**National Physical Laboratory**

# Proposal for a Digital Communication Network

**by**

**D. W. Davies**

**June, 1966**

# INDEX



**Figures**

# PROPOSAL FOR A DIGITAL COMMUNICATION NETWORK

## Introduction

In this report an outline case is made for the development of a new kind of national communication network which would complement the existing telephone and telegraph networks. In order to establish such a case on a firm basis three things will be needed. Firstly, an analysis of the communication needs which will be met by the network. Secondly, a description in detail of the facilities which the network will offer. Thirdly, a description of the method by which the network will be implemented, in sufficient detail to demonstrate its technical feasibility.

Not enough work has been done to provide in detail any of these three things, but the report contains arguments which it is hoped, will justify further research and development leading to a firm basis for implementation to begin.

To avoid misunderstanding it should be stated at the outset that, although the new network is described throughout the report as though separate from the existing networks it will certainly share digital transmission equipment with them, and will share buildings and local lines in some cases.

This report is based on lecture given at the National Physical Laboratory on March 18th, 1966. It represents the author's views and not necessarily those of the N.P.L. or the Ministry of Technology.

There is a summary of conclusions at the end of the report.

## Characteristics of the existing networks,  contrasting with the proposed one

Features of the existing telephone and telegraph networks which contrast with the proposed network will be described.

(a) Synchronous Transmission.

In passing from the source to the destination, the signal is delayed by a time which is constant for all parts of the signal. This feature is a consequence of the way in which these networks operate which consists of switching the transmission paths so that a path is formed between source and destination. The effect of this constant delay is that the received signal is a replica along the time axis of the signal that was transmitted. This preservation of the time axis of the signal is essential for speech, but not at all necessary for telegraphy.

An alternative method of handling telegraphic signals, and indeed any digital signals, is to store them at various points in the network and retransmit at a later time. “Store and forward” systems, often called by the misnomer “message-switching”, are employed in special circumstances, but not in the public Telex network.

(b) Design for human use and speed.

The original design of telephone and telegraph networks was based on the human operator (though alternative uses now exist such as data transmission by telephony). Each of the networks is designed for a predominantly uniform kind of terminal equipment.

(c) Single interface for transmitted information.

Here we must distinguish the content of the transmitted information, whether voice or digital, from the control information. Control information is exchanged with the network itself, and includes such things as dial pulses, tones, and ringing current. In both existing networks, the transmitted information has an interface which is substantially the same in all terminals.

(d) Control information format which is often incompatible with the transmitted information.

**The significance of the real time use of Computers**

We shall give some attention in this report to the on-line and multi-access use of computers for two different reasons.

(a) Realtime use of computers gives rise to many of the communication requirements which the network will satisfy. This category includes multi-access computers.

(b) The methods used in multi-access computers for their internal organisation illustrates, by analogy, some of the design features of the proposed communication system.

Multi-access computers have been exploited most fully in scientific research. Their use in commerce is most familiar in some specialised applications such as airline seat booking. The main characteristics of the scientific and recent commercial systems are similar and may be described briefly as follows:

(a) The machine interacts simultaneously with a number of slow peripherals. These are mainly designed for human operation and include keyboards, slow printers and displays.

(b) The computer shares its processing power with many users by working for them in succession in small time slots, so that in the absence of saturation each user gets a reasonably quick response no matter what other demands are being made on the computer's time.

(c) The effect of this time-sharing is to allow a conversation between the user and the computer, at speeds suitable to the user, and this proves to be one of the most significant features, giving the system great versatility. One effect of this conversational mode of operation is that most interactions involve very small amounts of computer time.

(d) The system includes a large store. To combine large size with (for most purposes) short access time, this store is arranged as a hierarchy of stores differing in size and speed. This large store is a key feature for most applications. It is important to note that this kind of store hierarchy is not exclusively associated with multi-access but can be found in batch processing systems as well.

(e) The system is provided with a comprehensive set of commands. In addition to giving it programs and data, the user must be able to tell the computer system what functions he wishes it to perform with them. For example, he may wish to compile, edit, obey or dump a program. This information is given by a set of commands, analogous to instructions, but addressed to the "operating system". Advanced operating systems for batch-processing computers, as well as multi-access systems may use a command language.

**Real-time business computer systems**

The terms "real-time", "multi-access" and "on-line", on the one hand and "batch-processing" on the other are widely used without careful thought about the real distinctions. In fact, payroll, a classic example of batch-processing is real-time in that it is tied to a time schedule and it may even have on-line features such as the collection of data.

The real distinction being sought after is one of response time. The response time is the elapsed time between the last necessary piece of data becoming available for a given transaction and the whole result becoming ready for output. (This method of definition avoids adding input and output times to the response time, when these processes are slow.)

The response time for scientific computers with conventional batch-processing operating systems is typically an hour or two, and batched business systems are of comparable or rather worse response time. The distinction between these and real time systems is not usually difficult because the response times of the latter are measured in seconds.

The various reasons why short response times have been sought after are related to the needs of a person interacting with the system. Some examples which come to mind are:

(a) A client at a service counter who must be dealt with in a short time.

(b) Someone using the computing or information service who wants the answers while he is still thinking about the question, thus avoiding the need to share his attention between a number of tasks.

(c) A conversation between a computing (or information system) and a human user which can save the human effort. The user need not formulate the problem or enquiry completely and in a standard form before starting to send it. Errors and omissions in the information can be dealt with step-by-step.

In a conversation it has been found that there is a critical time, near to 10 seconds, beyond which the user will tend to do something else to fill in time and thus spoil the effectiveness of this mode of operation. For present-day printers, printing responses of one line of type, which might take five seconds, the printing time is not objectionable and is not felt as a delay because something is happening. These are only rough rules of thumb and should be refined by experiment.

The range of response time between 10 seconds and one hour seems to be unexplored. For human interaction it does not deserve the term “real-time” which, according to the above rough arguments, should be reserved for responses in less than 10 seconds.

The emphasis on real-time business systems in this report is due to the belief that they will generate more real-time digital communication traffic than, say, scientific calculations or computer-aided design.

The commercial application of computers usually involves transactions of a relatively simple kind carried out in direct response to a human request. Much of the ingenuity employed in systems analysis and design for these applications is devoted to making up for the mismatch between the human user and the computer though this point is not very obvious.

The computer, working batchwise, gives a very poor service for the majority of potential applications, which results in their not being carried out on a computer at all. Human errors give rise to complicated procedures for segregating and rerunning the transactions in which they occur, long after the errors have been made.

The feature of conversation between the computer and user, preferably in short messages, will immeasurably widen the application of computers, so that batch processing will become less important. The spread of these on-line applications, involving big organisation changes, will take place slowly and it is very difficult to predict a time scale.

The commercial transaction typically involves several "authorities" which have different functions and keep their own file stores. For example, a purchase may involve the supplier, purchaser, transport agents, transporters and banks in these transactions:

(a) Availability and price information.
(b) Conversation concerning alternative specifications.
(c) An order and confirmation of order.
(d) Delivery advice and invoice.
(e) Transport booking.
(f) Payment.
(g) Cheque clearing or credit clearing.

This list is not a complete breakdown. For example, transport booking alone may involve conversation between several authorities' computers.

Typically, therefore, each human action may lead to several computer to computer conversations.

The great majority of potential commercial computer applications will never be embarked upon with the present inflexible and costly communication systems. They are costly when used for digital data transmission when all the additional equipment and the poor usage of lines is considered, and this is because they were designed for different purposes.

## The Organisation of the Multi-Access Computer System

Some features of the way in which multi-access computers work will be described here, because they illustrate in a more familiar setting features of the communications system to be described.

