ACM

Symposium on Operating System Principles

October 1–4, 1967 Gatlinburg, Tennessee

A DIGITAL COMMUNICATION NETWORK FOR COMPUTERS GIVING RAPID RESPONSE AT REMOTE TERMINALS

D.W. Davies
K.A. Bartlett
R.A. Scantlebury
P.T. Wilkinson
National Physical Laboratory
Teddington, Middlesex
England

Association for Computing Machinery 211 East 43 Street New York, N.Y. 10017

A DIGITAL COMMUNICATION NETWORK FOR COMPUTERS GIVING RAPID RESPONSE AT REMOTE TERMINALS

D. W. Davies
K.A. Bartlett
R.A. Scantlebury
P.T. Wilkinson

(National Physical Laboratory, Teddington, Middlesex)

Introduction

Those computer applications which involve rapid response to events at distant points create special problems in digital communication. Such applications are increasing in number, and could increase more rapidly if better solutions existed to the communication problems.

The present-day methods for communication of data in rapid-response systems employ ‘private wires’ for the transmission paths or, where the available data rate and reliability is sufficient, employ voice channels from the switched telephone network. Given these rather arbitrary transmission facilities the user adds the terminal equipment necessary to make a communication system and sometimes integrates a number of paths into a private network.

It is clear that a common carrier network could give great economies, since many of the communication functions would, in such a system be handled by common equipment. The difficulty in such an approach is to develop a system design general enough to deal with the wide range of users’ requirements, including those that may arise in the future. This paper proposes a design for a common-carrier data network. Earlier work has called attention to the problem, but not proposed a solution (1).

A proposal such as we are making has many aspects, technical, economic, organizational and political. This paper is chiefly concerned with the technical aspects of the network and the range of communication requirements which it is designed to meet. To put the proposal into context some introductory remarks are needed.

The economics of the network have been considered, in so far as it is possible to do so. The results were very encouraging, though subject to some uncertainty. A network which covers the main centres of population and is moderately well loaded with traffic would seem to be economically viable. On the other hand, starting a network would encounter the difficulty of building up traffic to an economic level. An experimental network serving a well defined community would be a necessary first step.

The network proposed has similarities to many existing systems. Thus the SITA high-level network for air-line seat bookings and the proposed CCITT No. 6 signalling system show some resemblance while there is a superficial resemblance to telegraph message switching systems. The reader acquainted with these systems should avoid being misled by the points of similarity because this proposal differs significantly from each of those mentioned.

The most relevant previous work has been by Paul Baran and is reported in a paper "On Distributed Communications" (2). Our proposal is like Baran's in its high-level network but the similarity does not extend to the manner of using the network. Baran's proposal was intended for a military communication system in the future, and carried secure speech.

This paper is concerned with the system design of the network and, in this context, transmission paths for digital data are components of the system but their design is not part of the subject matter of this paper. Digital methods of transmitting telephone traffic are expected to increase in importance and this will make available cheaper and more reliable digital transmission. Telephone economics have determined and will in the future largely determine the rate and manner of penetration of digital transmission techniques into the communication network. 24-channel P.C.M. systems with a bit rate of 1.5 Mbit/sec are in widespread use for short-haul purposes, and an experimental 224 Mbit/sec coaxial cable long-haul system has been reported (3) (4) (5) (6) (7). There is little doubt that microwave systems and satellite relay using digital transmission will carry both data and P.C.M. speech in the future. The lack of operating long-haul digital paths presents a difficulty in the next few years. More expensive paths over frequency division multiplexed facilities or the adaption of older cables for digital transmission will fill the gap.

The present proposal does not depend essentially on the installation of digital transmission channels, but these will be necessary for its full development. Our ideas have developed around the use of 1.5 Mbit/sec as minimum capacity in the high level part of the network. As will be seen later, a minimum data rate in the high-level network will be established mainly for reasons of response time.

It would, of course, be unreasonable on economic grounds to propose a network which was entirely separate from the existing telephone system. The system proposed can share the digital communication links with the telephone system. Its requirements for transmission capacity may well remain much smaller than those of the telephone system, and it should never be short of capacity in a fully-developed network.

We would like to assert that the local distribution part of the proposed network could share its lines with telephones. Unfortunately the use of the bandwidth of local lines for telephony has developed in a haphazard way. The full economy of the proposed network may not be realized until a new type of local distribution network suitable for data and telephony comes into operation. In the meantime, special wires for local ends might have to be provided wherever a high bit rate is needed.

Outline of the Proposal

We have not found it possible to present the design of the network in an entirely logical manner, starting from well-defined users' requirements, because the set of users for which it caters is not determined in advance. There is a wide variety of potential users of digital communication, and we must adapt our design to deal efficiently with the largest range of requirements.

The scheme we have adopted in this paper is to give a brief outline of the design, then describe, by examples, how it is used and how it appears to the user. Finally we describe the operation of the network in more detail.

The network carries short messages in the "store-and-forward" manner. These messages are handled by a *high level network* consisting of *nodes* connected by digital links. Each message enters the high level network in a well-defined format which includes a note of its source and destination. The responsibility for putting messages into this format belongs to the network, not the user, and this is a vital principle of the design. Between the high level network and the users there are *interface computers* each handling a mixed collection of subscribers within a geographical region. Figure 1 illustrates the form of the network. To the user, the store-and-forward nature of the network might in some instances be hidden. It will become obvious, when the design of the node is described, that storing each message before retransmission simplifies the design and makes it adaptable to technical improvements.

The short messages in fixed format which are carried by the high level network are called *packets*. They are directed through the network by an adaptive routing technique. For each destination, the output links are listed in a table in the node computer in order of preference. The highest preference link will be taken, subject to certain links being unavailable due to faults or congestion. The route tables can be amended by the network administration to improve performance after a failure has occurred. This is a form of the 'hot potato' routing advocated by Paul Baran in Ref. (1) and is appropriate when failures are not frequent.

There is no standard bit-rate, and no multiplexing in the sense of pre-arranged timeslots. Each link between nodes has a prearranged bit-rate, but these rates can differ and be changed without trouble as the technology advances. The requirements of P.C.M. telephony will lead to multiplexing of groups of telephone channels, in the range of speeds from 1.5 Mbit/sec upwards. Below this speed, the multiplexing of data channels would seem to be clumsy and unadaptable. It has no place in our network proposal.

Subscribers of the network are of many kinds, and subscribers in conversation will in general differ in their interfaces with the network having, for example, different bit-rates. The use they make of the high-level network with its capacity for handling short messages is organised by their respective interface computers.

