmed to operate a number of special-purpose terminal devices as well as general-purpose IBM and AT&T devices. In the next section several points will be raised concerning the general characteristics desirable in any terminal control unit and supporting supervisory system. Following this is a brief description of the conventional IBM terminal control equipment. Last is an overview description of the Data Concentrator. A more complete technical description of this device is presented in Appendix A.

8.1 Desirable Characteristics

A terminal control unit must provide character assembly/disassembly operations, of course, together with mechanisms for data set control and data transfer to and from processor storage. In order to minimize the overhead of message processing in the supervisory system, the terminal control unit must contain enough hardware to at least recognize a message-ending condition and to perform checksum accumulations and case-shift conversions. Beyond these fundamental requirements, however, a number of interesting tradeoffs exist between complexity in the processor supervisory system and complexity in the terminal control architecture. This point will be illustrated in following sections.

Using any terminal control unit and its supporting software system, it should be possible to configure the message path interconnections between devices so that separate physical devices can be used for input to and output from the timesharing system. In applications using hand-operated keyboard-type input devices in conjunction with display-type output devices, this need is obvious. In addition, the individual device characteristics as seen by the terminal user should be changeable dynamically to fit the particular environment. Thus, a teletypewriter should be operable in full-duplex, half-duplex, or any of several paper-tape transmission modes, parity generation and recognition should be suppressible, and special character code-translation behavior should be realizable. Furthermore, character-buffering and line-editing procedures should be alterable during operation in unusual situations, and, in particular, so that arbitrarily coded data messages can be transmitted to and from the system and remotely located computing and batch-entry equipment. Finally, the system should be readily modifiable, both in hardware and in software, for the support of new terminal equipment which becomes available from time to time. On all of these points, and in particular on the last one, a university computing center is particularly vulnerable.

Of principal importance in these times of rapidly evolving timesharing systems is a uniformity in the interface between the parent timesharing system and the dependent device service routines. Such a uniformity diminishes the system investment in reprogramming for new terminal types, since it is usually easier to reprogram the device service routines than to perform major surgery in the input/output structure of the larger system. This is particularly true in connection with such large systems as OS/360 and TSS/360, where the system generation and device support maintenance tasks can be such a major effort. This has been done in MTS with highly satisfactory results. It has been convenient in the development of the Data Concentrator to prescribe the behavior of the equipment so that it appears very similar and can easily be made compatible with that of the IBM 2702 and 2703 Transmission Controls. In principle, no System/360 reprogramming is needed, although in order to achieve special operational features impossible with the IBM equipment, such compatibility has been intentionally downgraded.

Several operational features are possible using the programmable control unit approach that are either impossible or very unwieldy using the conventional approach. For instance, it becomes possible to recognize the type of terminal which has called a particular data set by sending interesting control-answerback sequences and to dynamically reconfigure the character code translation and line control accordingly. In the Data Concentrator, an incoming call is recognized as originating from either a Model 33/35 or a Model 37 Teletypewriter, an IBM 2741 Communications Terminal or a 1050 Data Communication System in this fashion. Thus, only one trunk-hunting set of telephone numbers is necessary for all four types of equipment. In the same vein, if the System/360 duplex system is partitioned as two independent systems, the control unit can operate with both of them simultaneously, with some of the attached terminals communicating with one system and the remainder with the other. Error recovery can be standardized and streamlined so that large unit-check routines and their page-in delays can be avoided. Also, the use of asynchronous interrupts removes the necessity of "hanging reads" and time-outs on each communication line, and in these cases the residency requirements of the System/360 input/output device support routines and their buffers can be relaxed.

As an example of the economies of the programmable control unit approach, the following figures, obtained under typical operating conditions, can be demonstrated: the fraction of time spent in supervisor state for batch jobs run in the Michigan Terminal System (MTS) is 23.8 percent. The fraction spent in supervisor state for terminal jobs connected through the IBM equipment ranges from 40.5 to 79.9 percent, while the fraction for terminal jobs connected through the Data Concentrator is 33.3 percent.

8.2 IBM Terminal Control Equipment

IBM terminal control architectural philosophy has been to provide minimum complexity in the hardware and rely on the processor control program to perform the special operations unique to each terminal type. The terminal control itself is usually capable of performing all of the operations necessary to serialize and deserialize the individual characters, to compute checksums, perform case-shift operations and to recognize a fixed set of record-ending conditions. The terminal control usually does not translate the internal EBCDIC system interchange code to or from the particular interchange code used by the terminal itself, nor does it edit the character stream in a nontrivial way. All of these operations are performed by the processor control program.