Multi-access systems must provide for interaction with slow peripherals, and in Fig. 1 a time diagram is given for an interaction involving input from a keyboard computation and output via a slow printer. The key depressions shown on the left each give rise to an 8 bit character. The main computer is not usually involved in recording each of these key depressions but they are handled either by a "multiplexer" or a separate input/output computer. They are stored in a buffer until a particular character (in this case shown as "newline") indicates that a message is ready for attention by the computer and a response is expected. A restriction is introduced here, in that interaction is not on a character-by-character basis. This restriction is found to be acceptable, though in certain circumstances interaction in shorter units than one line of type may be desired.

When a message ready for attention by the computer has been received it is transferred to the computer store by an interrupt and, in the case we are studying, it is assumed to give rise to a demand for processing. The processing for this user is shown as taking place in four short bursts at the end of which the process is finished. At this time the subsidiary computer or multiplexer is alerted and will take a message into the buffer ready for distributing to the printer. We assume here that two lines of typed output are needed. The buffer contains one line at a time, therefore part of the output must remain in the computer store. On the right the transfer at a regular rate from the buffer out to the typing mechanism is shown. The assembly of messages from slow input peripherals and their distribution to slow output peripherals can in this way be carried out almost independently of the main computer. The main computer need only be concerned with the handling of complete messages, though in some cases (as in output side of Fig. 1) it will divide the complete message into more than one part for convenience.

The second feature of the multi-access computer system is the way in which it allocates processing time to the various jobs demanding it. It does this by giving them time-slots according to some allocation algorithm. The end of each time-slot, involving swopping from one job to another, often entails an overhead due to moving data about in the store. Even if the main store is very large and can accommodate several jobs there may be faster, smaller stores which have to be emptied and refilled. This overhead makes it inefficient to divide the time into very small slots. If, on the other hand, long jobs are allowed to run to completion, they will give rise to unsatisfactory delays for very short jobs which have to wait. Fig. 2 shows a simple example of different allocation methods applied to four jobs of lengths 1, 2, 3 and 4 units respectively. The best method is to run each job to completion always selecting the shortest remaining job.

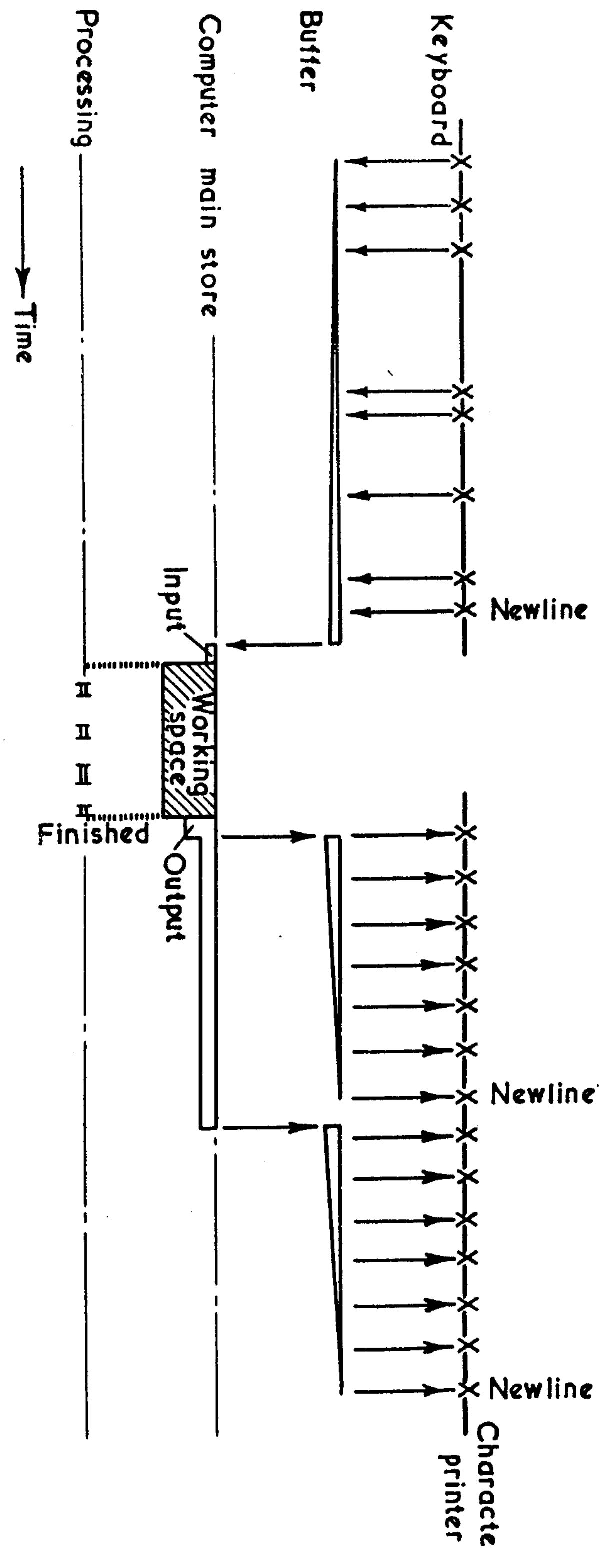


FIGURE 1. INTERACTION OF KEYBOARD, COMPUTER AND PRINTER

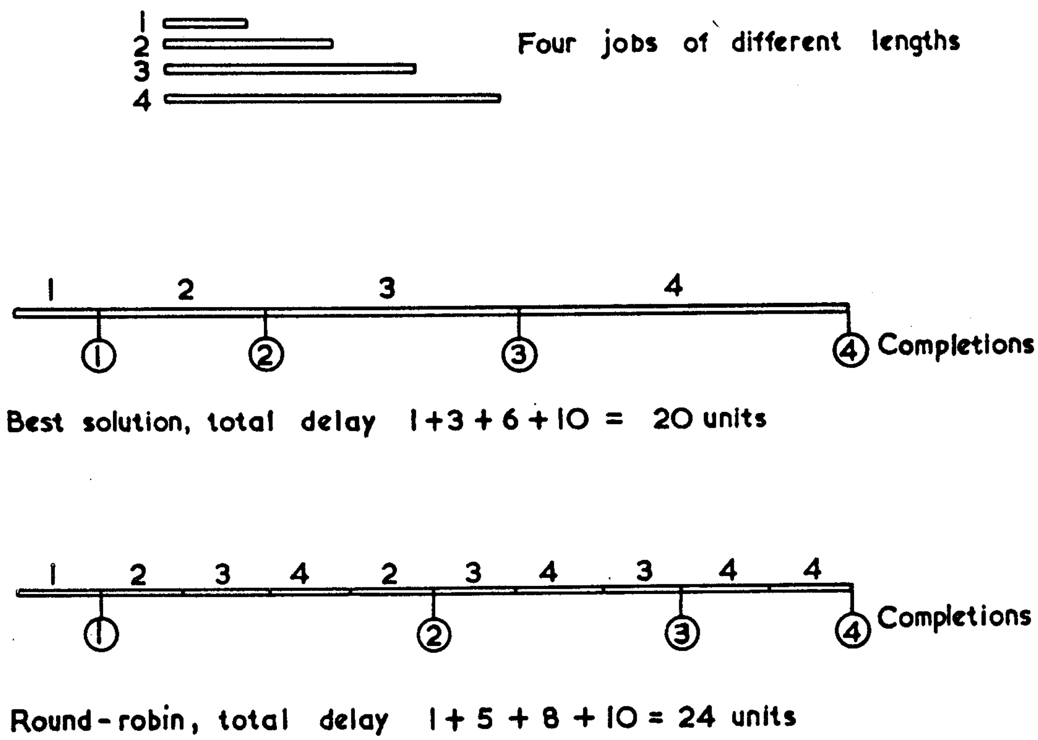


FIGURE 2. EXAMPLES OF ALLOCATION OF COMPUTER TIME

This leads to a total delay over all the jobs of 20 units, 4 swops between jobs being incurred. A simple "round robin" technique applied with time slots of unit length is also shown in the second part of Fig. 2 and gives a total delay of 24 units with 10 swops.