Because the network employs ‘store-and-forward’ methods there is a delay to the passage of data in addition to the unavoidable transmission delay. The time elapsing between the receipt by the network of the last character of a packet and the beginning of output at the far end is called the *response time* of the network. Our aim was a response time less than 100 milliseconds. Calculations described later indicate that this can be achieved with high probability.

Some consideration must be given to the way in which the network can be prevented from a major breakdown in the event of equipment failure. In the high level network the most powerful method is to arrange that the network is over-connected and use alternate routing to maintain service. A computer failure in a node would probably destroy the few packets that are stored there. The question then arises whether duplication of the computer at each node is needed to guard against this possibility. It is thought that all users of the network will provide themselves with some kind of error control and that without difficulty this could be made to show up a missing packet. Because of this, loss of packets, if it is sufficiently rare, can be tolerated. Corruption of a packet which escapes the internal error control would be much more serious.

At the interface computers, fail-soft provision is more difficult. One might use, for example, a pair of computers which share the work both of an interface computer and the adjacent node, either of which can keep a service going if the other breaks down.

The reconfiguration of the network to maintain service after a computer failure is made easier if standard interfaces are employed for many purposes. The “multiplexed” or concentrated lines from subscribers, to be described later, might then be extended by standard links to the nearest functioning computer which is able to carry out the interface function.

The study of fail-soft provision has not proceeded beyond these general principles.

The Users of the Network

The digital communication network will be attached to a wide variety of subscribers' equipment at its terminals having different characteristics as sources and acceptors of data. Keyboard consoles, enquiry stations, graphic display consoles, bank proof-machines, line printers, paper tape readers, file storage systems, multi-access computers, remote actuators and sensors for transport systems and pipelines, meteorological and hydrological instruments, and a steadily increasing variety of new equipments will be attached to it.

The design of the network would allow any of these equipments to send information to any other but in practice many corporate users of the network will have designed their various equipment to operate mainly as a sub-network. On the other hand, there will be standard terminals for human use such as consoles or enquiry stations that can make use of a wide variety of computer-based services provided commercially through the network.

Use of the network can be roughly divided into three categories, man-computer, computer-computer and computer-machine.

Man-computer interaction has been the subject of much research and development, but the use of this kind of interaction in industry and commerce is merely beginning. An important use of such interaction is the validation of input data, and at least one system in operation today exists for this purpose alone. The most common raison d'etr of an interactive system with many remote terminals is the sharing of a collection of data. Access to remote computing power, on the other hand, may become a less urgent requirement.

Among the computer-computer applications the development of business systems involving direct transfers of information, analogous to human correspondence by letter, can be expected to grow and to increase the pace of much business activity.

Services can be offered by one computer to another, often with very favourable economics. For example, small local computers can call on a remote service for backing storage, for the compilation of programs, for photo-typesetting or elaborate graphic work—any service which is cheaper if it can be purchased in bulk and retailed.

The application of the network to computer-machine systems such as the control of transport, electrical distribution and gas distribution is more uncertain because of physical difficulties with the local connections and the fact that special networks may have been installed before a common carrier network can come into being.

A natural employment of the proposed network is the handling of signalling information for the telephone switching network. Use of an independent signalling system would be a great advance in the technical developability of the telephone system.

Control procedures involving the users

The usefulness of the high level network depends on how well the interface computers are able to offer the facilities needed by users at their terminals. The terminals can vary greatly in their requirements, and we can only give illustrations of the kind of facilities that are intended. The development and standardisation of the best modes of operation will take time and experience. It is merely necessary here to show that the overall scheme allows sensible solutions to be developed.

We shall use for our main example a paper tape reader sending data to a computer. Assume that the tape may contain a few hundred characters, but the lengths of tapes are such that the computer can be expected to find storage space for the whole block of data.

Figure 2A shows in outline the path between tape reader and computer. The queues shown in the figure serve all those input/output processes of the computer which involve the network. In the computer, a program is ready to take in and store the data, but is held up for lack of data. It can be re-activated by the appearance of a relevant packet in the service position of the input queue.

The figure shows that complete packets are produced and received by the tape reader program and that these packets travel as far as interface computer Y but, in fact, changes of format occur in interface computer X. The communication between Y and the tape reader is in single characters only.

Figure 2B shows a possible control procedure for the reading of a tape. Only the error-free sequence of operation is shown. The procedure begins and ends with *status characters* and *status packets* and the conversion between character and packet is carried out by interface computer Y. These status characters correspond to the control characters concerned with supervising the transfer of data and the recovery from errors in the type of control procedure which is well-known, but they also correspond to the ‘status’ information which passes between a computer and the controlling logic for a peripheral.

The unit carried over the local network to and from the tape reader is a byte of information with some extra bits which, among other things, determine whether it is *data* or *status* information. Note that the data byte can be code-free without the use of special conventions or hardware, since the presence of status information is indicated separately. The value of this arrangement will be seen in what follows.

Status information in the packet form is distinguished by the value of a characteristic which is in the first character position of the message area. For the purpose of the control procedure considered here, the characteristic needs three possible values indicating, respectively:

status packet

data packet, with more packets to follow

last data packet

The input process of the computer recognises status packets and may store them in a special queue, as shown, if priority is found to be necessary. The interface computers may also have to examine the characteristic, so it is in a fixed position in the packet format. The significance of values of the characteristic should be fixed for the network as a whole, but local conventions are possible for the values of status characters or the content of status packets.

With a multi-packet block of data, such as that delivered by the tape reader, there may be a danger that packets will arrive out of order at the far end. With paper tape readers this is only a remote possibility below a reading speed of about 1000 characters per second. The control procedure can, of course, include serial numbers for the packets in a fixed position of the message area. These would be inserted by interface computer Y and checked by the interface computer X.

The development of new control procedures for different and more complex applications of the network will require experience that we do not have at this time. The information carried in status packets can be increased, and if there is a common requirement for a new kind of information associated with control procedures, a new value of the characteristic can be assigned. Therefore, there is little impediment to devising new control procedures, except the need to avoid an unnecessary load on the interface computer.

Before leaving the tape reader example, the way in which destination addresses are handled should be mentioned. Assuming that the tape reader must send data always to a designated remote computer, the destination data will be added by interface computer Y, from information in a table in store. If the user is to change the destination, he could do so by a message on tape, in an agreed format and code, but to distinguish this from the (code-free) data normally transmitted from the tape station a further value of the status character must precede this message. For many users, the ability to ‘lock’ a terminal to a distant computer will be a valuable safeguard.