These are three terminal control devices presently offered by IBM as part of the System/360 product line: the 2701 Transmission Adapter, the 2702 Transmission Control and the 2703 Transmission Control. These devices are distinguished principally by the number and type of terminals serviced. Several terminal types can be connected to any of these three devices and can be supported by the same or very similar control programs.

The 2701 Transmission Adapter [I1, I2] is constructed entirely of logcomponents, which perform all of the serial/deserial, buffer storage and control functions. A separate set of logic components is required for each terminal connected no common equipment other than power supply and and cable drivers is shared between the terminals. Up to four terminals can be attached to the 2701, including start/stop, serial-synchronous and special adapters.

The 2702 Transmission Control [I3, I4] can service up to 31 terminals of the start/stop type, including the IBM 2741 and 1050 and the AT&T 33/35. A sonic delay line is used for temporary storage in the serial/deserial and buffer operations.

The 2703 Transmission Control [I4, I5] can service up to 192 terminals of the start/stop type or a mixture of start/stop terminals and serial-synchronous terminals in various quantities. Temporary storage for the serial/deserial and buffer operations is maintained in a special magnetic-core memory integral to the equipment.

8.3 The Data Concentrator

Realizing the need for a highly adaptable transmission control unit to interface varied terminal equipment to the timesharing system, The University of Michigan initiated in 1965 the development of a special control unit to be used in conjunction with the System/360 Model 67. Called the Data Concentrator, this device has been constructed and has become available to members of the campus user community. Its service applications include, in addition to the connection of certain special-purpose communications terminals, the connection of standard Teletype and IBM terminals. The Data Concentrator offers several interesting operational features for all of these terminals, many of which are not possible with the conventional IBM equipment.

The design approach taken in the Data Concentrator has been to nucleate about a small general-purpose computer a number of special-purpose interfaces to the various data set equipment. An integral part of the design is an interface which connects directly to the Model 67 multiplexor channels on one hand and to the Data Concentrator computer on the other. This approach avoids the operational restrictions imposed by an off-the-shelf IBM control unit to interface the two machines. The characteristics of the Data Concentrator supervisory program itself are described in [26]. In [25] the operational features of this system, consisting of message protocol, device operational characteristics and command structure are reviewed. Appendix A contains an overview description of the hardware organization of this machine.

9. REFERENCES

1. Alexander, A.A., Gryb, R.M., and Nast, D.W., Capabilities of the telephone network for data transmission. B.S.T.J. 39, 2 (May 1960), 46 pp.

2. Bartlett, K.A, Scantlebury, R.A., and Wilkinson, P.T., A note on reliable full-duplex transmission over half-duplex links. Comm. ACM 12, 5 (May 1969), pp. 260-265.

3. Baker, P,A., Phase-modulation data sets for serial transmission at 2000 and 2400 bits per second. A.I.E.E. Trans., Part 1 - Communications and Electronics 61 (July 1962), pp. 166-171.

4. Battista, R.N., Morrison, C.G., and Nash, D.H., Signalling methods and receiver for "Touch-Tone Calling." A.I.E.E. Trans., Part 1 - Communications and Electronics, (March 1963), pp. 9-17.

5. Bodle, D.W., and Gresh, P.A., Lightning surges in paired telephone cable facilities. B.S.T.J. 40, 2 (March 1961), p. 547.

6. Brady, P.T., and Helder, G.K., Echo suppressor design in telephone communications. B.S.T.J. 42, 6 (November 1963), p. 2893.

7. Breen, C., and Dahlbom, C.A., Signalling systems for the control of telephone switching. B.S.T.J. 39, 6 (November 1960), pp. 1381-1444.

8. Bugbee, L.F., A tone disabler for Bell System 1A Echo Suppressor. A.I.E.E. Trans., Part 1 - Communications and Electronics 80 (January 1962), p. 596.

9. Carter, C.W., Jr., Dickieson, A.C., and Mitchell, D., Application of compandors to telephone circuits. A.I.E.E. Trans. 65 (Suppl.), (1946), p. 1079.

10. Clark, A.B., and Mathes, R.C., Echo suppressors for long telephone circuits. A.I.E.E. Trans. 44 (1925), p. 481.

11. Code extensions in ASCII (an ASA tutorial). Comm. ACM 9, 10 (October 1965), pp. 758-762.

12. Control procedures for data communication - an ASA progress report. Comm. ACM 9, 2 (February 1966), pp. 100-107.

13. Electronic Industries Association Standard RS-232A, interface between data processing terminal equipment and data communication equipment. Engineering Department, Electronic Industries Association, October 1968, 9 pp.

