# PACKET SWITCHING NETWORKS

R.A. Scantlebury

<u>Rapporteurs</u>: Mr. J.S. Clowes
Mr. D.S. Jones

In this talk Mr. Scantlebury discussed some of the fundamental problems which face the designer of a computer system communications network. He illustrated his remarks by reference to a data communications network which has been built at the National Physical Laboratory employing the packet switching technique (Scantlebury and Wilkinson, 1971).

In a system based on the packet switching principle no physical link or circuit is established between parties engaged in a call. Instead, the parties send each other comparatively short messages called "packets" via a communication sub-network. Thus two computers may engage in a conversation comprized of a longer or shorter exchange of packets. The sub-network regards each packet as a separate transaction and so the load on the system is governed only by the amount of data transmitted and not by the real-time duration of the conversation. The concept of a call in the telephone sense does not exist in the sub-network therefore, but is known only within the communicating parties. Because of this the packets leaving one party for the sub-network must have an "envelope" bearing the address of the other party. To this end the data portion of a packet must be preceded by a "header" containing this information, and must be terminated by a "delimiter". Similarly, packets arriving from the sub-network bear the address of the sender.

A communications system employing the packet switching principle was first described by Paul Baran of the Rand Corporation (Baran, 1964) although he did not use the term "packet switching". Baran's network was designed for speech transmission. D.W. Davies recognized in Baran's work the invention of a new technique and adapted it to produce a data network (Davies, 1968).

In the early days (1966-67) there was some exchange of ideas between the workers at the NPL and people concerned with the design of the ARPA network. Not surprisingly, the two concepts are very similar but they are not identical.

One of the main differences between the two systems arises from the fact that the NPL designers always intend that their design should be for a public network to be administered by a public authority. In Great Britain this would be the Post Office. A basic principle of Post Office operation is that only traffic actually destined for a particular customer, or originated by him, may ever enter that customer's presmises. This constraint leads to consider the kind of network illustrated in Figure 1. Here the main trunk system, which carries messages between different customers, is the overconnected network linking the points marked 'N'. These objects, called "Nodes", are trunk switches and reside on Post Office premises. In function they are akin to the ARPA "IMP". In addition to the trunk switches there are also "Local Exchanges" called "Interfaces" and marked 'I' in the Figure. The Interface provides a customer with his sole means of access to the main trunk system. It was also part of the original NPL concept that the interface would be capable of properly conditioning the customers' messages so that the trunk switches would have to deal only with highly stylized packets. The trunk switches therefore are concerned solely with the correct routing of packets.

A theoretical study was carried out by NPL on the main trunk system. An appropriate switch was designed to prove an "existence theorem" and to obtain performance figures through a software simulation. The performance attaintable is indicated by the curves shown in Figure 2. These show that the mean delay time is very nearly independent of the traffic up to a certain load when the delay increases rapidly and the system becomes saturated. The curves also indicate that for low loadings the delay time is largely independent of the packet size but saturation occurs earlier for larger packets. The actual figures indicate that the switch can handle up to approximately 1 megabit with delay of about 1 ms. This now seems rather optimistic in view of the performance of the ARPA IMPs which achieve only about half or three-quarters of this rate.

These results encouraged the designers to believe that the trunk node part of the network shown in Figure 1 was a possibility and the Arpanet has since shown that such a network can actually be built. What remained to be done was to investigate the feasibility of the postulated Interface. In particular, how much processing power would be required; is it a PDP8 or a 360/195?

A theoretical analysis was attempted but this proved to be very difficult. The situation here is much more involved than in the case of the highly stylized trunk network where many complicating factors are absent or can be ignored. A simulation approach was also impracticable, largely because of the lack of data about the load distribution to be expected from the wide range of devices in the local network controlled by the interface.

It was decided that the best thing to do would be to build a switch to handle traffic on a network within the NPL. This project was originally funded on the basis that it was an experimental network made available to users because their traffic was needed as part of the experiment. In such circumstances, of course, the users soon begin to expect that the system should operate as a regular service. The NPL network, which has been operating for about two years, has now passed out of the experimental stage and is available as a 10.30 - 22.00 hrs. service.

Before describing the NPL switch it is useful to consider in a general way the structure of such a device. One important general question is "Is it possible to devise a strategy for computer communication which is independent of the nature of the high-level communications subsystem?". This is an important practical question because, quite commonly, the communications sub-network and the computers which use it are designed by independent teams and each group is concerned primarily with maximizing the efficiency of their own equipment. Some research on this topic has been done recently at the NPL and much good work has also been carried out by the French team at IRIA concerned with the design of the CYCLADES system (Pouzin, 1973). This approach leads to the kind of configuration illustrated in Figure 3, which shows that three main divisions are necessary within the computing system.

The first division shown in Figure 3 as an LCM (Link Control Module) is a "front end" which is responsible for driving the communications subsystem. Now, if the interface to the front end is well defined, then, hopefully, the user machine environment can be changed at will without affecting the operation of the communications network. This is called the Message Interface in Figure 3.

The second division is called the Inter-Process Control Module (IPCM) in Figure 3. This is where the computers agree between themselves on how to communicate. Decisions about formats, commands, etc., have to be made at this level thus providing a number of primitives from which a basic transport mechanism can be built. Words such as "message formats" or "protocols" have been used to describe these primitives. In effect the IPCM is a kind of multiplexor allowing communication between processes in different computers, these processes running in the third division which can be thought of as the space in which user programs run.