At the computer end, it is probably convenient for each process delivering packets to the output queue to insert the destination in the correct place in the standard format.

For a second example we consider a remote connection to a keyboard with character printer (such as a teletype). Again, the local network communicates with the terminal in single bytes. (The two examples we have chosen both have this feature, but it is not universal). We can use figure 2A as an illustration, with the keyboard and character printer in place of the tape reader.

The new feature of this terminal is that a line of type is a natural unit to send as a packet. Multi-access systems commonly respond only to completed lines. We shall not describe the control procedure fully, but deal only with the method of delimiting the packets.

For input, characters are stored in interface computer Y until the return or newline key is depressed. If the printer allows a long line, greater than one packet, there is no problem in dispatching a packet whenever the store in Y contains enough characters, and following this with the remainder when the line ends.

The delimiter symbols for various users may not always be the same characters, and the task of interface computer Y in checking each input character against the delimiters for that terminal may be better carried out in the terminal. This is a question of cost reduction to be solved in a detailed design. The detector in the terminal could interpolate a suitable status character, leaving the interface computer to carry out its normal role of acting only on status characters.

For output, the packet travels to Y and is stored, and a character by character transfer to the printer then takes place. When the packet is exhausted, which is indicated by a control character, an acknowledgement returns to the subscriber’s computer which sends the next packet, if any. If, occasionally, two packets make up a line of type for output, the second one should not wait until the first is exhausted, but be sent and held by interface computer Y.

The control functions described for the two examples have mostly involved interface computer Y and this feature spreads the load caused by a subscriber’s computer conversing with many character-by-character terminals. Interface computer X checks the source of incoming packets and their sequence and adds the source field to outgoing packets.

Other problems will require different features in the control procedure, which will be developed as experience in data communication grows. The ability of the network to take part in the control procedures that involve the users contrasts with ‘data transmission’ as it is usually understood where the network is transparent to control information as well as data. Centralisation of some of the tasks of controlling data transfer in the interface computers is one of the factors that will make the network economically attractive.

Design of the links and node computers in the high level network

The high level network handles packets which are in a format shown in Figure 3. The length of a packet can be any multiple of 128 bits up to 1,024 bits . The 128 bit unit is called a *segment* and its length was chosen to give flexibility to the size of packets without complicating their handling by the computer. For most purposes the format of a packet can be considered in terms of 8-bit bytes, a segment containing 16 bytes and the maximum packet 128 bytes.

The format of the first segment differs from the rest since it contains the transmission envelope of the packet. The first 4 bits of this first segment comprise a packet start code which distinguishes standard packets from link messages which are used under error and overload conditions and also serve to identify the start of a packet after an ‘idle’ sequence. The *indicator* is a tag which is allocated to a packet during its stay in a node and remains as its identification until the packet has been accepted by the next node. In order to minimise the processing time, the packet is always left in its original store location and is represented in the queuing process by its indicator. The indicator is used to access the storage area at the output stage when the contents of the packet are transmitted onwards. During the packet processing, the indicator is used to read out the next 20 bits of the first segment which gives the destination address. The source address may be used under fault or overload conditions to ‘return to sender’. The first segment may then contain 7 bytes of message content. No routing information is carried in a packet. The output link to be used is determined by each node in turn by reference to the destination. It is possible under heavy load conditions that a packet may be caused to circle around its required destination node. In order that a packet may not remain in orbit for ever under these conditions an 8 bit *handover number* is increased by one at every node. When the handover number reaches a certain value, the packet can either be returned to its source or dropped.

Every segment carries a *more bit*. This bit is used by the link hardware to detect whether a segment is the last in a packet. Every segment also carries a system check sum used by the hardware to detect link errors.

The protocol for the link between two nodes can be described briefly as follows:

While it has packets in the appropriate queue, the node will send out packets at its own rate over the link. The whole packet is sent without stopping, but a predetermined interval is inserted between successive packets. Any interval between packets, is occupied by an idle sequence, the essential property of which is that the beginning of the next packet must be unmistakable.

Special hardware at the receiving node detects errors in transmission. When an error is found, a short message in redundant form is returned along the link; this is an example of a link message and is known as *Trace*. A Trace is detected, again by special hardware, and the necessary retransmission is carried out. The packets in an output queue are not erased when they are sent, but are held for a while in a continuation of the queue called the trace queue. The special short message which requests retransmission quotes the indicator of the packet at which retransmission must start.

There are other variants of these short, redundant messages used at the level of a single link. For example a *Shut-up* message tells a node at the far end to stop sending because the receiving node is in trouble.

The hardware of a node therefore consists of a general purpose computer and special units dealing with the input and output of each link. An outline design has been made and some optimization of design carried out for a node serving five links. A small, 16-bit computer of modern design was assumed and the logic design for the special units was carried out. Programmes for the main functions were written. The design has not been tested. The cost of the link hardware for five links equalled that of the computer.

A block diagram of the hardware associated with receiving data from a link is shown in Figure 4.

The input staticiser is a 16 bit shift register. There is also a 7 bit counter for the check sum, which is fed from the data line. When each 16 bit word has been staticised it is shifted in parallel to a 16 bit transfer buffer to make way for the next 16 bits which follow contiguously. When the transfer buffer is loaded, an autonomous data transfer into the computer store is initiated. A control counter in the link hardware controls the staticization process, determines the end of a segment and initiates the comparison of the generated check sum with that held at the end of the segment. The more bit is inspected and if it is the last segment in a packet, an interrupt to the computer is generated to inform it that a new packet has been stored.

Whenever a new packet arrives, the Indicator field in the first segment is extracted by the hardware into an Indicator buffer. It is held in this buffer until the whole packet has been stored. If this is accomplished with no check sum failures, the indicator is transferred to a Last

Successful Indicator register. If a check sum fails in any segment of a packet, this last successful indicator is encoded into a trace message and passed directly to the corresponding link output hardware. It is then dispatched to the sending node. The link input hardware which detected the failure remains in a state where it passes no data to the computer store until it detects a packet whose indicator matches that held in the last successful indicator register. This packet is also ignored as it was the last one successfully received but the Fail trigger is reset and normal communication for subsequent packets is resumed. In this way, errors in the indicator field do not harm the system.

The link input hardware receiving a trace message detects the identification character at the front which signifies a link message and transfers the packet to a decode matrix buffer instead of the transfer buffer. The decoded output of the matrix is then transferred to the computer with a special control word which indicates to the central processor that it is a trace demand.

The sending unit is similar to the receiving unit and will not be described.

