A DATA COMMUNICATION NETWORK FOR REAL-TIME COMPUTERS

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

Introduction

Computer applications which involve rapid-response to events at distant points create special problems in digital communication. 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. The user adds the terminal equipment necessary to make a data 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 (1).

The most closely related 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.

This paper is concerned with the system design of the network. Transmission paths for digital data are components of the system but 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 (3), (4), (5), (6), (7). The lack of operating long-haul digital paths presents a difficulty in the next few years. More expensive paths over frequency division multiplexed facilities will fill the gap. Rough calculations of the cost show that the more expensive long paths do not spoil the network’s economy for a country the size of Britain.

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.

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. The most common raison d’etre 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 less significant.

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

The application of the network to computer-machine systems such as the control of transport, electric power distribution and gas distribution is more uncertain because 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 and control information for the telephone switching network.

Outline of the Proposal

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. 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.

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.

There is no standard transmission rate, and no multiplexing in the sense of pre-arranged time-slots. Each link between nodes has a prearranged rate, but these can differ and be changed without trouble as the technology advances.

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.

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 2. 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.

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 distinguish 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 packing 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 it used to access the storage area at the output stage when the contents of the packet are transmitted onwards. Following the destination and source addresses may be 7 bytes of message content. The output link to be used is determined by each node in turn by reference to the destination. Because of the adaptive routing technique 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 receiving node hardware 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 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. The basis of the design was a small modern 16-bit computer. Logic design for the special units was carried out and 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 3.

The packet is staticised from the line 16 bits at a time and transferred to the computer through the transfer buffer. An A.D.T. (autonomous data transfer) demand is made for every word transferred. The Indicator is also retained in the Indicator buffer. If the whole packet is received without error, the Indicator then passes to the “Last Successful Indicator” buffer. The last word transfer of a successful packet is accompanied by an interrupt to the computer to draw its attention to the newly stored data. If the check run on any segment fails, the hardware at the receiving node requests retransmission by using the last successful Indicator in a trace message to the source. The receiving node computer is informed by interrupt that any stored parts of that packet should be dropped. The input link hardware causes no more data to be stored until the last successful Indicator packet has been correctly received again—but not stored. The next packet will be the one that failed and normal operation will re-commence. In this way, failures in the Indicator field do not affect 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.

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 ‘housekeeping’ 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. One queue is used to control the packet storage space, and 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). IQS routes the top item of the input queue. The routine 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. Indicators are later removed from the trace queue and returned to the free list.

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.

IP performs the immediate processing of input packets and so is associated with a high priority interrupt. IP checks the type of the packet; if it is standard, the indicator is placed at the end of the input queue. 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.

The total core store requirement is about 8K, 4K being reserved for packets since 64 may have to be accommodated when all queues are full.

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.

The node computer was assumed to have a 0.6 microsecond cycle core store and a powerful order code. If 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 independent of the average number of segments per packet and of the number of links handled. 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 simplified model of the system was constructed. 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 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 4. The asymptotoes 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 Shut-up link message being generated is less than 1%.

The very high data handling rate, compared with telegraph message switching systems, is obtained by the use of a packet format designed for computer handling, thus obviating the treatment by the node of the message content of the packet.

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 node handling cost per packet for transit through the network of about 1/20 penny assuming that the packet visits ten nodes. This is probably an overestimate of the node cost for a fully operational network.

Interface functions

The function of the high-level network is well defined and it has therefore been possible to produce enough of a design to show how it would work and at roughly what cost. Considerations of the interface functions and the local network have proved more difficult. Decision on the range of services which the network will provide depends on a compromise between the user’s detailed requirements and the cost of providing the exact variants the user prefers. Where the local network is concerned, much less is known about its capacity for carrying digital data at moderate speed than is the case for long distance circuits. In order to make progress it was necessary to use a real communication problem as the “guinea pig” and for this purpose communication between large and small computers, peripherals and a disc store within the National Physical Laboratory was chosen. The main traffic in this network will be between the main central computers and peripherals located around the site and between small computers which collect experimental data and the central services. A central disc store controlled by a small computer will be added because backing storage is frequently required by the small computers used for data collection and other purposes. In considering the detailed needs of users the following questions arose:

(a) The provision of routing information to the network. For simple terminals it was assumed that a destination for packets, once given, applied to all following packets until it was changed. A terminal may be allowed to specify the sources from which it would accept packets and in this case conventions are needed for the treatment of rejected packets.