It appears in practice that elaborate allocation algorithms such as were used in the first version of project MAC (CTSS) have no great advantage over "round robin" schemes. The correct choice of the length of time slot is very important and a rough rule would be that the length should be chosen so that a small proportion of jobs take less time than one slot.

A third feature of multi-access systems is the way in which they organise their allocation of store. The user is encouraged to break up his demands for stores into "segments" which are the divisions natural to the way he is using the computer. The operating system further sub-divides the segments into pages of roughly uniform size. The page is the unit for allocation, but the paging scheme is not apparent to the user. This has analogies in the way the communication system sub-divides messages.

## Properties desired in a digital communication network

Some of the aims of a digital communication network designed for use in conjunction with real time computer systems will be described here. Whether they can be achieved in an actual system can be determined only by detailed design of the system. This report will go part of the way towards showing that they can be achieved economically.

(a) The property implied by the concept of a network is that each terminal can send information to each other. There are two very different kinds of terminals attached to this network; human users' consoles or enquiry stations working at very slow speed and real time computers working at high speeds. A property of the network must be that it deals efficiently with terminals working at any speed within a wide range. The information coming from a keyboard, for example, can be sporadic or variable without incurring unnecessary cost to the user. However, the highest efficiency for the slowest user is not a primary aim.

Since terminals at different speeds must be connectable via the system the storage of messages in the system is essential.

(b) Non-synchronous transmission, which is a consequence of connecting terminals that work at different data rates.

(c) Control information compatible with message content. This means that information from the user to the communication system, telling him what functions he is requesting should be in a form compatible with message. Information from the communications system can go to the user in a similar form advising him of operational matters such as congestion. This property of the system simplifies the interface equipment and is necessary for the next property to be provided.

(d) A flexible command language. The user will be able to instruct the communications system to do a number of different things. For example the destination for a number of messages may either be specified before each message or set up in advance by a single command. Some of these facilities which the communication system might provide are described later. Certain of the useful facilities might, in practice, be provided by separate computers attached to the network. These could, for convenience, be called into play by commands as though they were part of the communications system. Also, common computation or information facilities could be provided by the network authority as responses to commands.

(e) The delay in transmission through the network may be variable but its statistical distribution should be well defined, and its maximum effectively limited. Typically, the probability of delays above a certain value will fall off exponentially with delay time. A realistic target seems to be 100 milliseconds as a delay time which is very rarely exceeded. With this kind of delay a transaction involving several computers will still give a response satisfactory to the human user.

(f) The cost of using the network, as opposed to fixed charges, should be related more to the quantity of data sent than the time occupied in sending it. Otherwise, the communication network will spoil one of their best features of multi-access services, the lack of pressure on the user to hurry. Of course, if the communication cost was negligible compared with the "computation" cost, it would not matter whether time or quantity of data was the basis of charging.

**Overall Description of the Proposed Implementation**

Fig. 3 shows a hypothetical communication network. It will be seen that there is a central system consisting of nodes marked N connected by links. This system is responsible for carrying the messages through the network and directing them along the right path. The users are connected into the system at various points by the interface units marked *I*. These units might be responsible for one or more computers and possibly for a large number of consoles or enquiry stations. Some of the connections between nodes are shown as dashed lines. These are tributary connections and have the property that a path between any two terminals can only pass through two of these links.

The principle of the system is that information is carried through the network in relatively small units, which are stored at each node through which they pass. The unit in which information is carried must be distinguished from the message as understood by the user. This is like the distinction between "segment" and "page" in a multi-access computer system. The user is aware of the message as the unit of information he wishes the system to carry, but for its own purposes in allocating channel capacity the network may break up the users' messages into smaller units. Smaller units for transmission must be distinguished and we shall call them "packets". Each packet contains, in addition to the information being carried for the user, certain data, amounting to about 16 characters which is needed by the communication system. The existence of this "red-tape" makes it desirable to use the largest possible packet size, but large packets have the undesirable effect that they increase the spread of the response time distribution. A good choice of packet size cannot be made until a system is working and the distribution of the lengths of message it is called on to carry is known. The evidence available from existing multi-access computer systems suggests that the majority of messages will be less than one hundred characters in length. We shall therefore assume for the purpose of this description that the total size of the packet, including the red-tape information, is limited to 128 characters. For messages shorter than this it is not necessary to employ the channel capacity or storage space appropriate to a full sized packet. It seems likely that the system will be used for a large number of messages of extremely small size. For this reason, as will be seen later, the system is designed to handle messages as small as two characters in length without wasting channel time or storage space.

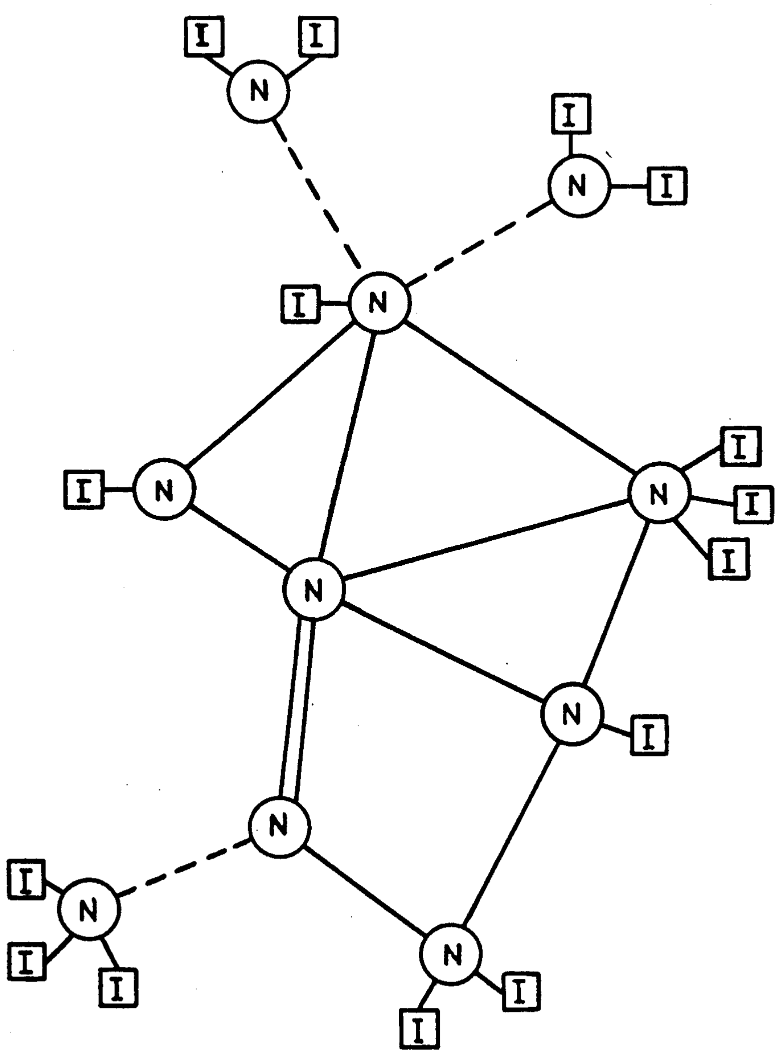


FIGURE 3. A HYPOTHETICAL COMMUNICATION NETWORK

The red-tape information carried in the packet for the benefit of the system, includes a source, destination and route. The source is an indication of the terminal at which the message originated and is inserted in the packet by the communications system. This information is available to the user at the receiving end. This is an essential service for many ways of using the system; it may establish a user's right to employ a certain computation or information to each of the nodes through which the packet will pass to tell it on which link the packet should go out. We choose to insert this information at the originating interface unit, in order to simplify the node computers and to confine the requirement for a large store to the interface units. It is also possible for the interface unit to route related messages by different paths, which may be important for cryptography or for security against breakdown.