Software organisation of the Node computer

The control programs handle the queuing and routing of packets, keep a check on the state of all neighbouring nodes and links and perform all other necessary 'house-keeping' operations. These tasks are divided among three programs operating at different priority levels, activation of the appropriate level occurring through the medium of interrupt signals; communication between the programs is made by means of various flags and pointers. Problems of interaction, which may arise because the three programs operate effectively in parallel on common data, are largely avoided due to the repetitive nature of the processing and because operations can be effected 'simultaneously' on opposite ends of queues.

A fixed area of store is allocated to hold packets; once a packet has been read in all manipulations are performed on its storage address, which is the indicator. Queues of indicators are maintained in circular buffers (list-processing methods are slower in practice); a 'free list' is used to control the packet storage space, while each duplex link has associated input and output queues, part of the latter being used to hold trace indicators.

The three programs are termed the 'Main Processor' (MP), the 'Input Processor' (IP) and the 'Output Processor' (OP). MP is always in execution when IP and OP are inactive. MP services the input and output queues of each link in turn on a round-robin basis, first entering an 'Input Queue Subroutine' (IQS) and then an 'Output Queue Subroutine' (OQS). Following the breakdown of a link, the round-robin is interrupted while the appropriate output queue is redistributed, indicators being placed at the top of successive input queues. MP also has the task of performing periodic machine tests and of shutting down the links when a failure is detected. IQS takes the top item from the input queue, increments and checks the handover number, updates the packet indicator field and selects the appropriate output queue. The routing makes use of a table containing output link numbers in order of decreasing preference, for each possible destination. The most preferred output queue is selected for which the link is not closed or broken down and whose queue is not full; this process is a very simple realisation of the network's adaptive routing strategy.

OQS initiates transmission of the top item in the active output queue, first checking that the link is available. The indicator then automatically moves into the adjacent 'trace' section of the output queue. If the top item in the trace queue has expired, the indicator is returned to the end of the free list. (Since at most two further packets can be output before a 'trace' link message is acted upon, the trace queue need hold only three indicators; by the time the top item in this queue is the fourth member, the corresponding packet must have been safely received by the adjacent node so that its indicator can be removed.)

OP is activated by low priority interrupts, from the output hardware. Its function is simply to set a flag indicating that the hardware is now free. OP also checks to see whether a link message had been generated while the hardware was busy, and if so, initiates its transmission immediately.

IP is responsible for the immediate processing of input packets and so is associated with a high priority interrupt from the link hardware. IP checks the type of the packet; if it is standard, the indicator is placed at the end of the input queue; if the latter is now full a 'shut up' link message is prepared for sending to the neighbour node (thus temporarily stopping its output to this link). Finally an indicator is taken from the top of the free list in preparation for the next packet. If a link message is detected, the type is decoded and appropriate action taken. A link breakdown message (generated by the hardware) or 'shut up' message causes the changing of a corresponding flag, while a 'trace' request causes the active head of the appropriate output queue to be re-indexed.

The total core store requirement is about 8K, 4K being reserved for packets since in the worst case up to 64 may have to be accommodated.

In designing equipment of this kind, the main problem of optimisation is the allocation of tasks between the computer and special purpose hardware. Three designs were developed in outline; the preferred design described above and designs with more and less hardware respectively. The preferred design showed a much better cost/performance ratio than the others, which indicates that it is not far from the optimum.

Performance of the Node

Approximate estimates have been made of the peak traffic that the node can handle as well as the queue lengths and delays as a function of traffic. Only simple queue theory was employed because at the present stage computer simulation would be an unnecessary refinement.

The node computer was assumed to have a 0.6 microsecond cycle core store and a powerful order code. If typical operating conditions are such that one Trace and one Shut-up link message are sent and received for every 50 standard packets, then the mean central processor time absorbed by each standard packet is approximately 240 μs. This figure is essentially independent of the average number of segments per packet and of the number of links handled by the node. Input and output from store (cycle stealing) accounts for 20 microseconds per segment. Thus if the mean packet length is 7 segments, the node can handle 2,600 packets per second, which is equivalent to 250,000 bytes per second of message content (i.e. excluding the transmission envelope).

Because the network operates asynchronously it is possible for packets to arrive at all inputs to a node simultaneously; all packets must undergo essential input processing before the node is able to receive any more, which means that every transmission must be followed by an idle period. For the design in question this period is approximately 500 microseconds. Provided that the node has more than two links, however, the peak traffic is still computer-limited.

In order to obtain estimates of the delays and queue lengths in a node, a model of the system was constructed which was simple enough to admit analytical solution. The model consists basically of three queues in series. The first queue represents input interrupts and is served by the Input Processor. The second queue represents the set of input queues and the third the set of output queues, both being served in turn by the Main Processor. The inputs to each queue were assumed to have a Poisson distribution. The service intervals and durations were treated as regular but were made to depend on the mean throughput rate of packets (λ) and mean number of segments per packet (S). All processors are slowed by the input and output of segments, the factor being a function of λ and S. Similarly the Main Processor is effectively slowed by interrupts into the Input and Output Processors, the factor being a function of λ only.

The mean total delay to a packet traversing the node was computed as the sum of the mean delays in each queue and the time taken to staticise the incoming segments; a plot of this quantity against λ for several values of S is shown in Figure 5. The asymptotes represent peak throughput rates. As a rough statement of the results obtained, if the traffic is below 80% of its peak value then the mean total delay time is less than 1.5 milliseconds, the total number of packets in the node at any instant is less than 3 (plus the contents of Trace queues) and the probability of a Shutup link message being generated is less than 1%.

The very high data handling rate, compared with telegraph message switching systems, is obtained for three main reasons:

a) the use of a packet format which is designed for computer handling, obviating the treatment by the node of the message content of the packet;

b) the simple routing and control policy adopted in the network, whereby each node takes cognisance only of its immediate locality;

c) a proper division of work between central processor and link hardware, the latter carrying out repetitive tasks such as byte assembly and error checking in parallel with, and more efficiently than, the central processor.

The interface computer

Interface functions could be carried out by the same computer system that performs the function of a node. It is however simpler to consider these aspects separately. The organisation of the interface computer represents the most critical part of the network design.

The two primary functions are to control the communication between different peripherals and to reconcile the fixed format of the high-level network with the widely varying requirements of the users. Other facilities which could be provided at a lower level of priority are, for example, directory enquiry services, code conversion and subscribers address tables. A secondary function is to collect the information necessary for logging and accounting.

For each user that is currently active the interface computer will maintain certain status information, such as the point that has been reached in the protocol and the current destination of packets from that user.