Another interface, the Process Interface of Figure 3, provides the means by which the processes running in the third division get a "handle" on the basic transport mechanism.

The configuration shown in Figure 3 is very close to what has been implemented in the ARPA network. The lowest level can be regarded as representing HOST-IMP protocol in the Arpanet while the next level corresponds to the HOST-HOST protocol.

Of course, even with the configuration as in Figure 3, it is still necessary that the communications subsystem and the computers using it should take account of each other's properties if maximum overall efficiency is to be achieved. In the present state of the technology, packet switching networks seem to be best adapted to the kind of traffic generated by computers. Certainly, most of the systems currently operating or being designed use this technique. One reason for this is that present-day computer operating methods tend to generate a rather "bursty" type of traffic load, for example buffered input/output. It is thought that traffic studies would show a bimodal distribution for message length.

One peak would correspond to fairly short messages of 30-40 characters from, for example, keyboards. The other peak would correspond to much longer messages of 1,000 - 2,000 bits. Packet switching is designed to handle this kind of load, by restricting messages to some maximum length and handling each message as a unit.

We now have a method of transporting information between different machines. The next question is "Where do the users fit in?". In particular, what do they want to use the system for? The ARPA people rightly assumed that the users wanted to share resources between their machines. They also assumed that the users would be at the centres where the resources were and would use their Hosts to access remote Hosts. In the event it did not turn out like this. Many users were not close to their own machines and some form of terminal handling facility had to be put into the system.

One of our original ideas was that the local exchange would be capable of handling both mainframes, which can generate properly packaged messages, and simple devices like tape-readers or key boards which are not capable of behaving in this well-mannered fashion. Thus it was envisaged that the kind of link-up shown in Figure 4 should be possible. Here is a simple device, in this case a tape reader, capable of emitting only single characters connected via its local exchange and the high level network to a distant machine.

Thus the local exchange, or interface computer, has to perform at least two functions. One is as local entry for packet devices and the other is to handle simple terminals. These two functions are reflected in the structure of the local exchange shown in Figure 5 which depicts the original NPL model of a trunk network of nodes and local exchanges or interface computers. Two processes run in the interface computer, the Communications Processor and the Terminal Processor. The Communications Processor handles packet traffic from local user machines. In function it lies inside the communications network interface in Figure 3. The Terminal Processor handles unpackaged traffic from local terminals. It is responsible for the proper packaging of this traffic and is treated by the Communications Processor exactly as if it were a user machine. The Terminal Processor therefore lies in the region marked Inter-Process Control Module in Figure 3.

The above indicates that the physical boundaries in the system do not necessarily coincide with any of the conceptual boundaries shown in Figure 3. Figure 6 shows how the physical boundary at the common carrier level might intersect the hierarchical boundaries in an actual system.

Coming down to practical details, at the NPL we have a single switch like the interface computer described above. It is just the local exchange without incoming or outgoing trunks, since the size of NPL does not warrant a high-level network. Since our experimental interest was the local exchange this suited our purpose but, of course, we can packet-up traffic and treat it as if it were to be transmitted on a high-level network.

The machine used is a DDP 516 with another as standby and we have a digital local transmission system with 1 Mbit lines. This network carries both kinds of traffic, packets and raw character data.

The software space of the DDP 516 is divided into three regions as shown in Figure 7. Basically there is a real-time operating system which "fields" the signals from external hardware, administers buffer pools and allocates run time to the other processes. On top of this there are two partitions, one for the packet switch and the other for the terminal processor.

Local peripherals attached to the DDP 516 are an operator console, a paper-tape punch for gathering statistics, and a device for automatically reloading the system from magnetic tape cassette in the event of a crash. There is no attempt in the present system to keep track of calls. If the system crashes it just bootstraps itself in again. The users do not seem to mind this which is rather surprising. The mean time between failures approaches one week.

We have a very simple set of protocols. This was an advantage in speeding development but created difficulties later when previously discarded "elaborate" protocols were found to be desirable. The formats used are illustrated in Figure 8. The packet working computers work into the switch in the top format. They send packets of any length up to 255 8-bit characters (the transmission is all character oriented). The header contains four fields:

| | | |
|---|---|---|
| 1. | Type code. | (1 byte) |
| 2. | Length. | (1 byte) specifying size of data field |
| 3 & 4. | Address space. | (2 bytes) |

The types of message identified are "data" and various control messages, for example, "error typecode", "error in length", "destination not available" etc. This is the level of processor-switch communication.

The data field itself is subdivded as shown in the lower part of Figure 8. It is here that we find the HOST-HOST protocol. Generally, Hosts are allowed to use any format or protocol they like, but we have been obliged to define a system protocol since one of the Hosts is the terminal processor inside our switch. Many Hosts use this for Host to Host communication. The format has four fields.

1. Type code - set up a call, break a call, etc.

2 & 3. Reference numbers identifying processes at either end.