(b) End to end error control methods.

(c) The control of data rate and confirmation of receipt. In a network where data rates are not fixed, it may be necessary to synchronize the source to the speed at which the destination can receive data. For a computer sending packets to many slow peripherals, further packets will be sent in response to a reply from the distant interface computer which will employ single or double buffering of packets according to the output speed in relation to delay through the network.

(d) Delimiters for the assembly into packets of data sent in the form of single bytes.

(e) Status and control information. In the case of a computer controlling a remote peripheral, in addition to the data there is also status and control information passing between the computer and the control system of the peripheral.

In the design for the NPL network, provision is made for the solution of problems (d) and (e) by the use of an extra bit in the characters passing between terminals and the interface computer. This distinguishes normal bytes from ‘status’ bytes. Packet delimiters will usually be detected at the terminal equipment and sent with a status indication so that the interface computer does not need to examine each character for the delimiters. In the high-level network, status characters will be sent as separate packets.

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 having sufficient intelligence and possibly a high enough data rate to assemble their messages into packet-like blocks, e.g. computers.

(b) Less complicated devices which reply 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).

The interface computer is the only point in the network with the intelligence for message assembly and manipulation. All subscribing terminals are connected to this point through concentrators.

A novel kind of line concentration has been devised which would 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.

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 signals should be sent.

As mentioned earlier, information is of two types. Status information is required between the computer and the peripheral in order that data may 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 the 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.

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. Rapid response. The complete transfer is completed in half the time required to set up a path in a switched network with comparable technology.

2. Tariff can be related to data traffic, not holding time.

3. Adaptability to improvements in technology. Packet format is standardised, not bit rates or codes. The speeds of links and capacity of nodes can be increased as traffic requires. Local networks can be of different designs, provided certain standard interfaces to the user are maintained.

4. Adaption to the users' requirements. Higher peak rates, different units of transmission and new control functions can be provided, always bearing in mind the cost of too great diversity.

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. Delay will give rise to many separate, special-purpose networks, and a completely uneconomic method of solving the problems.

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

The work described in this paper was carried out at the National Physical Laboratory, Teddington, Middlesex, England.

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.