The handling of non-standard routing under exceptional conditions is complicated by having the route determined at the source, but the properties of the system allow congestion or failure information to be broadcast quickly, and interface units will be able to amend their route tables.

The method to be used for routing has not been thought out, but one or two indications can be given. Route information can be "packed" so that, for example, the choice of output links at each node could be represented by a four-bit field. As the packet goes on its way, the place reached in the "route" field can be recorded, by shifting the data or by a counter.

The *destination* is used to check the arrival at the correct interface unit and to indicate which terminal attached to this unit is being addressed. It may also help to deal with the packets "lost" between a fault occurring and the route tables being revised. Such packets could, however, go back to their sources and start again.

It is not proposed that serial numbers of messages and time of origin should be added as a universal system feature, but these could be added to the text at the originating interface unit, when requested, by a command.

The interface unit has the most severe task to carry out when it is handling a large number of slow peripherals. In this case it is responsible for assembling the sporadic input from each active user and preparing the packets for transit through the system. It is also responsible for storing messages received from a node for sending to slow peripherals and spelling them out at the appropriate rate. It is believed that the largest cost of the communication system, apart from peripherals, will be in these interface units.

The choice between a completely "store-and-forward" system as compared with line-switching is difficult to justify without designing alternative kinds of system and comparing them. However the requirement we have stated for speed of response and the existence of quite short messages rule out the possibility of line-switching being used exclusively. For a fast line-switching system the routing information would have to be carried through the system by a store-and-forward method and a message would have to return to the user to verify that the route was complete before the information could be sent. For a network covering long distances and for very short messages this is plainly not a practical proposition. Because it connects users having different information rates, the system is necessarily a store-and-forward system, but this does not rule out the use of line switching over some part of the network with storage provided at fewer points.

Line-switching, in a digital system which uses the high data rates available for future communications, amounts simply to reserving certain time slots in advance where the existence of a "virtual path" between two nodes requiring a certain data rate is known. While this is feasible, it seems to add unnecessary complexity to the system without any compensating advantages. The complexity of the interface unit is a consequence of giving the communications system the job of assembling and distributing messages for many slow terminals. The universal use of "store-and-forward" results in a requirement for storage at each node. If, as we believe, the cost of nodal computers is not a large part of the whole system there seems to be very little argument in favour of using line-switching.

Fig. 3 is not a geographical representation of a network and the units *I* and *N* may be grouped in buildings in many different ways. It might happen that several node computers are grouped together in order to overcome a limit upon the number of links each node can handle, or to give some degree of protection against failure of an individual node.

It should perhaps be stated that, like the telegraph system, the network would share buildings and communication plant with the telephone network. Thus, out of a group of channels carrying PCM speech, a few will be allocated at any given time to the digital network. Fortunately, the channel capacity required is small compared with that required for telephony. The increase in trunk telephone traffic and the introduction of digital transmission should enable it to be provided easily.

This overall description of the system shows a major organisational change. Present day multi-access computers each have equipment which assembles messages from keyboards and distributes them to printers. What we are proposing is that this function should be carried out by the network, not the attached computers.

A possible consequence of this change of responsibilities is that consoles might be connected to their local, privately owned computing service via the network in order to give them and the computer the convenience of attachment to other nationally available services. The local service could easily be reserved for in-house use as will be seen later.

**Varieties of Interface**

The interfaces between the network and its various terminals must be well-defined, and there could well be some concern that standardisation would limit the design of the attached devices. For example, airline reservation systems have already generated many kinds of agent's set and special consoles for use as "extended desk calculators" are now appearing.

Consoles designed for communicating with a wide range of computation and information facilities must use a standard character node, though they might have added special features for their one, most frequent application. Any of the characters except "end of message" ones could be used for their special features. Other consoles might be designed for one use only, and these need have no restriction on their coding except in the way they end their messages. It is not necessary for them to be able to give commands or routing information to the system, because this can be permanently stored in the local interface unit. Thus there is very little restriction on their design.

Computers communicating with consoles will use the appropriate character code, but some computer-to-computer transmission will need a "transparent interface", that is, the ability to send unrestricted bit patterns. It is suggested that these should, nevertheless, be in the form of messages of reasonable length, for convenience of error checking and control. A special command from the computer would give the length of the message, and would be followed by the stated number of characters in code-free form. The network would afterwards revert to the standard mode of working, and if more code-free data were to be sent a new command would be needed.

A number of speeds of interface might be needed, but these speeds would be maxima in each case, determined more by the local lines than the network capability. Among the slow keyboard/printer interfaces variants to deal with standard teleprinters and specially designed machines might be needed. There might have to be about ten standard interfaces to cover all purposes.

**Estimation of Response Time**

The response time is the time which elapses from a fully assembled message being available at the source to the same message starting to arrive at the destination terminal. This is determined by the time taken to transmit one packet through the system. For simplicity we shall assume that all packets are of maximum size, 128 characters, which will give an over-estimate of the response time.

The response time is the sum of the time spent in queues in various nodes and the time spent in transit between nodes. We shall assume that the bit rate of a channel is never less than 1.5 megabits per second. This will be a modest rate when PCM transmission of speech becomes common. However, it will be a long time before this rate is available between all points of the proposed network, and it will be found that a useful network can still be made even with considerably less performance. The tributary connections shown in Fig. 3 are significant here because the delays they introduce are added only twice into the total path and can be somewhat larger than delays in standard links.

The maximum time for transmission of a packet through one link, denoted here by *t*, equals 680 microseconds.

Exact calculations of the delay through such a network are difficult to make. There is no point in detailed simulation to get exact results while the basic data on the traffic to be carried is not available. A certain set of approximations have been justified by some simulation work and are stated in a book "Communication Nets" by Leonard Kleinrock, McGraw Hill, 1964. Roughly speaking, the assumptions amount to applying queueing theory to the individual queues in the network as though their behaviour was independent and all their inputs were exponentially distributed. The output from the queue is at a regular rate (provided the queue is not empty). This rate consists of one packet in each time *t*. The length of the queue is determined by the degree of saturation $\rho$ which equals the mean arrival rate divided by the departure rate. The waiting time in the queue measured from the instant of arrival until the beginning of service is given by

$$W = \frac{t}{2} \frac{\rho}{1-\rho}$$

In common with most queueing situations this waiting time increases rapidly as $\rho$ approaches unity. We shall assume that extra capacity is in general provided so as to avoid a degree of saturation greater than 0.8. In these circumstances the mean waiting time is less than 2*t*. It will be seen later that the packet visits two queues at each node and assuming that it visits no more than 5 nodes this means a total of 10 queues.

Allowing for the “service time” which is the time taken to transmit the packet down one link, namely, $t$, the average delay time through the whole system would be $30t$ which is 20 milliseconds.

This result shows there is an ample margin between the estimated performance of the system and the stated requirement of 100 millisecond response. At the same time the response requirement allowed a good margin compared with current multi-access systems. These margins could be used in two ways, during the development of the system, they can be used to accommodate lower data rates than the 1.5 megabits per second we have assumed, and when the system is well developed they can be used to increase the maximum size of the packet.

**The Manipulation of Packets in Queues**

The feasibility of the proposed system depends, among other things, on whether cheap nodes can be designed which work at the speeds we have assumed. Comparison with existing message switching systems shows that a considerable increase in message handling capacity is being postulated. We hope to show that this increase can be achieved. Fur this purpose we must show how packets can be manipulated economically in the node computer.