4. Parameter field (N).

We have attached to the system about 100 terminals of various kinds, paper-tape readers to input jobs to the large machines, graph-plotters and display devices. We have two classes of computers attached - Hosts to provide services and others just using the system. In the former class there is for example a PDP11 offering a text manipulation service: two DDP 516's running a large Burroughs disc filing system with tape archiving. This is the store available for use by small computers. Job spooling for a KDF9 is also done here. A second KDF9 offers a time-sharing-like service via a front end. There is also an information retrieval system called SCRAPBOOK which runs on a modular 1, designed to be accessed by VDU terminals.

Future plans are to connect to ARPA and the new EPSS - the Experimental Packet Switched System.

Discussion Session

Professor Randell asked if packet switching was necessary for the design of the communication network described. Mr. Scantlebury replied that packet switching was not a necessity but 'simply an appropriate mechanism which fits well for many purposes'. Patterns of traffic for various applications were very different, said Mr. Scantlebury and it may be possible in future to design a very rapid switch to cope with the different patterns. Equally, it may be decided that there is no one method.

Professor Michaelson asked about an alternative method of packet handling by the switch, and not in the customer's processor at all.

Mr. Scantlebury agreed that the method was valid but remarked that 'whatever is provided will not be entirely adequate'. He continued by describing the design by the Post Office, which has adopted the method outlined by Professor Michaelson, for EPSS. The Post Office have put Host-Host protocol inside the customer interface to administer and control 'calls' on behalf of customers, said Mr. Scantlebury. He illustrated this by referring to Figure 6 again and commented that Host-Host communication should be hierarchical with well-defined boundaries in the hierarchy.

Dr. Browning asked about the problem of deciding how to route data from source to destination, espcially the problem of packets which get mislaid around the network or get out of order.

In reply, Mr. Scantlebury said that fixed routing was advocated because under light loading the traffic will keep along a particular path in the network, and under heavy loading it is better not to spread the traffic but control its input. Adaptive routing was also proposed. But at NPL, simulation studies had shown adaptive routing better suited for handling line failure rather than load control. Packets out of order must be coped with, if necessary, at the Host-Host level.

Professor Whitfield followed up by asking about the possibility of including sequencing information in the protocol to detect packets out of order.

In agreeing, Mr. Scantlebury pointed out that even in the simple format of the NPL system, there was a sequence number which allowed checks on proper working. Both ARPA & EPSS contained sequencing information in their message protocol.

Professor Michaelson then asked about the proportion of space taken up by the protocol in a message. Mr. Scantlebury replied by illustrating the EPSS network where a header of 10 bytes and packet of 255 bytes means the overhead is about 4%. For bulk traffic using long packets, he explained that this was necessary overhead. But for short messages, for example from a teletype, the protocol was of the same order of size as the message. And in reference to Professor Michaelson's earlier question, Mr. Scantlebury pointed out that by taking the inner protocol through into the switch, EPSS uses abbreviated addressing after a call has been established, leading to greater efficiency.

Mr. Scantlebury concluded by commenting on measurements of the distribution of message lengths. The ARPA network, is designed to have a message length of 8000 bits which is then packeted into 1000-bit packets. However the vast preponderance of traffic gets inside one packet because many of the current users are terminal ones.

'On the SITA Airlines Network, it is claimed that ninety per cent of all messages come within a message length, which is 255 characters. Fifty per cent are claimed to be less than 100 characters long, peaking the traffic in that region'.

## References

Baran, P. (1964) "On Distributed Communications". Rand Corporation Memorandum RM-3420-PR.

Davies, D.W. (1968) "The Principles of a Data Communications Network for Computers and Remote Peripherals". Proc. IFIP Congress 1968 (Edinburgh). Hardware D11.

Pouzin, L. (1973) "Presentation and major design aspects of the CYCLADES computer network". Proc. ACM/IEEE Third Data Communications Symposium, Tampa, USA. IEEE 73 CHO828-AC.

Scantlebury, R.A. and Wilkinson, P. (1971), "The design of a switching system to allow remote access to computer services by other computers and terminal devices". Proc. ACM/IEEE Second Symposium on Problems in the Optimization of Data Communications Systems, p.160. Palo Alto, USA. IEEE 71C59-C.