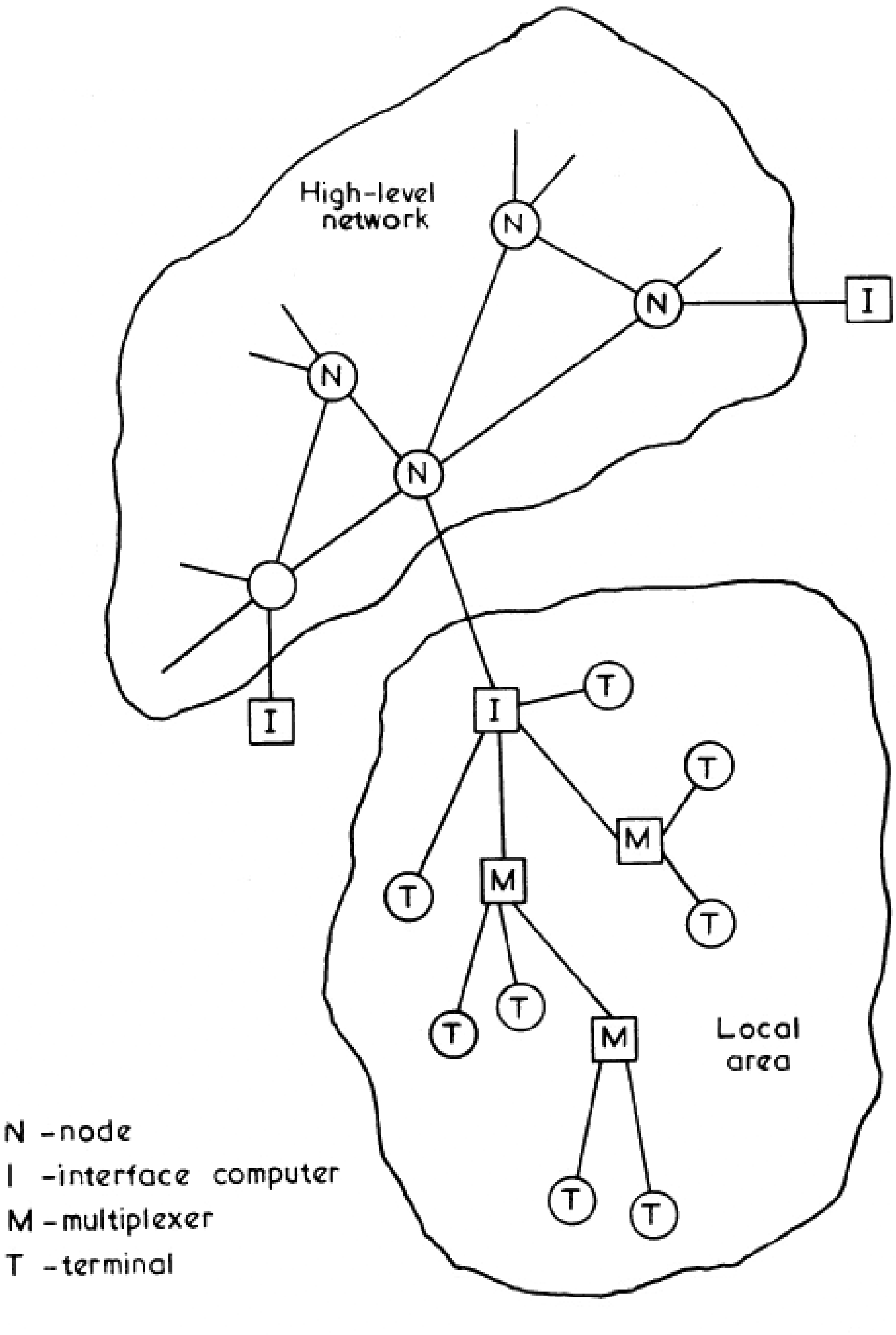

Fig. 1. Schematic Diagram of Data Communication Network.