Fig. 4 shows a suggested format for a packet. This particular packet contains the minimum amount of text namely 2 characters. It is shown as entirely made up from 8 bit characters which are themselves grouped into 48 bit words, but the word grouping is arbitrary and depends only on store economics. The packet itself contains a characteristic (1 character) a heading (16 characters) and a text which can vary in size. The heading comprises the source, destination and route mentioned before. The characteristic gives the size of the packet in units which will become apparent later, and some indication of its priority and purpose. For example, some messages which are communications between one part of the network and the other must be handled in a special way. An example of such “system messages” is a request from one node for the repeat of a faulty message from a preceding node.

In the example in Fig. 4, the packet itself which consists of 19 characters has attached to it further information concerned with its storage in the node computer. We shall assume, for simplicity, that the store in the node computer is allocated in units of 4 words. Therefore the example given occupies one unit of storage. The first character in this set of 4 words contains a code which relates this particular store unit to the other store units holding the remainder of the packet. Packets can be spread over 5 store units and in this case the first unit, the inner ones and the last unit have different coding in this first character. Following the code character is a field of 2 characters which is a pointer to the next store unit concerned with this particular packet or queue. The last two characters in the store unit are reserved for sum-checks to check the correct operation of the links.

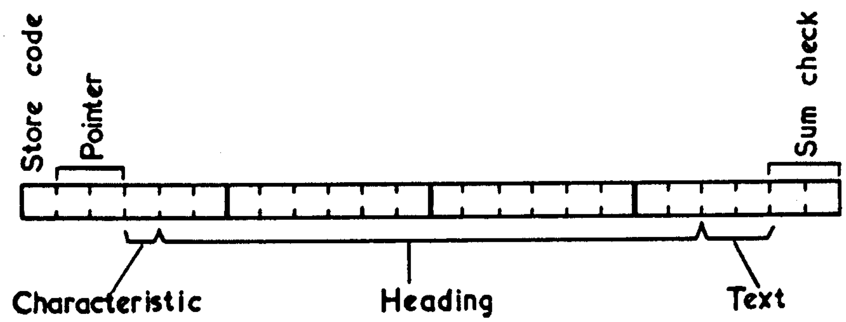


FIGURE 4. FORMAT FOR A STORED PACKET OF MINIMUM-SIZE

This kind of format allows the text to be code-free, in other words it need not conform to a character code and no particular values of character are reserved for indicating the beginning and end of text.

Store units are strung together to form packets and packets strung together to form queues by means of an elementary kind of list structure as shown in Fig. 5. Store units of four consecutive words each are represented as rectangles. They are strung together to form packets by the pointers pointing from one unit to the first word of the next.

The last store unit of the top packet contains a pointer indicating the address of the first word in the next packet, and so forth, down to the bottom of the queue where in the last store unit the value of the pointer is irrelevant.

The store units may be located anywhere though they are shown stacked in the figure.

Fig. 5 illustrates the two kinds of queues that will be present in the node computer, input queues and output queues. The input queue is the simplest, and it has a top and bottom packet the positions of which are indicated by pointers shown on the diagram. These pointers, which may either be in special hardware registers or in the main store, are the addresses of the first word of the store units concerned. New packets coming in from a communication link are added to the bottom of the input queue and packets are taken from its top to the bottom of the appropriate output queue. The output queue also has a top and bottom element with pointers, and packets are taken from the top of the output queue for sending out on the associated link. When they are sent, however, they are not removed but are allowed to stay in the store to form the upper part of the output queue called the "trace". The top of the trace also has a pointer. The purpose of the trace is to provide a backup if errors have been made in the transmission of data through a link. The time delay through links is such that a dozen or more packets might have been sent before the existence of the failure in check digits at the receiving end of the link can be signalled back to the sender. For safety, more than this number of packets must be held in the trace.

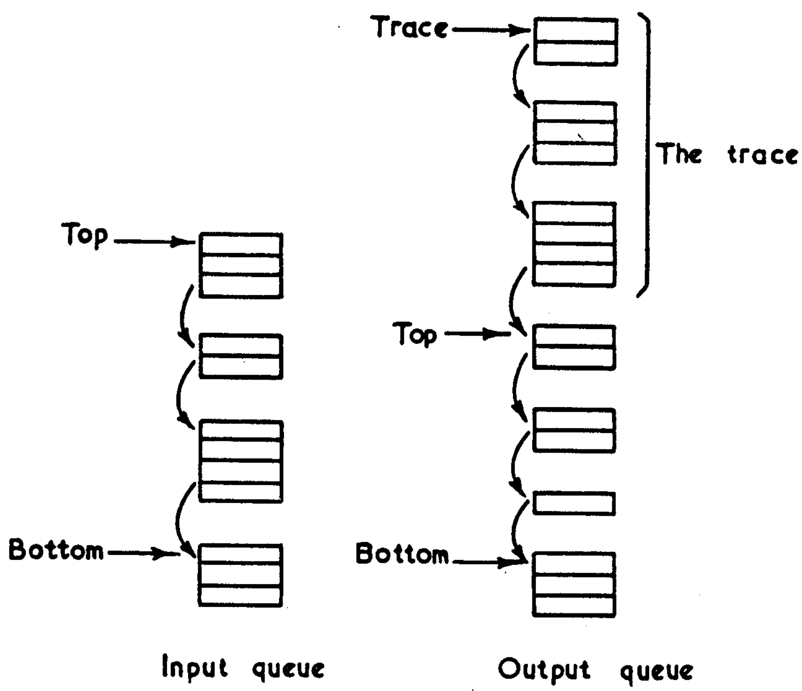


FIGURE 5. TWO VARIETIES OF QUEUE IN A NODE COMPUTER

Spare or unused store units are kept in a free list. From time to time portions of this list are detached and added to the bottom of input queues to provide space for incoming packets. This need not be done for each incoming packet because space for several can be provided at a time. When packets are sent out on a link, the value of the pointer need not be transmitted, but instead, these characters can be used to indicate the location in the store of the sending node of the first word of the packet i.e. the pointer value of the previous packet. From time to time the receiving node will indicate that a certain number of packets have been received correctly by sending back the "pointer" value of the last one. This enables the trace pointer to be shifted down at the sending end and storage units restored to the free list. The use of a free list allows flexibility in the allocation of space between the various queues.

The shifting of a packet (or consecutive set of packets) from an input to an output queue involves changing the values of the top pointer of the input queue, the bottom pointer of the output queue and the pointer contained in the last item of the output queue. This can easily be shown to involve 6 store accesses if all pointers are in the main store. If, however, the pointers to the tops and bottoms of the queues are held in hardware registers, only two main-store accesses are needed. Since the input or output of a full size packet could involve 20 main-store accesses, the economy in accesses obtained from the use of hardware registers may not be considered worthwhile.

The process of manipulating these lists would be very time consuming if all the operations had to be carried out with an unsuitable order code. However, the operations of shifting items between the free list and the queues and from one queue to the other are all very similar and it is probably that in a micro-programmed machine, special operations could be provided which would carry out these processes with a minimum number of store cycles. Under these circumstances it is not unreasonable to assume that about 50 store accesses would be sufficient to deal with the movement of one packet through the system. In a modern computer this would occupy perhaps 50 microseconds. Compared with the time of 680 microseconds required to transmit one packet over a link, this indicates that a well designed computer could handle about 13 links. In fact, the node would never be worked close to saturation on any link, certainly not on all links, and it seems that a node handling say 10 links would have half its capacity available for dealing with routing and with exceptional conditions such as errors in the received packets, or congestion.