14. Eisenbies, J.L., Conventions for digital data communication link design. IBM Systems J. 6, 4 (1967), pp. 267-302.

15. Elliot, E.O., A model of the switched telephone network for data communications. B.S.T.J. 44, 1 (January 1965), p. 89.

16. Federal Communications Commission, Docket 16979, regulatory and policy problems presented by the interdependence of computer and communications services and facilities. Washington, D.C., May 9, 1969.

17. Fennick, J.H., and Nasell, I., The 1963 survey of impulse noise on Bell System carrier facilities. I.E.E.E. Trans. On Communications Technology COM-14, 4 (August 1966), p. 520.

18. Froehlich, F.E., and Anderson, R.R., Data transmission over a self-contained error detection and retransmission channel. B.S.T.J. 43, 1 (January 1964), pp. 375-398.

19. Hamming, R.W., Error detecting and correcting codes. B.S.T.J. 29, (1950), pp. 147-160.

20. Henry, J.B., Jr., Legalize your phone patch. QST 53, 5 (May 1969), pp. 17-20.

21. Hinderliter, R.G., Transmission characteristics of Bell System subscriber loop plant. A.I.E.E. Trans., Part 1 - Communications and Electronics (September 1963), p. 464.

22. Horton, A.W., The occurrence and effect of lockout occasioned by echo suppressors. B.S.T.J. 17, (1938), p. 258.

23. Johnson, W.C., Transmission lines and Networks, McGraw-Hill, New York, 1950, 361 pp.

24. Lynch, W.C., Reliable full-duplex file transmission over half-duplex telephone lines. Comm. ACM 11, 6 (June 1968), pp. 407-410.

25. Mills, D.L., The Data Concentrator. Concomp Project

Technical Report 8, University of Michigan, May 1968. Also in Proceedings of University of Wisconsin Engineering Institute, December 1968, pp. 1-113.

26. Mills, D.L., Multiprogramming in a small-systems environment. Concomp Project Technical Report 19, May 1969. Also in Proceedings of University of Michigan Engineering Summer Conference, June 1969.

27. Morris, R., Further analysis of errors reported in "Capabilities of the Telephone Network for Data Transmission." B.S.T.J. 41, 4 (July 1962), pp. 1399-1414.

28. Nasell, I., The 1962 survey of noise and loss on toll connections. B.S.T.J. 43, 2 (March 1964), p. 697.

29. Nasell, I., Some transmission characteristics of Bell System toll connections. B.S.T.J. 47, 6 (July-August 1968), p. 1001.

30. Nasell, I., Ellison, C.R., and Holmstrom, R., The transmission performance of Bell System intertoll trunks. B.S.T.J. 47, 8 (October 1968), p. 1561.

31. Nyquist, H., Certain topics in telegraph transmission theory. A.I.E.E. Trans. 47, (1928), p. 617.

32. Peterson, W.W, Cyclic codes for error detection. Proc. I.R.E. 49, 1 (January 1961), pp. 228-235.

33. Peterson, W.W., Error-Correcting Codes. MIT Press, Cambridge, Mass., 1961, 285 pp.

34. Proposed American Standard, Character structure and character parity sense for serial-by-bit data communication in the American Standard Code for Information Interchange. Comm. ACM 8, 9 (September 1965), pp. 553-556.

35. Proposed Revised American Standard Code for Information Interchange. Comm. ACM 8, 4 (April 1965), pp. 207-214.

36. Proposed USA Standard, General purpose alphanumeric keyboard arrangement for information interchange. Comm. ACM 11, 2 (February 1968), pp. 126-131.

37. Proposed USA Standard Data communication control procedures for the USA Standard Code for Information Interchange. Comm. ACM 12, 3 (March 1969), pp. 166-178.

38. Rappeport, M.A., Digital computer simulation of a four-phase data transmission system. B.S.T.J. 43, 3 (May 1964), pp. 927-964.

39. Reference Data for Radio Engineers, Westman, H.P. (Ed.), International Telephone and Telegraph Corp., 1959, 1121 pp.

40. Schleicher, G.P., Phone patching - legitimately. QST 53, 3 (March 1969) pp. 11-17, 41.

41. Shannon, C.E., Communication in the presence of noise. Proc. I.R.E. (January 1949), p. 10.

42. Shaw, T., The evolution of inductive loading for Bell System telephone facilities. B.S.T.J. 30, (1951), p. 149.

43. Townsend, R.L., and Watts, R.N., Effectiveness of error control in data communication over the switched telephone network. B.S.T.J. 43, 6 (November 1964), p. 2611.