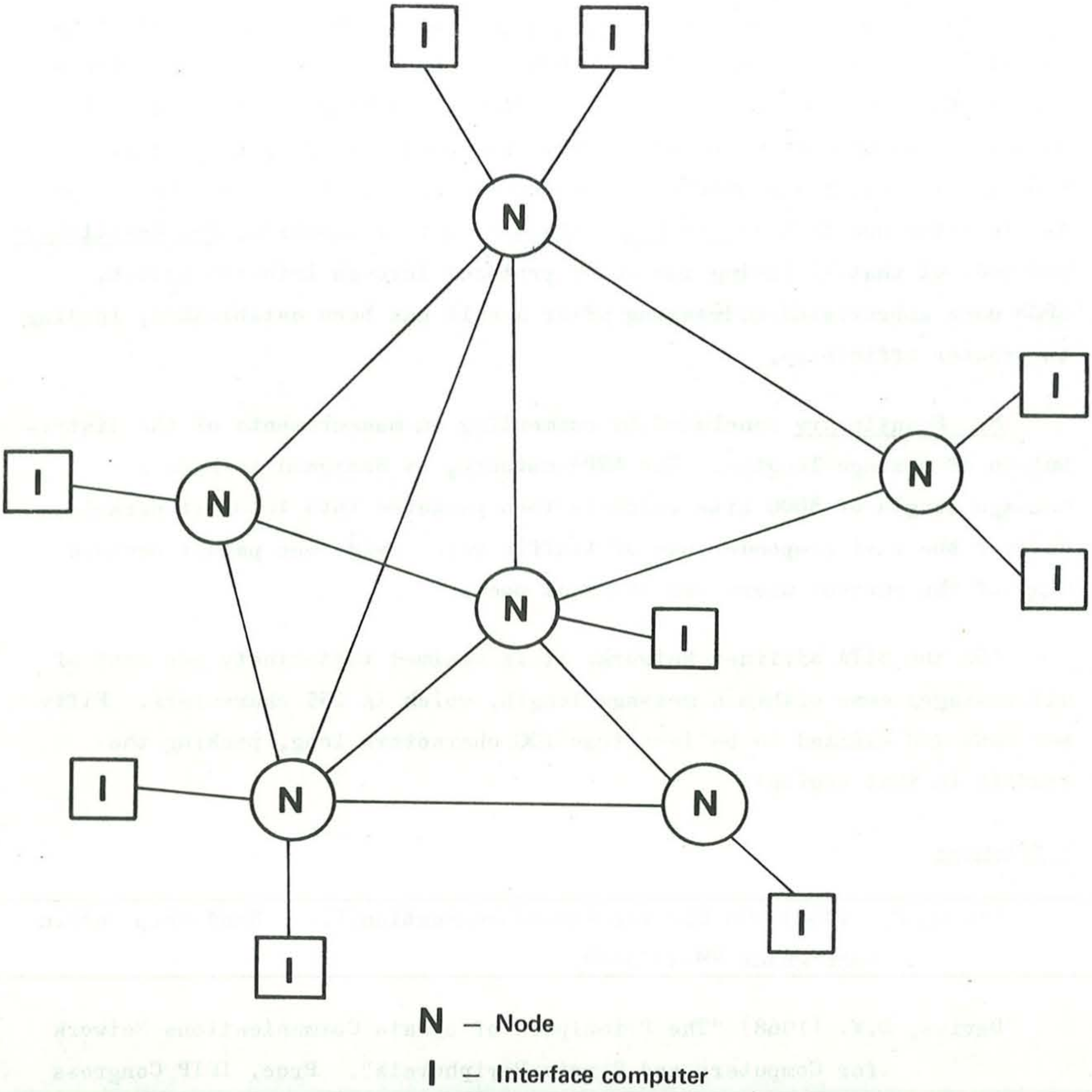

Figure 1 HIGH LEVEL NETWORK WITH INTERFACE COMPUTERS – AN EXAMPLE

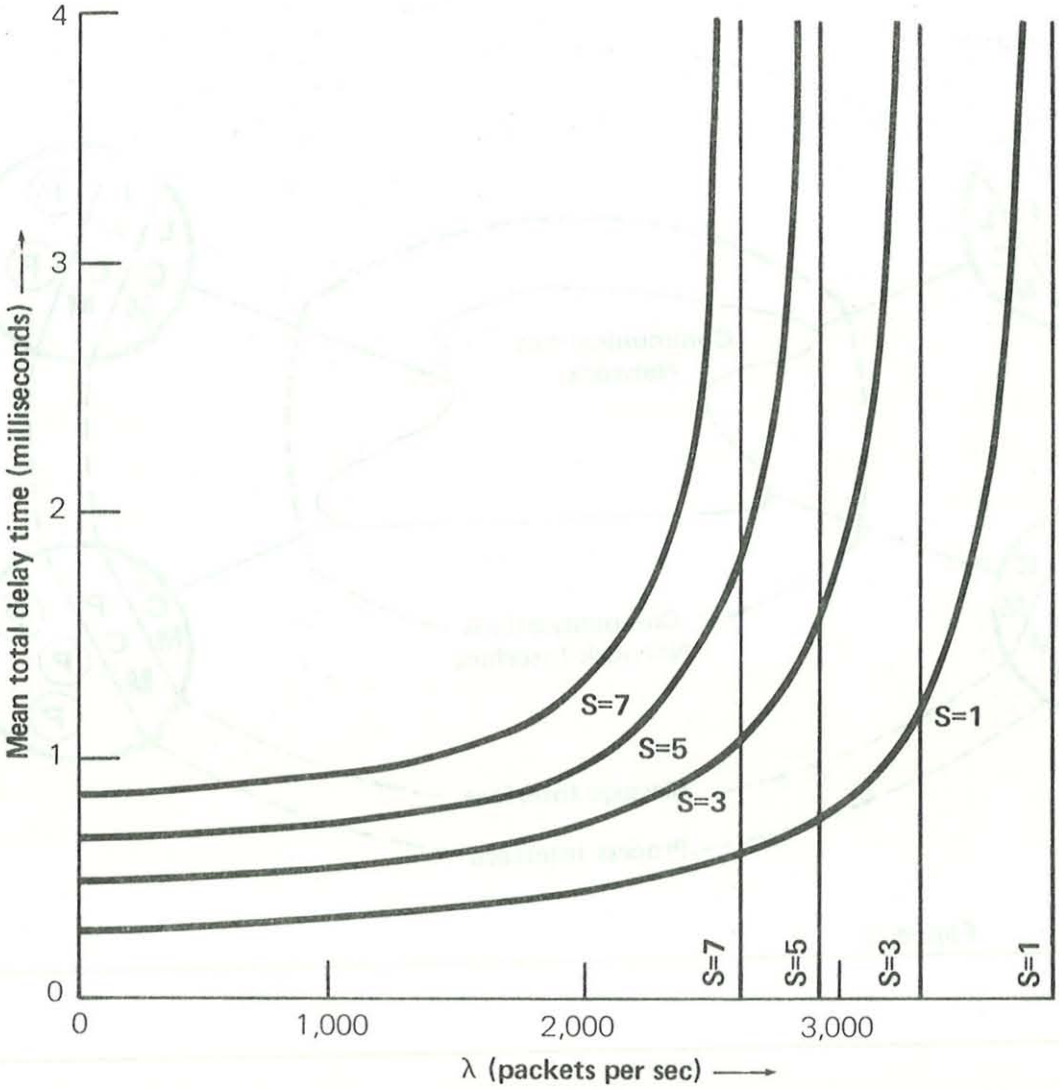

**Figure 2** MEAN DELAY TIME AS A FUNCTION OF TRAFFIC