The peripheral logic associated with the interface computer is closely related to the local network, one form of which has been studied in detail, but the software is dependent on the protocol employed for interaction between users via the network. The extent of the software depends on the variety of control procedures allowed, which is not yet known.

It is difficult to determine how many subscribers an interface can handle but a rough estimate is that around 5000 simple consoles or 50 fast line printers could be simultaneously active. This assumes that only basic transmission facilities are being used. The cost of a computer suitable for such a load might be £20,000–£40,000 depending on the need for backing storage.

The interface is a single point through which all local traffic must pass and so is a potential bottleneck and source of unreliability. To overcome this, an implementation might actually consist of two or more computers and more than one link into the high level network. Alternatively the node and interface functions could be combined in this complex. Spare computing power could then be used to provide the extra facilities mentioned.

The tentative nature of the above conclusions concerning the interface computer should be noted. More definite information depends on a fuller understanding of the interface design for typical peripherals.

The local network

A design for a local network has been partly developed with the assumption that special cables will be provided for the purpose.

Subscribing devices were divided into two main categories:

a) Those devices Raving sufficient intelligence and possibly a high enough data rate to assemble their messages into packet-like blocks e.g. computers. These subscribers could be connected directly to the interface computer via lines resembling those used between nodes in the high level network.

b) Less complicated devices with various data rates which would require concentration in various groups and at various levels to make economic use of transmission paths and which rely on the interface computer for message assembly.

The prime considerations governing the design of the network to service the latter group were:

a) Short response time between peripheral subscribers and the interface computer.

b) Simplicity of out-lying equipments—both subscribers' terminals and concentrators.

c) The ability of the network to carry all data codes and procedures (i.e. transparency).

To satisfy the response time requirement, high data rate connections have been assumed and a system of on-demand byte multiplexing has been devised. This method of concentration also covers the requirements for equipment simplicity and transparency to subscriber code. The requirement for equipment simplicity necessitates there being only one point in the network with the intelligence for message assembly and manipulation—the interface computer. All subscribing terminals are connected to this point through concentrators.

In the particular design undertaken the concentrators join 8 cables to one. Not only can these concentrators be connected in tandem over serial links but they can in principle, also be stacked locally to any depth.

The design has been kept as general as possible but it would probably be used to carry data in the form of 8 bits as a unit. The concentrators handle this unit, together with various extra bits, in a store and forward manner, each byte waiting to be told before it moves up to the next level. No scanning of lines is carried out by the concentrator and under light load conditions a data byte would gain the attention of the unit on demand. In the event that two bytes arrive at the same time from different lines, a fixed priority system causes one to wait while that with the highest priority is dealt with. The concentrator can pass signals in either direction. Signals from the higher level of the system take priority over those coming in from the periphery.

The terminal equipments employed at the ends of serial links are the same at each level in the hierarchy. They also constitute the major part of a concentrator. This allows for best use of mass production techniques, and also permits the connection of a peripheral into the system at any level. Signals passing up the network from the periphery pick up a 3 bit address at each concentrator, which is used by the assembling computer to identify the peripheral. In the reverse direction each 3 bits of the address is inspected by successive concentrators to determine along which output line the signal should be sent. For every piece of information which passes over a link, extra signals are required to control the link and initiate any required repeat sequences. Where the information block is only 8 bits wide, this scheme achieves an actual data rate rather less than 1/4 of the raw bit rate of the cables, but the nature of the concentration leads to a high utilization of the links near the computer. It is perhaps appropriate to a compact region well populated by users, and was devised for use in this context.

As mentioned earlier, information is of two types. Status information is required between the computer and the peripheral in order that data mat be transmitted. Thus a peripheral will interrupt the system with a status word indicating that it wishes to transmit. It will signify the end of its transmission with another status word. Other status words are necessary to enable either device to abandon the exchange and to inform of inoperable states.

It must be emphasized that this is only one of many possible local network schemes and it developed in the direction it did because of the circumstances for which it was devised—a compact region, namely a large laboratory site, provided with cables for the purpose.

Network Economics

Any conclusions about network economics are necessarily tentative in the absence of a planned project. We nevertheless consider it worthwhile to give some rough estimates of the cost of the network services.

Two rough estimates were made, the cost of using the high level network and the cost of local line equipment. The first of these is of interest to the user with a high data rate. The second indicates the problem of cost reduction for the low-rate terminal.

The cost of using the node computer depends on the extent to which it is used. Its computing facilities can be idle for many reasons; the need to keep a margin of capacity in the busy hour, the random arrivals of packets and poor distribution of traffic among the lines it serves. With adverse assumptions for these factors a node might nevertheless handle 2 million packets per day, giving a cost per packet for transit through the network of about 1/20 penny, derived from the cost of the nodes. The cost of the links between nodes depends on the average distance between nodes. For Britain, assuming foreseeable P.C.M. developments, link cost will be roughly the same as the cost of nodes. The total cost per packet is 1/10 penny for the use of the high level network.

The cost of local line equipment can, no doubt, be greatly reduced by development of an integrated local network and the application of the emerging techniques of large-scale integration but we have assumed current I.C. techniques and the use of telephone pairs. The largest item of cost is the pair of wires to the nearest concentrator and the logic required at the ends. This cost is estimated at £700 per terminal, which is larger than we would like it to be.

As a rough indication, when the user’s mean data rate is one packet per 20 seconds during the working day, the cost due to using the high level network would equal the annual rental of the local line equipment.

The capital cost of the network, it seems, will lie mainly in the local distribution system, and the attention of communication engineers should be directed to the problem of cost reduction in this area.

In considering network economics, the cost of the network must be compared with the cost of the alternatives, including the control equipment at the ends of data links and the multiplexers and communication terminals attached to computers engaging in conversation with remote peripherals. The comparison has not been carried out because the data are not available.

Conclusions and Recommendations

The possibility of a common-carrier communication network for digital data has been explored and a particular system design has been described in this paper. The system has a number of features which are obtainable, if at all, only at greatly increased cost from systems based on adaption of the switched telephone network. These features are:

1. Control procedures to suit the terminals are provided by the network. This can reduce the cost of control equipment at the subscriber's terminal and concentrate many of the control tasks at central (interface) computers.

2. Bit rates to suit the terminal are provided by the network, with no restriction to standard rates. Data transfer can be sporadic or irregular over short time intervals. The use made of long distance facilities depends on the quantity of information sent, rather than time occupied at the terminal.

3. The network can provide transparency for data with no extra cost.

4. The routing of data is handled by the network which can provide the equivalent of a 'private wire' without tying up equipment.