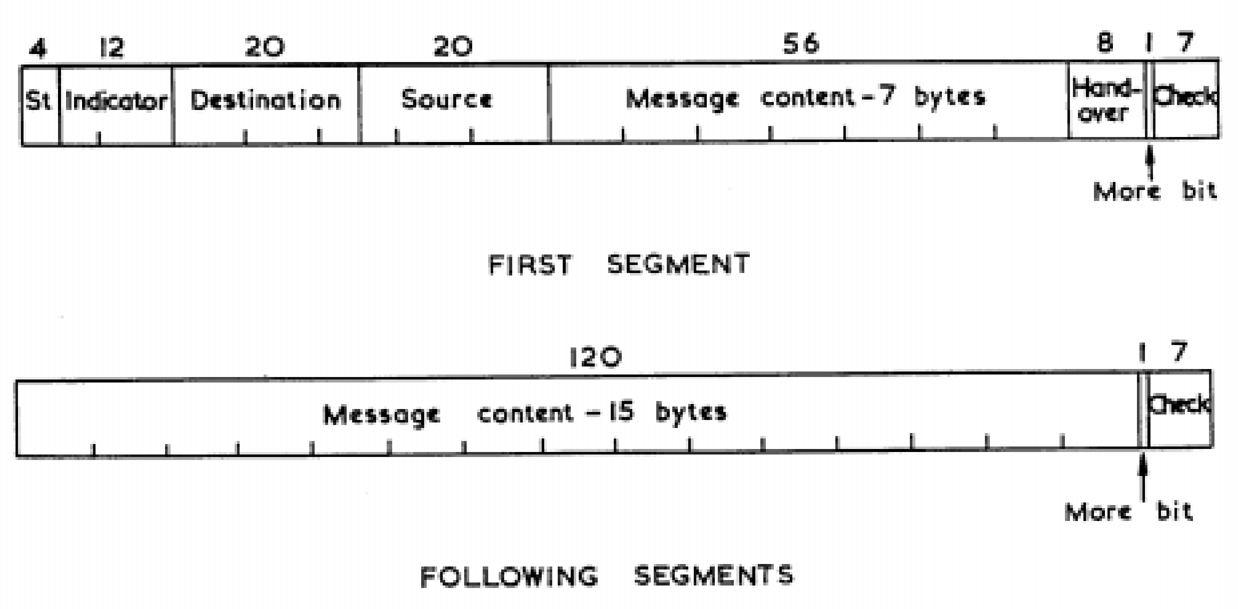

Fig. 2. Format of a Packet

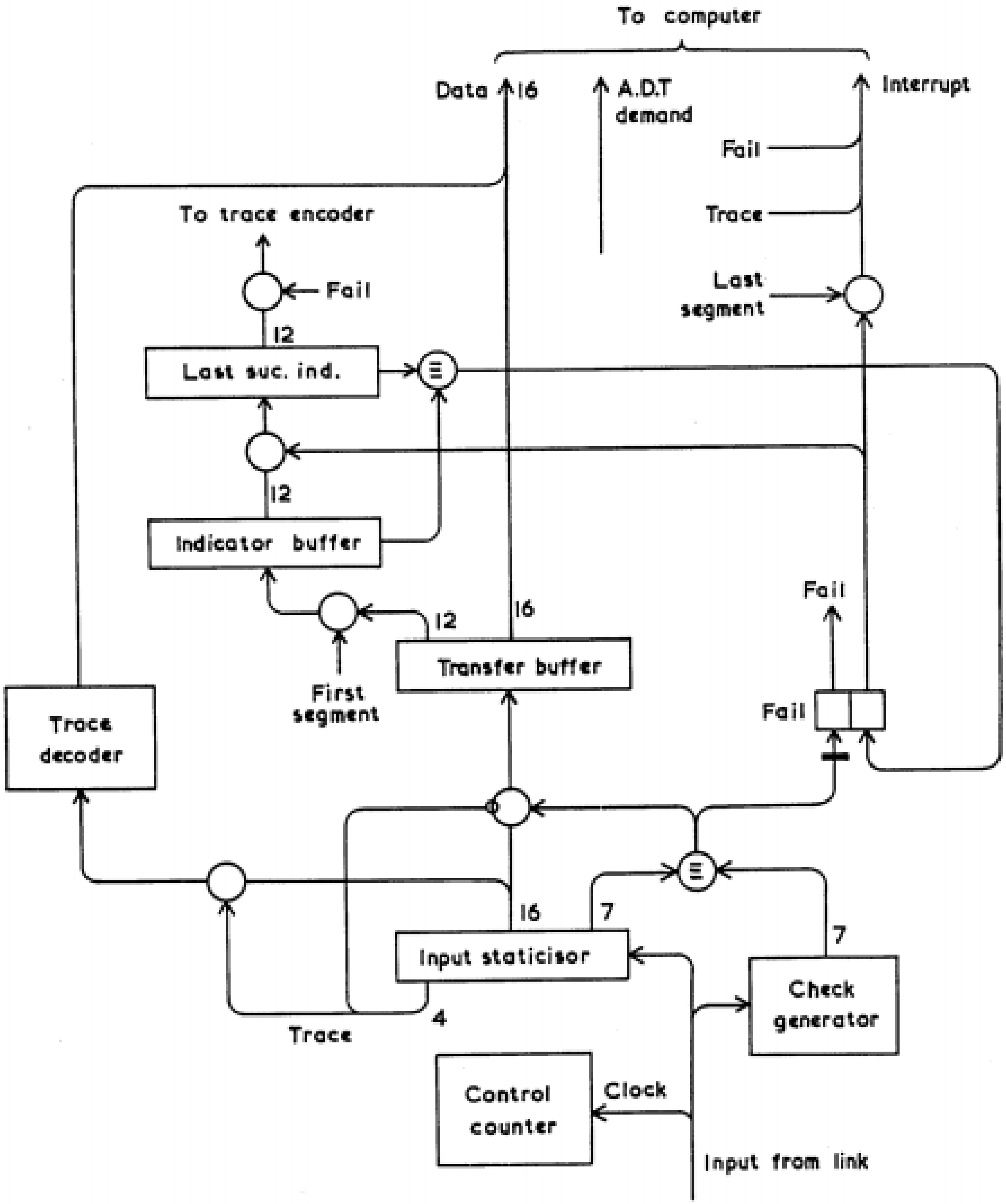

Fig. 3. Schematic Diagram of Link Hardware (Input)