FOR SEVERAL VALUES OF MEAN PACKET LENGTH

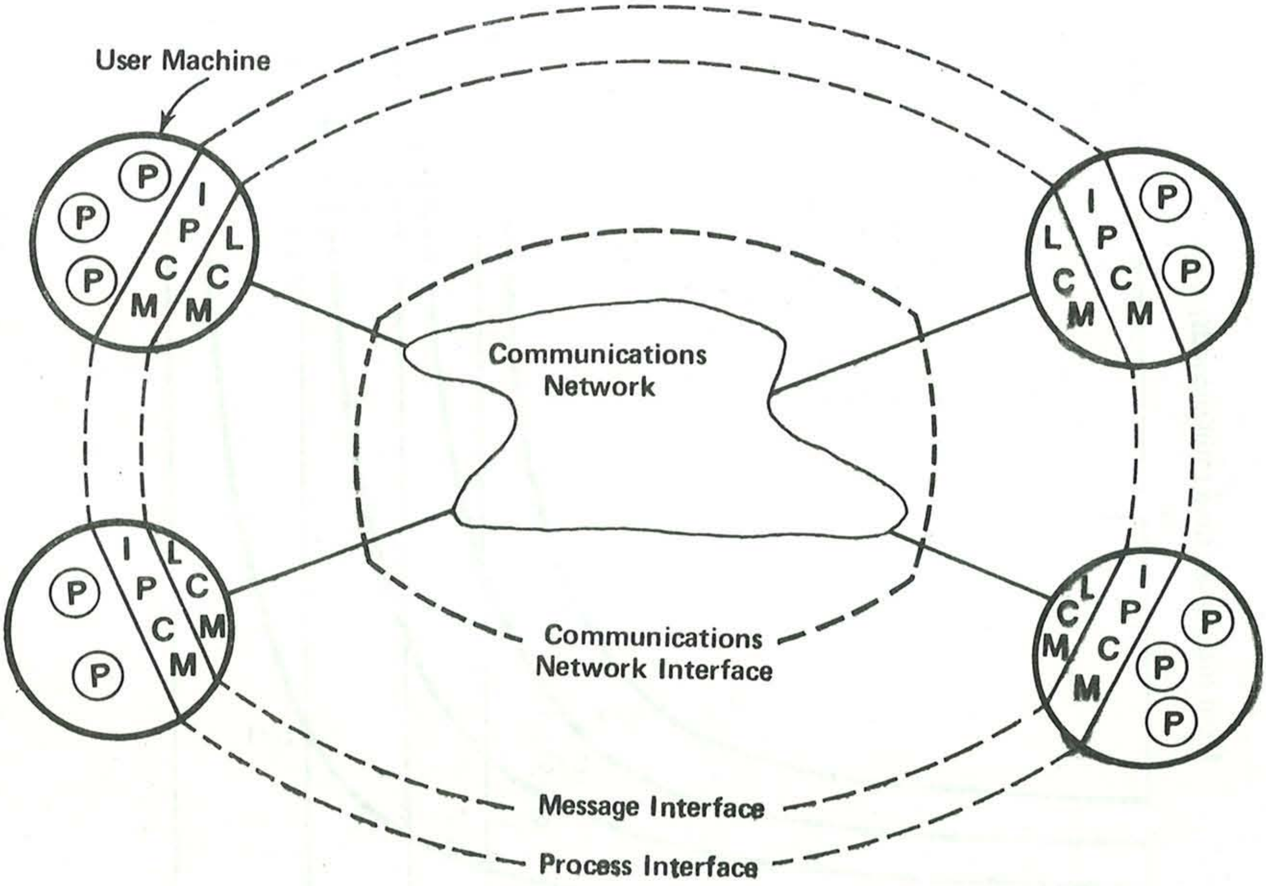
User Machine
P
P
P
I
P
C
M
L
C
M
Communications
Network
Communications
Network Interface
Message Interface
Process Interface