Suppose, however, that the amount of traffic to be carried justified the provision of links with considerably greater bit rate than we have so far assumed. In this case, one simple solution would be to provide several links with the standard bit rate and have complexes of nodes at the switching centres. This would enable nodes of standard design to be used, and ease the manufacturing problem. But it may prove more economic to design nodes capable of a higher throughput of packets, and this can be done by multiplexing stores. All the link hardware would then have to be capable of accessing any of the stores, which requires a full availability switch between links and stores. Only detailed design will indicate which is the best arrangement, the high performance node or a multiplicity of nodes. The multiplicity of nodes lends itself to simple “fail soft” provisions.

**Function of the Interface Units**

Fig. 6 shows diagrammatically the function of an interface unit in relation to three kinds of users: computers, displays, and keyboard/printers. It does not mean that the units will actually be arranged in this form, for example, the computer in the interface unit *I* might have spare capacity available to handle displays. The relative location of the units is also not meant to be indicated by the figure, which is purely concerned with their functions.

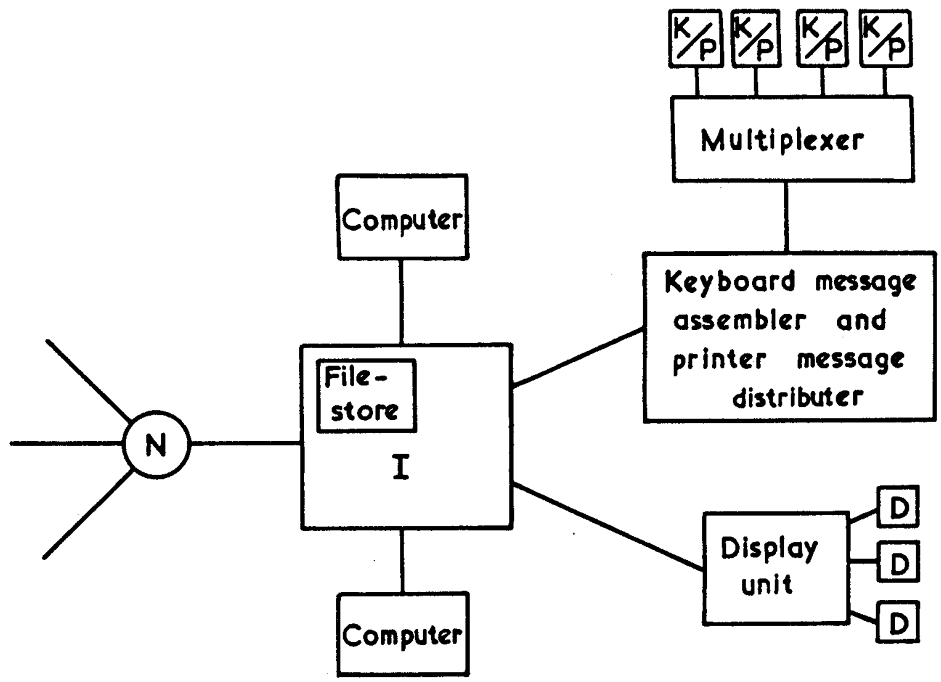


FIGURE 6. AN INTERFACE UNIT SHOWN IN BLOCK DIAGRAM FORM

The simplest of the interfaces which the system has to provide are those with computers. From a computer it can be expected that complete messages with the correct destination number in a fixed format and with sum checks will be provided. The main functions of the interface unit will be:

(a) Adding the route and source information. The route will be obtained from a table in the store of the interface unit *I*. Many users will have a special route information which must be referred to from a table.

(b) Dividing the message into packets if necessary.

(c) Adding the necessary characteristic, pointer and sumcheck according to a format such as the one given in Fig. 4 and placing the packet in a queue for sending to the attached node.

(d) Accounting, and collecting operational statistics.

(e) Dealing with the limitation of input necessary for certain kinds of congestion.

(f) Updating the routing information when necessary. For this purpose the interface unit must have a complete model of the network in the form of a data structure held in its store so that the consequences of system breakdowns or congestion can be worked out easily and reflected in the routing table.

(g) Taking incoming packets in turn from an input queue and storing and assembling those which are shown as being only parts of messages.

(h) Despatching the text and source information from incoming packets at an appropriate speed to the attached computer.

(i) Interpreting commands, and maintaining the status information for each user.

There will no doubt be other operational features that will be discovered by the practical operation of a system but it is expected that the ones listed above will form the main load. This can be seen to be comparable with or at most greater by a small factor than the computing load required to handle the same number of packets in a node computer. To deal with the routing table and the possibilities of storing messages for some time at the request of users, the interface unit will be provided with the moderately large store, possibly a hierarchy involving discs at the large capacity end. It is possible to confine the provision of large stores entirely to this type of unit in the network.

The way in which displays are accommodated is liable to change as the technology develops but at the present stage, displays must be driven from moderate-sized stores which are not situated too remotely. Such a store, for economy, drives several displays as shown in Fig. 6. A display store would contain for each display it drives a list structure not unlike the queues described for the node computer. Therefore they create no new problems for the communications system, apart from their storage cost in their present form, which it must be the goal of further development to improve.

The most difficult interface to handle is that of the keyboard/printer. Fig. 6 shows 2 units interposed between the keyboard/printers and the interface unit. The density of the keyboard/printers even in a commercial district may be low, and a multiplexer is shown in the diagram to reduce the cost of connecting them to the nearest assembler/distributer unit. In fact, two levels of multiplexer might be provided for further economy in the cost of local lines. The assembler/distributer unit has the job of demultiplexing to recover the original coded characters, assembling these characters into packets for sending to the interface unit and the reverse of these two processes.

Therefore, for these peripherals the system table of functions (a) to (i) must have two more functions added.

(j) Assembly of characters from the keyboards to form complete messages, recognising the terminating characters. Despatch of these messages to the unit *I*.

(k) Distribution of received messages at the appropriate slow rate, signalling back when they have been completely sent.

The convention used with keyboards to determine when a complete message is ready for sending is normally to regard "newline" as signalling the end of a message. But the convention really should be adapted to the user's needs. In order to keep the design of the assembler/distributer simple, not too many choices should be offered. One—an extreme case not frequently employed—would be to send each character separately. Another is to regard spaces, or alternatively characters other than letters and figures, as signalling the end of a message. The choice, in each application, is determined by where in the stream of input the computer is able to take action that would help the user. If no other convention than "one line-one message" was employed, the system would be no worse than most existing multi-access systems.

Consider an outline system design for the use of teleprinters as terminal machines. These machines have the advantage of cheapness but present a difficult problem to the assembler/distributer.

The multiplexer could sample each of the attached machines at five millisecond intervals. Twenty-five such machines could be connected to a multiplexer to produce a pulse train at 5000 bits per second which could easily be carried by a cheap cable to a higher level multiplexer. At this point 20 such channels could be multiplexed giving a group of 500 channels and a data rate of 100 kilobits per second.

The interpretation of the incoming multiplexed teleprinter signal could be achieved by logical operations on the input pulse train in combination with a dozen or so bits of information for each teleprinter giving the current "state". Such an arrangement is familiar in the theoretical construct called a “finite state automaton”, and it has been exploited in experimental electronic telephone exchanges. The logical steps would have to be carried out at 10 microsecond intervals, referring to a different state word at each occasion. A small core store could hold the state words and some built-in logic employed to achieve the finite state machine. At relatively infrequent intervals, complete characters would be formed, and a piece of programme would be initiated to place the assembled character into the appropriate place in a packet assembly area.

A small computer, provided with necessary fixed logic, could probably carry out this work for two or three groups of 500 lines each, leaving time for the despatch of assembled packets. More detailed design is needed to prove the feasibility of this scheme.