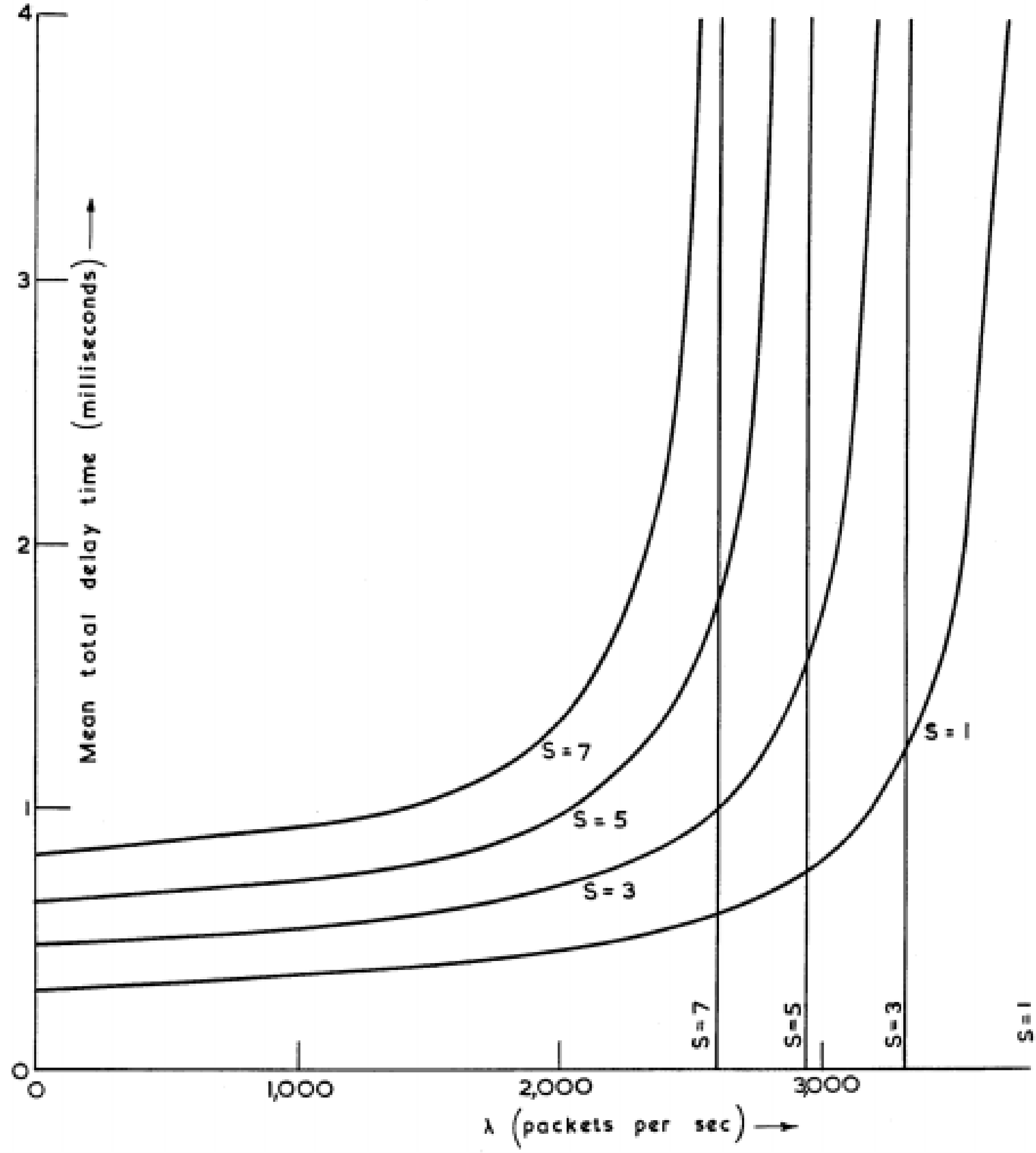

Fig. 4. Mean Delay Time as a Function of Traffic for Several Values of Mean Packet Length.