Figure 3

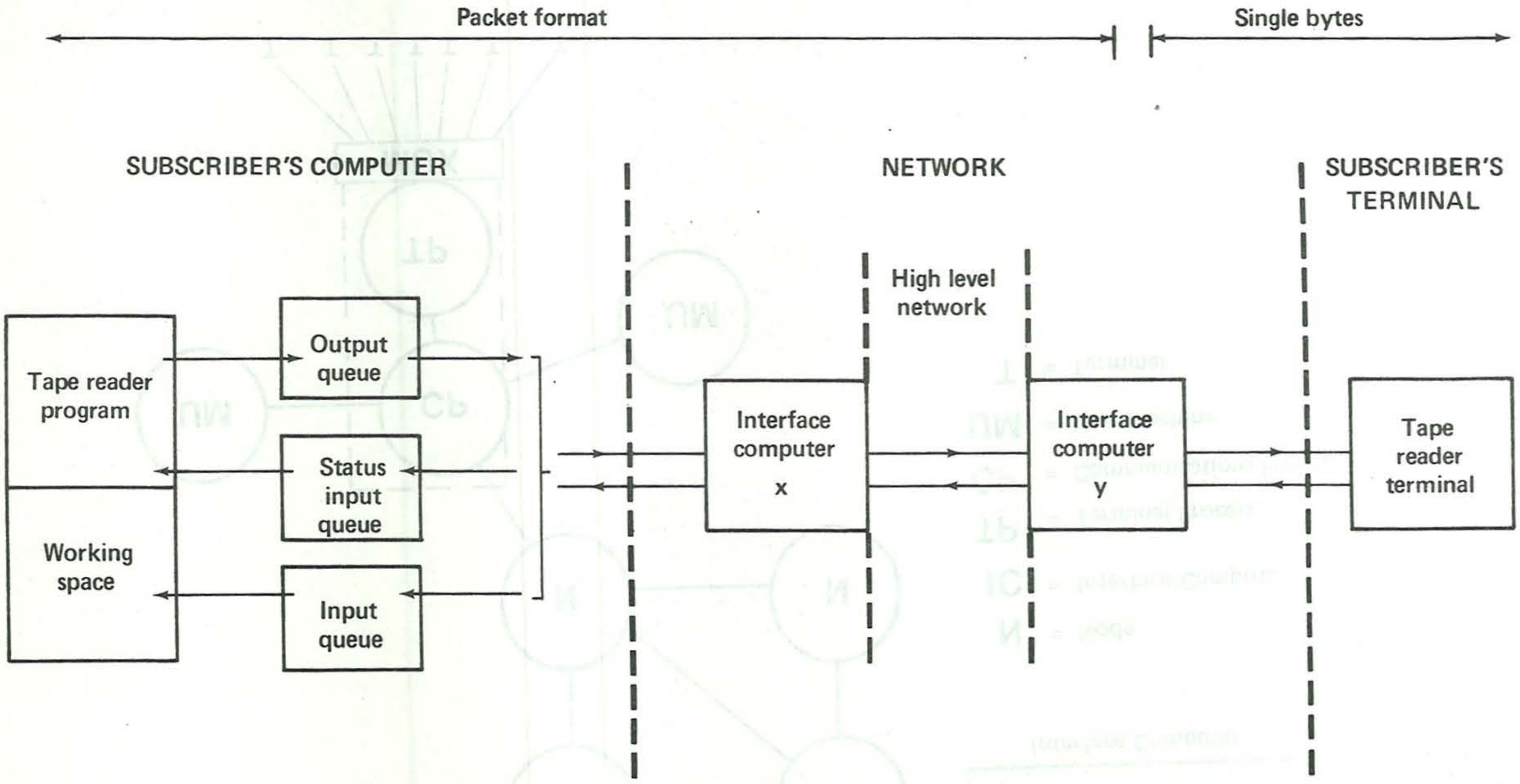

Figure 4 DATA PATHS FOR AN ILLUSTRATIVE CASE

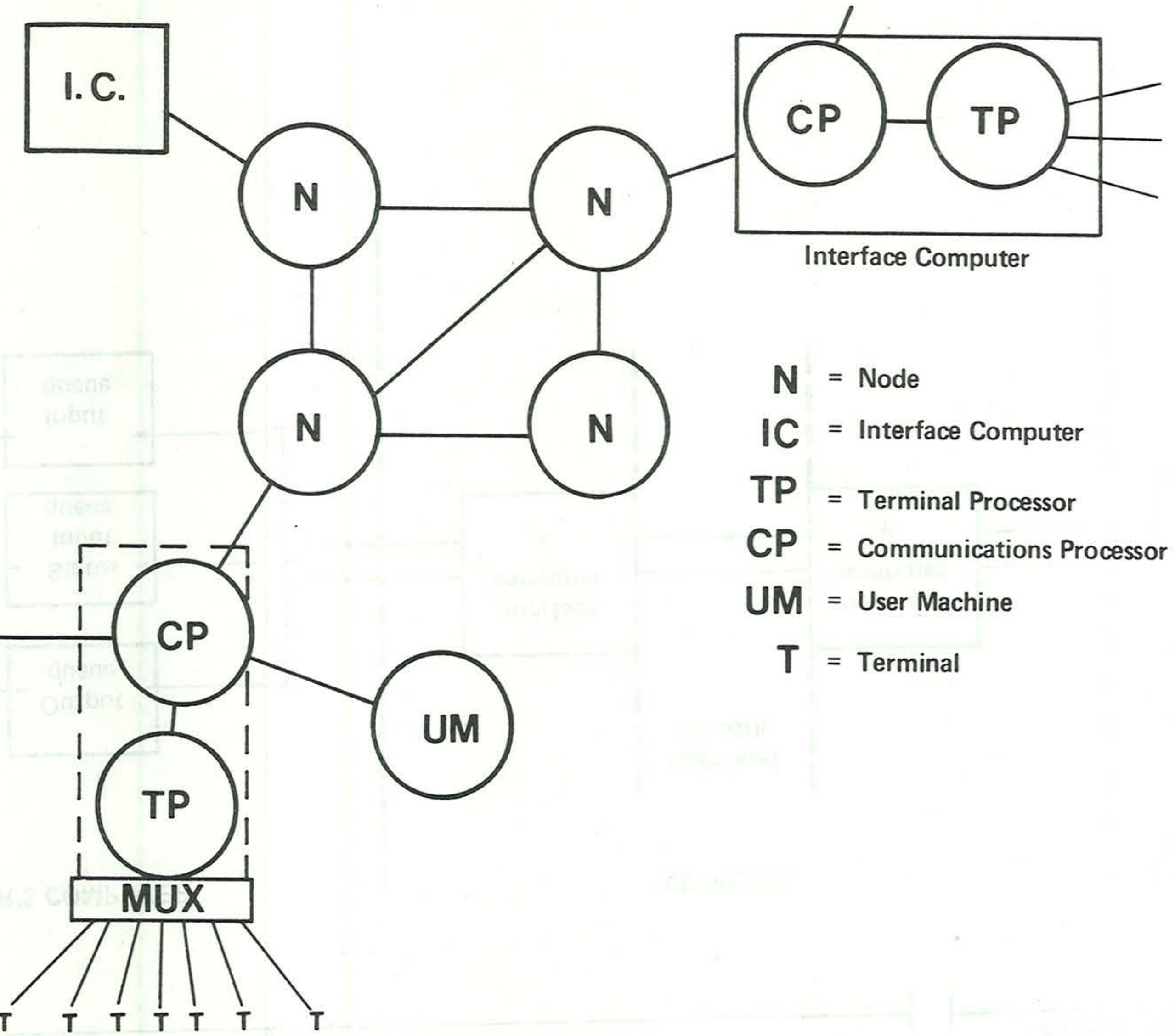
I. C.
N
N
N
N
CP
TP
Interface Computer
UM
CP
UM
TP
MUX
T
T
T
T
T
T
T
N = Node
IC = Interface Computer
TP = Terminal Processor
CP = Communications Processor
UM = User Machine
T = Terminal