5. The network design can be adapted to take account of technological advances in many ways. Almost the sole standard (which it would be difficult to change) is the packet format.

Maximum packet size for the whole network can be increased, and the speeds of links and nodes can be increased piecemeal. Local networks can be modified and improved and a control procedure over the local network can be chosen to suit special conditions. A certain number of overall control procedures must be standardised, but development of new procedures or variants is possible.

The rate of growth of computer-based systems depending on rapid response at remote locations, and the number of new schemes of this kind now being developed, indicates the urgency of decisions about digital communication networks. A leisurely sequence of market surveys, economic studies, system design and experimental networks would lead to a nationally-planned digital network in perhaps ten years, but the demand will arise much sooner. Delay will give rise to many separate, special-purpose networks, and a completely uneconomic method of solving the problems.

The authors of this paper believe that only a simultaneous attack on all aspects of the problem will produce a working network quickly enough.

We hope to have shown the way in which the system design will develop for a digital network to meet economically the full range of users requirements and allow full advantage to be taken of future developments in computer and transmission technology.

REFERENCES

1. Data Transmission-Current Trends and Future Prospects. J.W. Halina, Electrical Communication, Vol. 41, Nov. 1966, p. 177.

2. On Distributed Communication. P. Baran, Rand Corporation Memorandum, RM-3420-PR, Aug. 1964.

3. Trends in Digital Communication. R.H. Franklin and H.B. Law, I.E.E.E. spectrum, Nov. 1966, p. 52.

4. Experimental 224 Mb/s P.C.M. Terminals. J.S. Mayo, B.S.T.J., Vol. 44, Nov. 1965, p. 1813.

5. An Experimental 224 Mb/s Digital Repeatered Line. B.S.T.J., Vol. 45, Sept. 1966, p. 993.

6. Wideband Data on T1 Carrier. L.F. Travis and R.E. Yaeger, B.S.T.J., Vol. 44, Oct. 1965, p. 1567.

7. The T1 Carrier System. K.E. Faultz and D.B. Penick, B.S.T.J. Vol. 44, Sept. 1965, p. 1405.

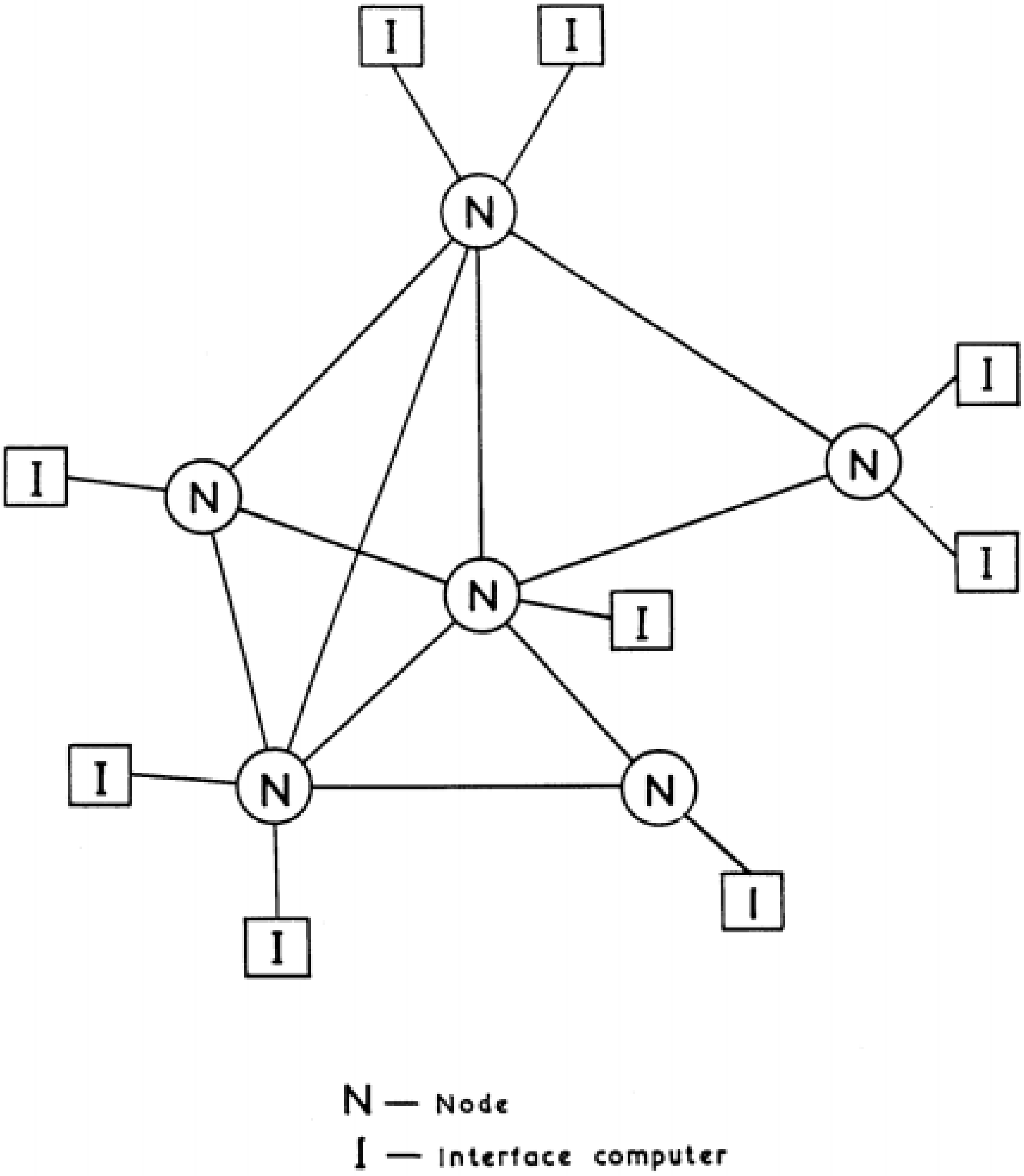


FIG. I. HIGH LEVEL NETWORK WITH INTERFACE COMPUTERS – AN EXAMPLE

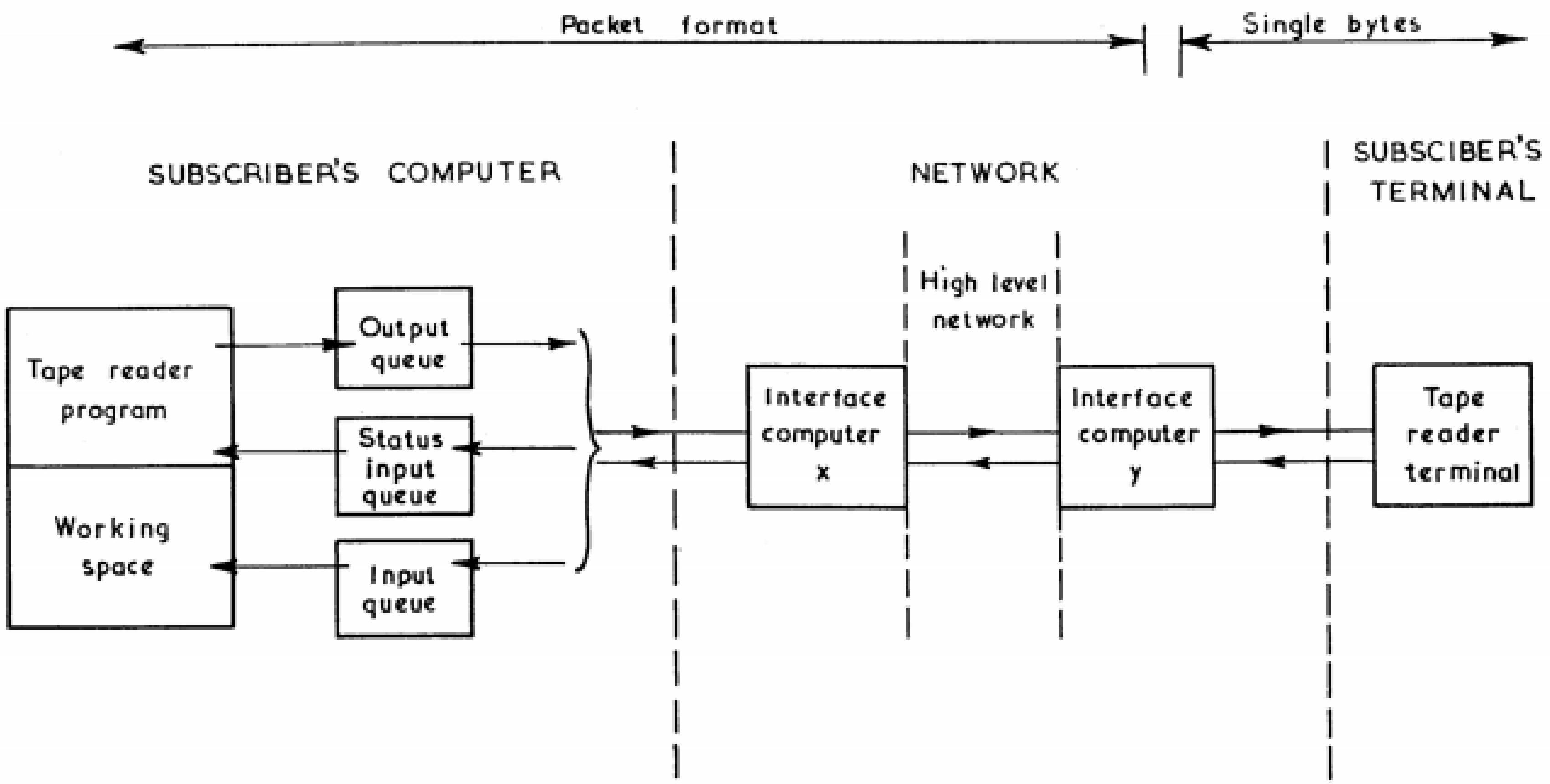
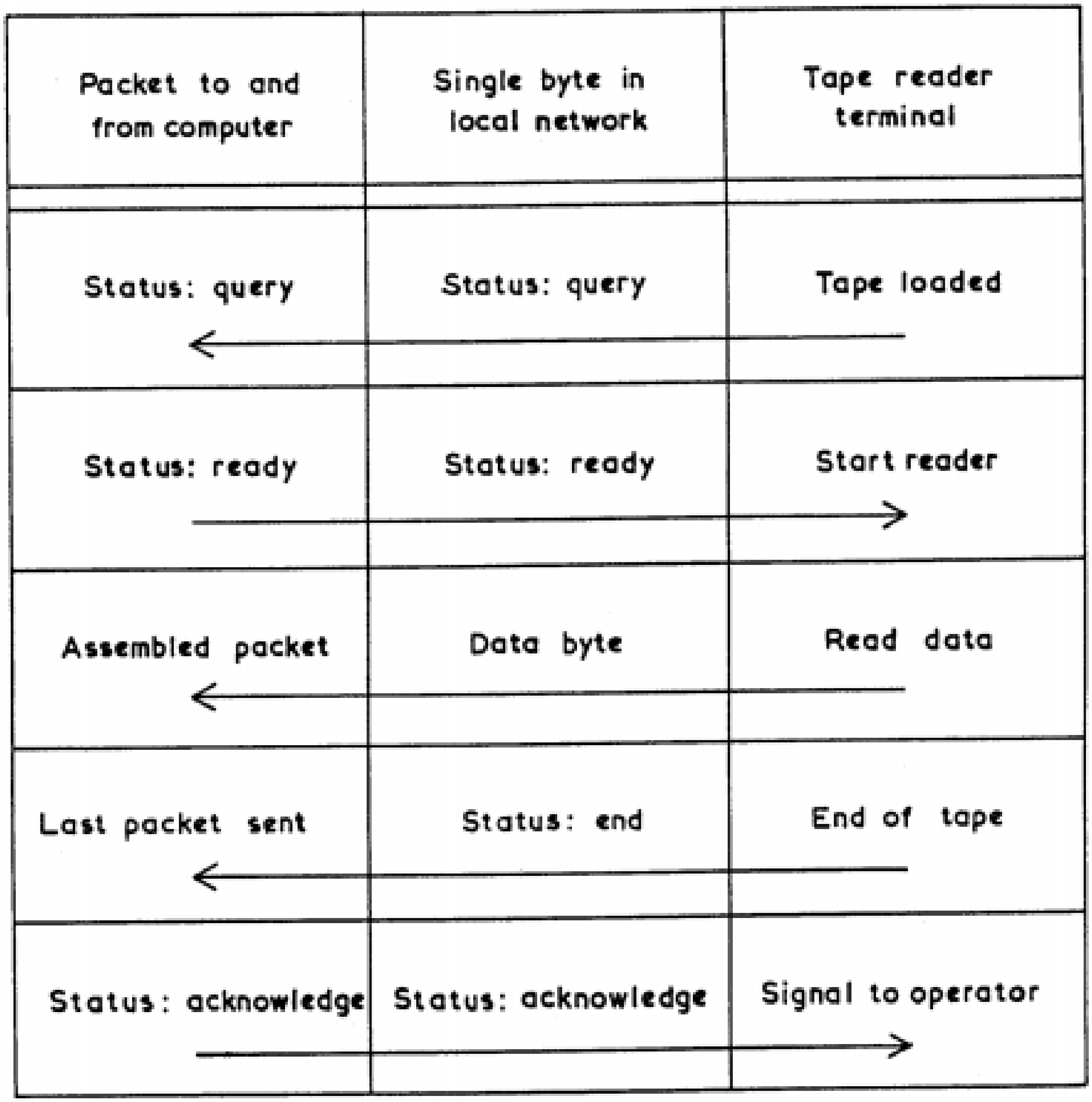
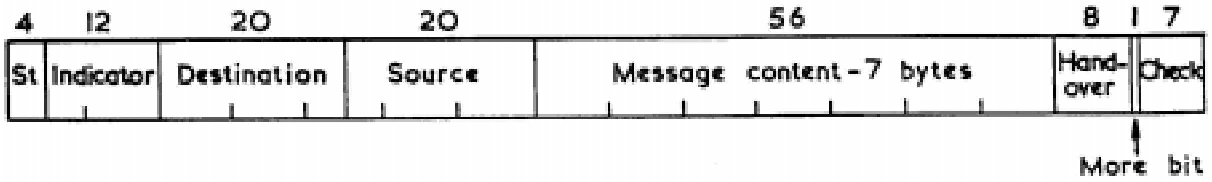