In the output direction, a similar mechanism could be used. An addition feature is needed for output. When a packet has been completely transmitted to the printer, a message containing the pointer must be returned to the interface unit, to request the next packet in the message. At the end of the whole message, the interface unit must send a short message in standard form back through the network to the source to indicate that printing had been completed. Such a feedback is essential if computers are to handle long printouts economically at these slow rates.

The complexity of an assembler/distributer associated with 500 to 1500 users would be similar to that of a node computer. The data rate would be so low that as many as 100 of these units could be accommodated on one interface unit if the geography allowed.

The outline design for an assembler/distributer described above was based on the standard teleprinter. If advances in technology, such as the “metal-insulator-semiconductor-transistor”, make it possible to produce cheaply, logical units of some complexity then completely different schemes become possible. For example, a number of user’s instruments can be joined to a common highway and be sampled by short pulse trains carrying the terminal address which would be followed by output data and, after a short interval, replied to by the peripheral. The depression of a key would cause the coded character to be sent only once, at the next sample time. The task given to the assembler/distributer would thus be greatly simplified but there is a danger in any highway scheme of putting a whole group of users out of action due to a fault in one machine.

**Line Concentration**

Telephone terminals have two states, in the rarer of which, the “off-hook” condition, they occupy much more equipment. The technical means of exploiting the competitive rareness of actual use is the line-concentrator which only connects the active lines.

It must *not* be assumed that these considerations carry over unchanged to a digital network, and that line-concentrators will necessarily be needed.

The outline system design described above employed multiplexers rather than line-concentrators to economise in the cost of local lines. This would enable it to deal effectively with users whose terminal is permanently “on”. The only additional cost incurred in this arrangement is possibly a larger store in the assembler/distributer to hold a few bits for each user and up to about 1000 bits for the users who delay the completion of a message. At present-day costs of core stores this capital cost is small.

It seems unnecessary to encourage users to hurry and get off the line when the computing cost, which is another factor they must consider, is not dependent on “thinking time”.

The only way to achieve a substantial saving is to share a peripheral with others, thereby sharing the fixed costs.

The place of line-concentration in the network is therefore not decided, and it may have no place.

**Error Control**

The network has error control features built in, and some of these have been described. For the most part, they will not be apparent to the user, and their purpose is to improve the accuracy of the network.

It has been assumed that each link within the system has error detection by sumchecks. In the event of an error being detected the system is capable of retransmitting the offending packet. This principle will almost certainly be extendable to the links between external computers and the interface units.

The user’s error control requirements will differ according to the application, and it is important that the networks’ properties should allow the users to apply the error control schemes they want. In many cases in which a computer is accessed from a keyboard, the returning information acts as a check of the correctness of what was sent. In this case specific measures for error control are unnecessary. In other applications, the error correction properties of the human user will make overall error control unnecessary.

There are applications in which nothing short of a repeat back of what was sent will satisfy the user. Repeat back from the far terminal is a possibility requiring no special provisions but the delay through the system is likely to slow down the operator. Repeat back of each character requires sending each character as a separate message, which would be expensive, but not out of the question. An alternative is to repeat back from the local assembler/distributer. The message can then be checked by the sender before the final character is added which causes it to be sent off. This facility would require a special feature to be included in the assembler/distributer design, so it would be important to discover the extent of the demand for it.

The users should be made responsible for all forms of overall error control but the facilities offered by the network should be such that the user's error control is easy to provide.

**Redundancy features, Dependability**

It must be admitted that the redundancy features needed to ensure a service in the presence of faults have not been carefully thought out.

A network with many similarities to the present one was described by Paul Baran in "Distributed Communications". Rand Corporation Memorandum RM-3420-PR August 1964. A large part of that report was concerned with the behaviour of the network when many nodes and links were out of action (in a military context, this is important). The highly-connected networks there considered are not needed in a civil environment, but nevertheless the value of overconnection as a fail-soft feature was well demonstrated.

If we suppose that paths for 1.5 Megabit per second signals, because of their telephone use, will become widely available, rearrangement of the main units of several neighbouring interface systems become possible. For this purpose, it would be useful if all the signals involved (such as those from the multiplexer to the assembler/distributer) could be carried, in an emergency, to a different place for processing.

It seems likely that redundancy at the level of major units can best be employed to improve the dependability of the system. Users requiring the highest dependability will have to make more than one connection into the network, at different interface units.

The question has not been carefully looked at, but it does not seem necessary to assume more than 30% of extra equipment will be needed.

## The Command Language

In many multi-access systems, a distinction is made between “instructions” and “commands”. Instructions form part of a program. They are not obeyed until the program is entered and they are reached, then they are interpreted by the control unit of the computer. Commands are directed to the operating system which manipulates the computer’s resources, namely the system programs (compilers etc.) the peripheral machines and file stores. Commands are usually included in input streams but may sometimes be included in programs to request output/input operations, for example.

The form of a command can be a function word followed by a number of parameters. For example, using an obvious mathematical notation, a command called “output” could appear as

output (a,b)

The delimiters, in this case parentheses and the comma, could be replaced by any other symbols. As an example for a communication network, the simplest of all requests would be to send the text “abc” to the destination “123”. This could be expressed as:

* send to 123/abc

Here the delimiters are the asterisk and solidus and the function word is “send to”. The lack of a delimiter after the function word need not cause ambiguity but it might be decided for greater clarity the function word should be distinguished by a form of underline caused by a non-escaping character.

This method of handling commands is appropriate only to a character-coded source of messages. Messages directly from computers would probably be handled in a different way, for example by a fixed format in which the first field was the destination number. In this case the text could be […a line of text is missing from the bottom of original document page 21].

Many human users of the system will require to be connected to the same destination either all the time or for a large part of it. Facilities should be provided so that they can become unaware of the communications system in the sense that their peripheral equipment seems permanently connected to the service they are using. For this purpose there should be a command which selects a certain destination and retains it for all future messages until a different kind of command is given.

Another command would indicate to the network that the terminal concerned was, until further notice, interested in receiving messages only from one, named, source. This would be a reasonable precaution when starting a conversational interaction with a computer. Commands would also be available for requesting that incoming messages be stored for a while and for having these stored messages printed out.

The range of commands which is needed will only be discovered in practical operation. None of the ones mentioned seem to offer any great difficulty in their provision.

Each user employing these facilities will need a small table stored in the interface unit and holding the data relating to the command at present in force. The disc store can generally be used for these tables, which, are referred to when a packet is sent out or received.

**Control of Congestion**

In so far as the traffic behaves predictably, with known statistics, it is hoped that the capacity of the system can be kept well above saturation and that congestion will not occur. Terminals capable of high activity such as computers, may have to be restricted in their output rate when conditions are critical.

A more difficult condition to deal with is congestion due to over-use of one particular computer service. In some cases, this could be handled by the computer service requiring users to “log in”. One destination number could be used for attempting to “log in”, and another for using the service, to avoid the one blocking the other. The destination number of the *service* could be “ex-directory” and would be sent back when a log-in was successful, but only to the interface unit of the caller, where it would reside until he logged out.

Another case is that of a computer offering a very simple and rapid service which became overloaded because of some public event. In this case there would be no desire to restrict access to a category of logged in customers, but beyond a certain point its traffic would have to be reduced in order to avoid saturating nearby links. One technique would be to choose a ring of nodes surrounding the trouble point and turn back a proportion of the messages in question. Messages refused in this way could not be stored locally because nodes have no excess storage capacity.

**Cost Estimate**

At the present state of knowledge, to estimate cost is most hazardous, but it is worthwhile to make the best possible guess.