44. Transparent-Mode Control Procedures for Data Communication Using the American Standard Code for Information Interchange - a Tutorial. Comm. ACM 8, 4 (April 1965), pp. 203-206.

AMERICAN TELEPHONE AND TELEGRAPH TECHNICAL REFERENCES

A1. Bell System Data Communications Services, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, March 1964, 6 pp.

A2. Data Set Interface Connectors, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, July 1963, 5 pp.

A3. Data Access Arrangement (Preliminary), Technical Reference Manual, Amer. Tel. And Tel. Co., New York, November 1968, 17 pp.

A4. Data Set 103A Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, February 1967, 10 pp.

A5. Data Set 103F Interface Specification, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, May 1964, 6 pp.

A6. Data Sets 201A and 201B Interface Specifications, Technical Reference, Amer. Tel. And Tel. Co., New York, September 1962, 10 pp.

A7. Data Sets 202C and 202D Interface Specifications, Technical Reference Manual, May 1964, 23 pp.

A8. Data Set X203A (M10) Interface Specification (Preliminary), Technical Reference, Amer. Tel. And Tel. Co., New York, August 1967, 11 pp.

A9. Data Set 301B Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, March 1967, 5 pp.

A10. Wideband Data Stations 303 Type, Technical Reference, Amer. Tel. And Tel. Co., New York, August 1966, 20 pp.

A11. Data Sets 401A and 401E, Technical Reference, Amer. Tel. And Tel. Co., New York, April 1966, 9 pp.

A12. Data Set 401H Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, July 1965, 9 pp.

A13. Data Set 401J Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, September 1965, 9 pp.

A14. Data Sets 402C and 402D Interface Specification, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, November 1964, 21 pp.

A15. Data Sets X403B and X403C Interface Specification (Preliminary), Technical Reference, Amer. Tel. And Tel. Co., New York, April 1966, 11 pp.

A16. Data Set 403D 403E Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, November 1967, 21 pp.

A17. Data Auxiliary Set 801A (Automatic Calling Unit) Interface Specification, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, March 164, 6 pp.

A18. Data Auxiliary Set 801C (Automatic Calling Unit) Interface Specification, Technical Reference, Amer. Tel. And Tel. Co., New York, September 1965, 2 pp.

A19. Error Control Unit Interface Specification, Technical Reference Manual, Amer. Tel. And Tel. Co., New York, January 1964.

INTERNATIONAL BUSINESS MACHINES TECHNICAL REFERENCES

I1. 2701 Data Adapter Unit Component Description, Form A22-6864, I.B.M., White Plains, N.Y., 1969.

I2. 2701 Data Adapter Unit Original Equipment Manufacturer's Information, Form A22-6844, I.B.M., White Plains, N.Y., 1969.

I3. System/360 Component Description - 2702 Transmission Control, Form A22-6846, I.B.M., White Plains, N.Y., 1969.

I4. 2702/2703 Transmission Controls Original Equipment Manufacturer's Information, Form A27-3012, I.B.M., White Plains, N.Y., 1969.

I5. System/360 Component Description - 2703 Transmission Control, Form A27-2703, I.B.M., White Plains, N.Y., 1969.

I6. IBM 1050 Data Communications System Principles of Operation, Form A24-3474, I.B.M., White Plains, N.Y., 1969.

I7. IBM 1050 Data Communications System Original Equipment Manufacturer's Information, Form A24-3143, I.B.M., White Plains, N.Y., 1969.

I8. IBM 2741 Communications Terminal, Form A24-3415, I.B.M., White Plains, N.Y., 1969.

I9. 2740/2741 Communication Terminal Original Equipment Manufacturer's Information, Form A27-3002, I.B.M., White Plains, N.Y., 1969.

I10. 2780 Data Transmission Terminal - Component Description, Form A27-3005, I.B.M., White Plains, N.Y., 1969.

I11. 1403 Printer Component Description, Form A24-3703, I.B.M., White Plains, N.Y., 1969.

APPENDIX A. THE DATA CONCENTRATOR - TECHNICAL DESCRIPTION

The Data Concentrator is constructed of a number of special-purpose input/output interfaces, all connected to a Digital Equipment Corporation PDP-8 computer. The basic PDP-8 computer itself contains a 12-bit central processor together with a 4096-word bank of 1.5-microsecond core memory. The basic machine includes a Model 33ASR Teletypewriter and its control unit as the basic input/output device. For the Data Concentrator application, three additional 4096-word banks of core memory have been added to the basic machine, along with a high-speed paper tape reader (300 char/sec) and punch (60 char/sec) for convenience in program development. Hardware for high-speed multiply-divide (Extended Arithmetic Element) and automatic power restart has been field-installed on this machine. Several small modifications have been made in the processor to provide for special behavior in connection with some of the interfaces.