FIG. 2 a. DATA PATHS FOR AN ILLUSTRATIVE CASE.

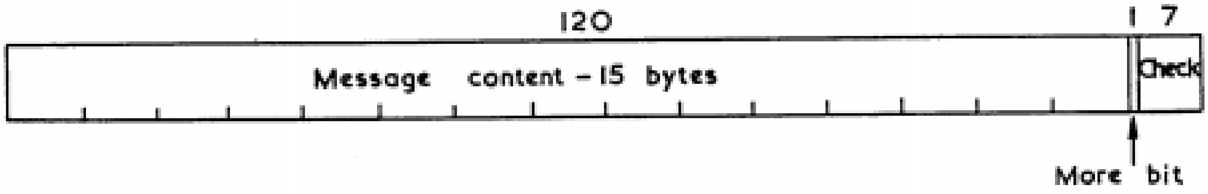


Packet to and from computer	Single byte in local network	Tape reader terminal
Status: query ←	Status: query	Tape loaded
Status: ready	Status: ready	Start reader →
Assembled packet ←	Data byte	Read data
Last packet sent ←	Status: end	End of tape
Status: acknowledge	Status: acknowledge	Signal to operator →

FIG. 2b. ILLUSTRATION OF A CONTROL PROCEDURE

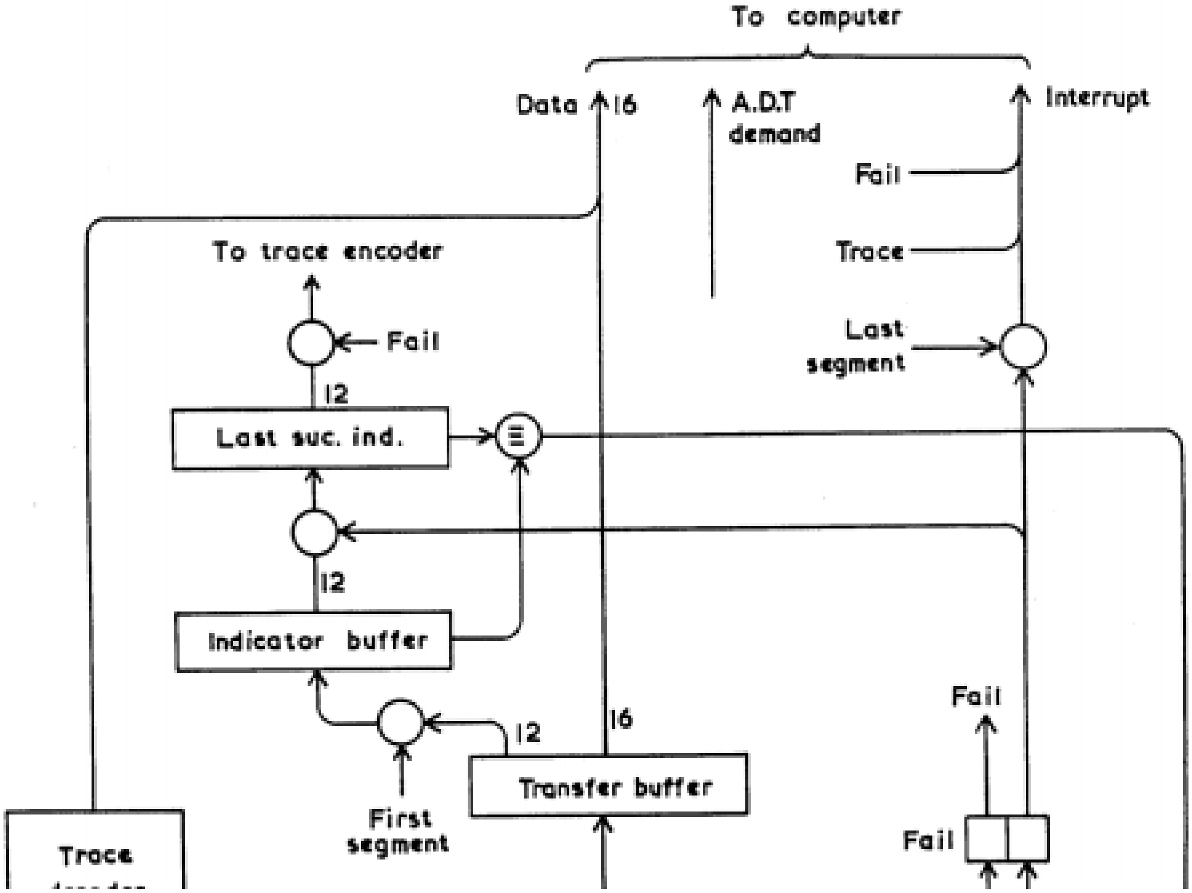


FIRST SEGMENT



FOLLOWING SEGMENTS

FIG. 3 FORMAT OF A PACKET


To computer
Data
16
A.D.T
demand
Interrupt
Fail
Trace
Last
segment
To trace encoder
Fail
12
Last suc. ind.
≡
12
Indicator buffer
12
16
Transfer buffer
First
segment
Fail
Fail
Trace