In the rough description of the system given above the conclusion was arrived at that the node computer, the part *I* of the interface unit (see Fig. 6) and the assembler/distributer for 500 lines were all of about the same degree of complexity. There would be computers with perhaps 8K of one microsecond store, and an instruction repertoire of average size. They each contain some special equipment as well. Looking some years ahead to the installation of such a system a figure of £50,000 for one of these machines seems not unreasonable. The interface unit also contains a file store so we put the price of this unit as £100,000.

For each 500 users we assume that there will be provided one assembler/distributer, one quarter of a node computer and one quarter of an interface unit. This gives the total cost of £175 per keyboard user. To this must be added the cost of the local line and part of the multiplexer system, say £50. The cost of the trunk lines when high rate PCM is common will be very small and by comparison with the telephone system we will add £35 per keyboard user for buildings. The capital cost per keyboard user totals £260. The terminal equipment, assuming a competitively priced teleprinter, will be £400, the largest item of cost. The capital cost per terminal is from 3 to 4 times that of the telephone system, and this might be paid for by a rental charge.

Human users of the system would not generate enough traffic to load it fully. It is difficult to predict how large computer generated traffic could become, but the charge made for the communication service would have to take into account the cost of extra traffic. It could do this by a fixed charge per packet carried, taking no account of distance.

We suppose that the main cost in adding new carrying capacity to the system is in providing node computers. For a node computer capable of carrying 15,000 packets per hour, we might allow for short term and long term variations in traffic by assuming only 50,000 packets pass through per day. The cost of the node computer, reckoned at £10,000 per year gives a cost per packet per node of 1/8th of a penny. The charge per packet would therefore be in the neighbourhood of one penny. This amount would be of no significance to the keyboard operator, but it could give rise to significant cost in a rapidly changing display, for example.

Costs calculated in this way cannot be taken too literally. In particular they depend on a full complement of subscribers, which would not exist in the early evolution of a network.

These rough considerations show that the critical areas for attention to cost in the design are the interfaces with the slow peripherals, and the slow peripherals themselves.

This conclusion is reinforced by the trend in the cost of computers. The cost of high speed computing, measured as cost per computation has reduced by a factor of 2 every 3 years for some time. Developments in stores and in interconnection techniques give promise of a further large factor in cheapening computation before computer technology stabilises. These changes will emphasise even more the cost of the peripheral machines and their connection into the network where the utilisation of fast electronic devices is poor.

**Problems in the Implementation of the New Network**

The demand for a digital communication service of the kind described must be demonstrated before large scale implementation can begin. But the demand will not be expressed in the right terms until the technical possibilities are known. There must therefore be an iterative process of studying the stated demands, designing a system to meet them, publicising the technical possibilities and in this way stimulating the expression of communication demands in a way which takes account of the new techniques.

Communication networks are resistant to technical advances because new equipment must be able to interwork with all the existing equipment. Therefore the assumptions made in the first designs may determine the form of the network for a long time. This will tend to delay the start on fullscale implementation. But the network, as described, can be developed in many directions without difficulty. The one standard that affects all the network at once is the packet format because the packets should, for economy, travel about unchanged anywhere in the network. In particular, the contents of the heading and the method of organising routing are fundamental decisions. At some point, international standards will be needed.

The need to gain experience can be met by a trial with a small network which does not set standards but can be used for experiment and later replaced.

The network as described, with its high minimum data rate between nodes, fits well into a national communication scheme where speech is largely carried digitally. The bit rate for speech is high enough that channels of 1.5 Megabits per second do not represent a large requirement by comparison. The network is therefore well suited for a telephone transmission situation that does not yet exist. Fortunately, the digital network can give moderate service with a much lower minimum data rate, which hopefully will ease the problem of its provision using wide-band F.D.M. telephone circuits. A useful start could be made with 100 Kilobit links, for example.

It is important, however, that a goal is set early on, towards which the network will evolve, and which will, in turn, influence the development of P.C.M. telephony. A technical and an economic plan over a period of the order of 5 years will be needed in order to justify the investment, and in its early form the new network may be disappointing in performance and cost. It is probable that, because computer technology is rapidly evolving, the planned life of equipment may have to be rather shorter than has been the case for telephone exchanges.

**The Future of the Existing Networks**

The new network could obviously be used for telegraphy. It could offer a wider range of facilities than the Telex network and, according to the rough estimate, cost about the same amount of the user. The facilities accessible by the digital network will be both publicly and privately owned. As the clerical systems of large businesses become computerised, much of the communication which it is the aim of the Telex service to capture would become computer-computer or human-computer in nature. Therefore it can be predicted that the Telex network will be absorbed into the digital network when the latter is large enough.

The telephone network is a different matter because of the high bit rate and the synchronous nature of its transmission requirements. Given enough local storage to smooth out the variation in response time for messages, it is technically feasible to carry voices over the proposed network. Computer developments in the distant future might result in one type of network being able to carry speech and digital messages efficiently. This will necessitate a network of the store and forward kind because line-switching is fundamentally more limited. This hypothetical development towards a single network is sufficiently remote and uncertain not to delay the development of the new network.

The picture that emerges is of two communication networks, one digital, asynchronous, and using the store and forward principle; the other using line-switching, synchronous in nature, and designed for carrying speech but able to carry bulk digital information at constant rates. The same digital links would serve both networks.

**Summary**

Multi-access scientific computer systems and real-time business systems are the kinds of computer applications that will greatly increase in quantity and scope. They will give rise to new sorts of communication requirements.

A network design is proposed with the following main properties:

(a) It connects different kinds of users, for example computers offering real-time services and keyboard/printers for human use, in a way which allows them all to intercommunicate usefully.

(b) It handles data in small packets which are stored at each switching point or “node” and forwarded after they have been completely received and checked. The delay through the network can be small, say 100 milliseconds.

(c) The network design, its economics and the method of charging for the communication service does not penalise the user’s terminal which is switched on i.e. connected, but idle.

(d) The network carries out part of the function at present given to each multi-access computer. It assembles incoming messages from the keyboards and distributes outgoing messages at a suitable rate to slow printers, so that the main network handles only messages or packets (parts of messages).

Enough detail of the design of the system is given in the report to make the feasibility and economy reasonably assured, when the necessary digital transmission paths become available widely.

It is proposed that the network design should be made for a future time when P.C.M. telephony is the general rule. A version with rather less performance can build up in the meantime, and if a few, key, design decisions have been correctly made there will be no difficulty in improving the network performance as the necessary digital paths become available, until it reaches the planned goal.

For a very long time it seems that a store and forward digital network must coexist with the switched telephone network. The latter will still, perhaps, carry bulk digital data that has no great urgency. Ultimately, if high speed digital equipment, including stores, continues to cheapen, it

will be possible to combine both networks in a single digital, store and forward system, but this is a long way ahead and should not affect the proposal made here.

The views given here are those of the author and are not necessarily endorsed by the National Physical Laboratory or the Ministry of Technology.

D. W. Davies
June, 1966.

**Erratum**

The cost calculation in para 5 of page 29 must only be regarded as 'order of magnitude' but nevertheless needs correction because there were errors in the calculation. Our present estimates are the following:

| | |
|---|---|
| Node computer cost/ annum, assumed, | £10,000 |
| Peak packet handling capacity, | 5000/sec |
| Maximum capacity allowed by queue length considerations, | 2500/sec |
| Factor allowed for traffic imbalance at a busy hour, | 5 |
| Factor allowed for mean/busy hour ratio, | 10 |
| Cost per packet per node resulting from the above data, | $1.5 \times 10^{-3}$ pence |
| Cost per packet, assuming the use of 6 nodes, | $10^{-2}$ pence |

This calculation covers only the use of the node computers, and was made to deal with the assumption often expressed that 'message switching is expensive'. The cost of the interface computer and local distribution system is, of course, more significant for a terminal matched to human speeds, and we have no cost estimate readily available for this.