The special-purpose input/output interfaces attached to the Data Concentrator comprise most of the bulk hardware in the system. These interfaces connect the system on one hand to both IBM 2870 Multiplexor Channels used on the System/360 Model 67 duplex system and, on the other, to AT&T data sets of the 100, 200, 400 and 800 series, providing transmission rates on the switched telephone network to 2000 baud. The various interfaces operate both on a core memory cycle-steal basis for data transmission and on a programmed accumulator-transfer basis for control operations. Liberal use is made of the interrupt facilities in conjunction with special line-adapter scan equipment which has the capability of selectively disabling the interrupt facilities by blocks of devices. Figure A1 shows the interconnection of the various major components in the system, which are described in detail in subsequent sections. Reference A1 contains a more detailed description of this equipment than is presented here.

Data Multiplexor and Clock Generators. A special PDP-8 core memory data multiplexor has been constructed to satisfy the requirements of the various interfaces using cycle-steal operations. This multiplexor operates using a common-bus distribution system for memory data transfer in both input and output directions. Each device attached to these busses provides transmission gates between its registers and the busses as required, and conditions data transmission by signals distributed by the data multiplexor separately to each device.

Included in the data multiplexor itself is a set of buffers which distribute the various PDP-8 input/output busses to the scan control and data set line adapter equipment attached to the system. Also included is a set of crystal-oscillator clocks and associated countdown circuitry which provides timing signals for the data set line adapter equipment. Signals of all popular transmission bit rates are provided from 45.45 baud (Teletype Models 28 and 32 attached to AT&T type 103E data set) to 2000 baud (AT&T type 201A data set). In addition, an interval timer using a basic clock rate of 100 Hz provides a maximum countdown interval of 40.95 sec. The timer is implemented using a memory-increment cycle-steal operation in PDP-8 memory. A timer overflow is signaled by a program interrupt.

```
                                 line adapters

                                     +-----+
                                  +--| 103 |  (16)
                                  |  +-----+
                                  |
                                  |  +-----+
                                  +--| 202 |  (4)
mpx chan                          |  +-----+
                                  |
+-----+  +-----------+  +------+  |  +-----+
|360-1|--|           |  |      |  +--| 201 |  (4)
+-----+  |   PDP-8   |  |scan  |  |  +-----+
         |           |--|      |--+
+-----+  |   (16K)   |  |ctrls |  |  +-----+
|360-2|--|           |  |      |  +--| 403 |  (2)
+-----+  +-----------+  +------+  |  +-----+
                                  |
                                  |  +-----+
                                  +--| 801 |  (4)
                                     +-----+
```

Figure A1. Data Concentrator

System/360 Interface. The System/360 interface connects directly to the channel-control unit interface cables which interconnect the multiplexor channels and the various IBM control units to which in turn are attached such input/output devices as card readers, line printers, and communications devices. Usually, several devices of the same type share a common control unit, which serves both as a relay station between the channel and each device for command and status byte processing and as a temporary buffer storage medium for their data streams. This channel-control unit interface normally operates in a byte-interleaved multiplex mode in which data bytes are transmitted one at a time to each control unit using separate signal sequences for each byte. Thus, several devices may transmit data records in an overlapped fashion at relatively slow transmission rates to about 110 kilobytes/sec. The transmission sequences are arranged into three classes: those intended to transmit commands from the channel to a control unit, those intended to transmit status information from a control unit to the channel, and those intended to transmit the data bytes themselves between the channel and the control unit.

Each of the two System/360 interfaces appears to its respective IBM channel-control unit interface as a standard control unit, and processes commands and supervises data transmission for a set of 64 unit addresses. Each interface, containing buffer registers, parity generators, and sequencing circuitry, provides temporary storage and isolation between the channel-control unit interface and the PDP-8. The construction is such that channel-command sequences can be processed concurrently with data transmission sequences and such that these operations do not require program intervention on the part of the PDP-8 until the entire sequence has been concluded. The various interface control operations are supervised via the PDP-8 accumulator and programmed input/output operations. Transmission of both data and status bytes is via the three-cycle data break facility using a type of cycle-steal operation in which the word count and current address registers are implemented in core memory. Transmission rates can be as high as 80 kilobytes/sec., depending upon the instantaneous loading conditions imposed by the other control units attached to the channel. The interface is non-overrunable, that is, data cannot be lost if the channel temporarily refuses data byte service due to a pending service request by a higher-priority control unit. The construction of the System/360 interface is described in detail in Reference A2.

<u>Line Adapter Scan Controls.</u> All of the data set line adapter equipment is connected to a special interface called the scan controls. This interface connects on one hand to the PDP-8 accumulator, interrupt and cycle-steal facilities, and on the other to a set of transmission busses that is common to all line adapter equipment. The various line adapter service requests are processed for each adapter in turn by a high-speed scanner-control circuit which uses a seven-bit counter called the scan address register. The contents of the scan address register are distributed on a common bus to each line adapter, which decodes its address consisting of the high-order six bits. Thus, each of the

possible 64 line adapters decodes two addresses - one with the low-order bit equal to a zero (receiving address) and one with the low-order bit equal to a one (transmitting address).

If a line adapter requires scan service, its circuitry generates a signal to the scan controls when one of its addresses is decoded on the scan address bus. This signal causes the scanner to stop while the service requirements are met. If no further action is indicated at the conclusion of the service operation, the scanner automatically restarts. If a character-assembly complete condition is recognized, the scanner is not restarted and a program interrupt is requested. The scanner is then restarted by a program command.

Both the character assembly and line control functions of each line adapter are controlled through the same scan address. Associated with each line adapter are one or more 12-bit control registers which perform functions unique to each line adapter. The various bit positions in each of these control registers are accessed through a programmed procedure which first requests the scanner to stop scanning the line adapters, then waits for a possible pending service request from a line adapter to be cleared, and finally sets the scan address register to the required line adapter address. Following the control register operations, the scanner is restarted by a program command.

The 64 line adapters addressable through scan controls are divided into eight blocks of eight line adapters each. Each block of eight can be separately enabled or disabled for interrupt requests using an eight-bit mask register called the scan priority register. Using properly constructed programmed operations it is possible to mask off interrupts from a particular block of low-speed low-priority line adapters while one of its members is being serviced, yet while allowing interrupts to be serviced from higher-priority blocks.

Two types of line-adapter service requests can be processed by the scan controls. One type, used for serial-synchronous line adapter equipment, involves no other operation than stopping the scanner when a program interrupt condition due to a character-assembly complete signal is generated within the line adapter itself. The other type, used for start/stop line adapter equipment, involves an interesting cycle-steal-shifting operation in PDP-8 core memory. This operation is initiated whenever inbound or outbound bit service is required at the line adapter. When this occurs a special cycle-steal operation is requested by the scan controls. This cycle, when granted, proceeds starting with a core memory read cycle, followed by a one-bit rightwise shift of the memory buffer register, and ending with a core memory write cycle. The fifteen-bit core memory address at which this operation takes place is formed by the scan address register as the low-order seven bits and by an eight-bit register called the scan page register as the high-order eight bits. Thus the serial/deserial operation of each line adapter is implemented in two words of PDP-8 core memory, one for the transmit operation and the other for the receive operation. As it happens, the PDP-8 memory buffer register is already provided with shifting circuitry in connection with an analog-digital converter option. A few logic components have been pirated from unused options for the purpose of interfacing this shifting circuitry to the scan controls.

A character-assembly-complete signal is generated by the equipment for each character when the serial/deserial process has been completed. For serial-synchronous line adapters, this signal is generated in the line adapter itself when its self-contained serial/deserial register contains a fully assembled character during a receive operation or has completely emptied during a transmit operation. This signal stops the line adapter scan process, following which the line adapter generates a standard cycle-steal operation which transmits a character between the serial/deserial register and the addressed core memory location. For start/stop line adapters, the character-assembly-complete signal is generated in this fashion: on transmission when, during the shifting of the right-hand bit of the serializer word at the addressed core memory location to the line adapter transmit bit buffer, an all-zero word is detected; on reception when, during the shifting of a bit from the line adapter receive bit buffer to the left-hand bit of the deserializer word at the addressed core memory location, a one-bit is shifted out of the right-hand end of the same word. Thus, the various frame sizes of the several start/stop transmission codes can be accommodated by loading the proper mask bits in the core memory serial/deserial words prior to the character assembly/disassembly operation.

<u>Line Adapters.</u> A line adapter is associated with each data set serviced by the scan controls. It contains the circuitry to interface the control and data transmission characteristics peculiar to each data set to the common-bus interface of the scan controls. Depending upon the characteristics of each data set, the line adapter contains none, a single bit, or a full character of buffer storage between the scan controls and the data set transmission circuitry. Each line adapter contains either one or two twelve-bit control registers used for the various data set control functions. Several bits assigned in these words

have the same interpretations among all data set types, so that common program servicing routines can be conveniently implemented. The bit/character buffers of each line adapter are accessible only through the cycle-steal operations described previously, while the control registers are accessed through the PDP-8 accumulator using programmed input/output operations. In either case, the scan address register establishes at each instant which line adapter is in fact accessible for the various operations and which core memory location is in fact available for data transfer.

There are currently five different types of line adapters connected to the scan controls. These five types correspond to the five types of data sets connected to the Data Concentrator: low-speed start/stop (AT&T 103E), medium-speed start/stop (AT&T 202C), serial-synchronous (AT&T 201A), Touch Tone digit receiver (AT&T 403E), and the automatic calling unit (AT&T 801C). The 103E data set is a full-duplex start/stop type utilizing frequency-shift modulation for simultaneous transmission rates to 300 baud in each direction. This data set is used for connection of almost all low-speed Teletype and IBM equipment, such as Teletype Models 28/32 (45.45 baud), 33/35 (110 baud), and 37/38 (150 baud), and IBM 2741 Communications Terminal and 1050 Data Communications System (both 134.5 baud). The 202C data set is a half-duplex start/stop type utilizing frequency-shift modulation for transmission rates to 1200 baud. This data set is used for connection of medium-speed data communications equipment utilizing special message conventions. Both the 103E and 202C can share the same type of line adapter in the Data Concentrator, although a special 202C line adapter has been constructed to minimize the system overhead. The 201A data set is a half-duplex serial-synchronous type using four-phase modulation at a transmission rate of 2000 baud. This data set is used for connection of IBM Binary Synchronous Communications devices and in particular for the connection of remotely located PDP-7, PDP-8, and PDP-9 computing facilities to the parent Model 67 system for purposes of remote job entry and graphics processing. The 403E data set is a receiver for Touch Tone multifrequency dialing digits transmitted by the standard telephone instrument equipped with the Touch Tone feature. Included in this data set is an audio answerback channel which enables computer-generated replies to be transmitted to the caller in response to a dialed-digit message. The 801C automatic calling unit is a transmitter for Touch Tone multifrequency dialing digits and is arranged so that outbound telephone calls can be placed in connection with certain data set types. The automatic calling unit incorporates circuitry for the recognition of the appropriate data set type at the called terminal so that the transmission circuit can be verified before use.

The low-speed start/stop line adapter contains two single-bit line buffers, a clock frequency selection matrix and various clock countdown and control circuitry. Four clock frequencies are distributed to all start/stop line adapters on a common bus at eight times the basic transmission rates of 45.45 baud, 110 baud, 134.5 baud, and 150 baud. The clock selection matrix circuitry within each line adapter determines which clock frequency is active at any time and allows dynamic reconfiguration during the terminal recognition phase of system operation. Both transmit and receive operations are completely independent except in choice of clock frequency, which is common to both operations. The control circuitry provides program monitoring of all data set interface lines and, in addition, program control over the lines used to enable the data set for automatic answering and to sieze an automatic calling unit. Indications of data set and control circuitry faults such as character and bit service overrun are available to the program. It is possible under program control to recognize both the break signal and the space disconnect signal, each consisting of various periods of space-signal transmissions, and to generate either of these signals.

Using the standard start/stop line adapter with the 202C data set places a very severe strain on the interrupt system. The reason for this is that only about 800 microseconds is available for the program to respond to the character-assembly complete signal and to restart the line adapter scanner. In order to relax this requirement a special medium-speed start/stop line adapter has been constructed which operates in the same fashion as the 103E start/stop line adapter, except that four core memory words are assigned instead of the usual two. On receive operations a swing-buffer technique is used, so that while the program is servicing one core buffer the line adapter may be assembling the next character into the other. Such an operation requires that the scanner be left running during the interrupt-processing time but disabled for character service. In this manner character serial/deserial operations proceed normally but the character-assembly complete signal is inhibited from stopping the scanner.

The serial-synchronous line adapter contains a twelve-bit serial/deserial register, a four-bit frame counter, a four-bit frame size register, a parity generator, and various control and synchronization circuitry. The serial/deserial register is used as the primary character assembly/disassembly element and the core memory words at the receiving and transmitting scan address locations are used as backup buffer registers. Since only one serial/deserial register is implemented in this line adapter, transmit and receive operations may not occur simultaneously. The frame counter is loaded from the frame size register just before the first bit is transmitted between the serial/deserial register and the data set. Thus the frame size register, which is accessible from the program, establishes the number of bits per character (up to twelve) in the data stream. This adapter contains circuitry for the automatic synchronization of character frames using a special synchronization character which can be established, at least in some line adapters, by the program. A parity generator provides an odd vertical parity bit for each transmitted character. A corresponding detector verifies the odd parity condition on each received character. These parity features may be enabled and disabled by the program.

As in the start/stop line adapter case, the control circuitry provides program monitoring and control over all of the data set interface lines, including those to sieze an automatic calling unit. Indications of data set and control circuitry faults such as character service overrun and vertical parity check are available to the program. All of the serial-synchronous line adapter equipment is connected at present to a 2000-baud clock frequency source for use with AT&T 201A data sets on the switched telephone network. However, these same line adapters can be operated at much higher clock frequencies up to the limit of the logic components themselves, about two megabaud. The construction of the serial-synchronous line adapters is described in detail in Reference A4.

The 403E Touch Tone digit receiver line adapter and 801C automatic calling unit line adapter are simple devices with the capability of a single-character parallel transfer between the scan control register data busses and the data set. Neither of these adapters makes use of the cycle-steal facilities in the fashion of either the start/stop or serial-synchronous line adapters. The 403E Touch Tone digit receiver adapter provides a character-assembly complete signal and a program interrupt when a dial digit has been detected; the 801C automatic calling unit adapter causes a bit to be set in its control register when the data set is ready to receive the next dialing digit. At present two automatic calling units have been installed in the system. One of these is switchable over the sixteen 103E data sets connected to the start/stop line adapters; and the other is switchable over the four 201A data sets connected to the serial-synchronous line adapters. The association of a particular line adapter to one of these automatic calling units is by a program-selected bit in the control register for each line adapter. The automatic calling unit remains selected to the line adapter until the attached data set signals that the call is complete, after which the automatic calling unit can be selected to a different line adapter.

Auxiliary Equipment. A number of special-purpose input/output components have been added to the Data Concentrator for utility purposes. One of these is a programmable audible alarm, consisting of an extraordinarily obnoxious siren, which is used to attract the attention of an equipment operator for the purpose of manual intervention. A scurrilous programmed-timeout operation has been implemented within the current supervisor which sets off the siren at noon and midnight, presumably to alert the machine-room crew for a shift change.

In connection with binary synchronous communications a requirement exists to calculate a special 16-bit checksum using a procedure involving shifting and modulo-two bitwise addition. A subroutine to implement this procedure requires a calculation time of about one-half millisecond per character. A special hardware adapter has been constructed which implements this procedure using a 24-bit shift register and a set of modulo-two adders. Using this adapter, the calculation time, including shift-register loading and storing, is about twenty microseconds per character. This device is described in reference A3.

REFERENCES FOR APPENDIX A

A1. Mills, D.L., *The Data Concentrator.* Concomp Project Technical Report 8, University of Michigan, May 1968. Also in Proceedings of University of Wisconsin Engineering Institute, December 1968, pp. 1-113.

A2. Mills, D.L, *System/360 Interface Engineering Report,* Concomp Project Memorandum 13, University of Michigan, March 1965.

A3. Burkhalter, K.E., Jr., *A Cyclic Check Computer for Error Detection,* Concomp Project Memorandum 19, University of Michigan, June 1968.

A4. Wood, D.E., *A 201A Data Communication Adapter for the PDP-8: Preliminary Engineering Design Report,* Concomp Project Memorandum 15, February 1968.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* THE UNIVERSITY OF MICHIGAN CONCOMP PROJECT	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

TOPICS IN COMPUTER COMMUNICATIONS SYSTEMS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report 20

5. AUTHOR(S) *(First name, middle initial, last name)*

DAVID L. MILLS

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1969	106	74

8a. CONTRACT OR GRANT NO. DA-49-083 OSA 3050 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report 20 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Qualified requesters may obtain copies of this report from DDC

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency

13. ABSTRACT

This report surveys several aspects of data communications systems applicable to timesharing utilities. Topics presented include propagation characteristics of the telephone network and and comparisons of the various modulation techniques suitable for use on this network. A description of several types of coding systems is presented and various aspects of these systems are studied, including those of symchronization and graphic/control-function interchange assignments. A detailed description of a proposal for a half-duplex processor-processor communication link protocol is presented to illustrate the problems in a practical design. Finally, a brief description of the operation of certain types of popular terminals and terminal control units is presented, along with a description of a special terminal control unit constructed by the University staff. An extensive bibliography is included.

DD FORM 1 NOV 65 **1473**

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Timesharing						
Data Communications						
Error-Correcting Code						
Transmission Code						
Interchange Code						
Half-Duplex System						
Modulation/Demodulation						
Data Set						
Terminal						
Terminal Control Unit						
Switched Network						

Unclassified
Security Classification