Figure 5

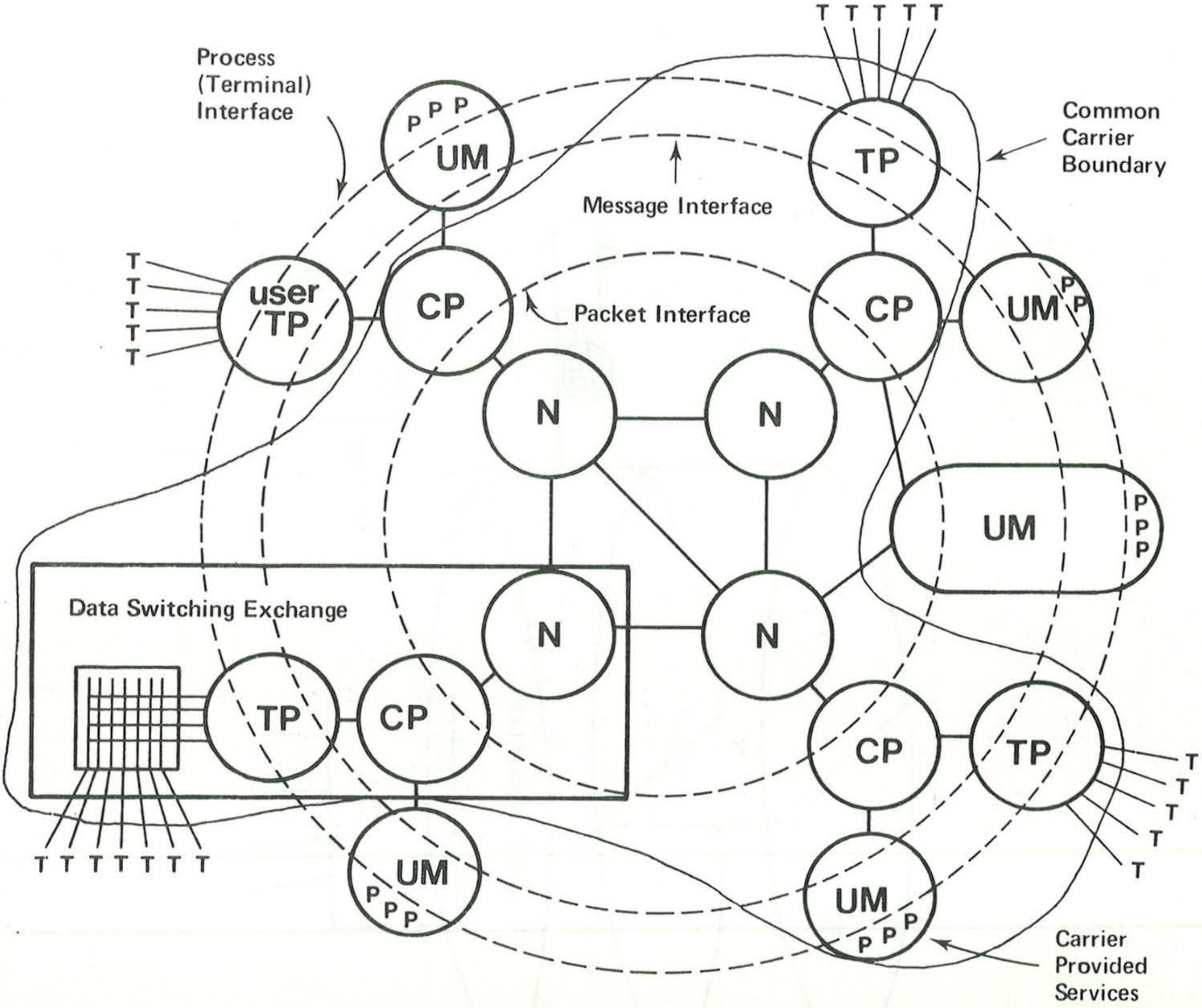

Figure 6 THE 'TOTAL' HIERARCHY

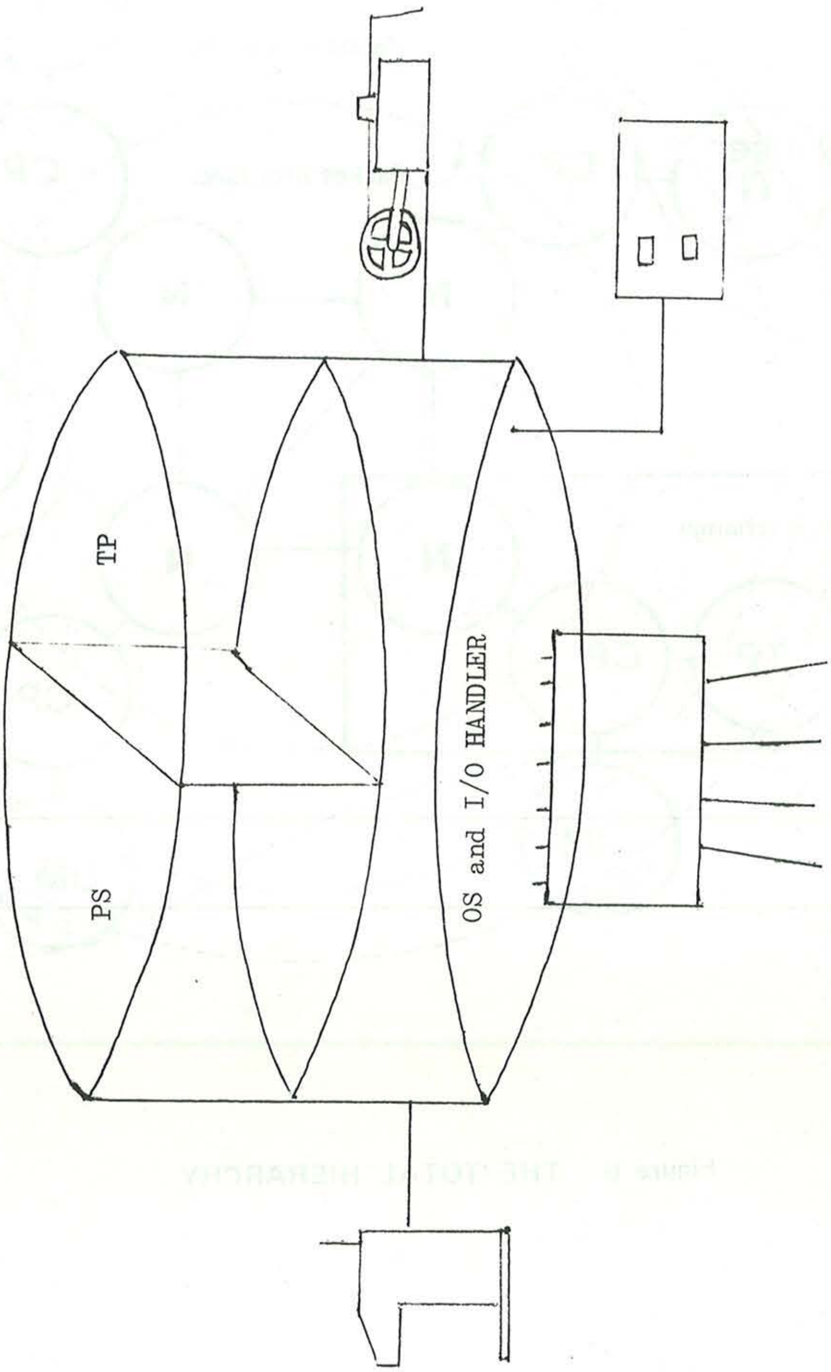

Figure 7 DDP 516 MESSAGE SWITCHING COMPUTER SOFTWARE STRUCTURE

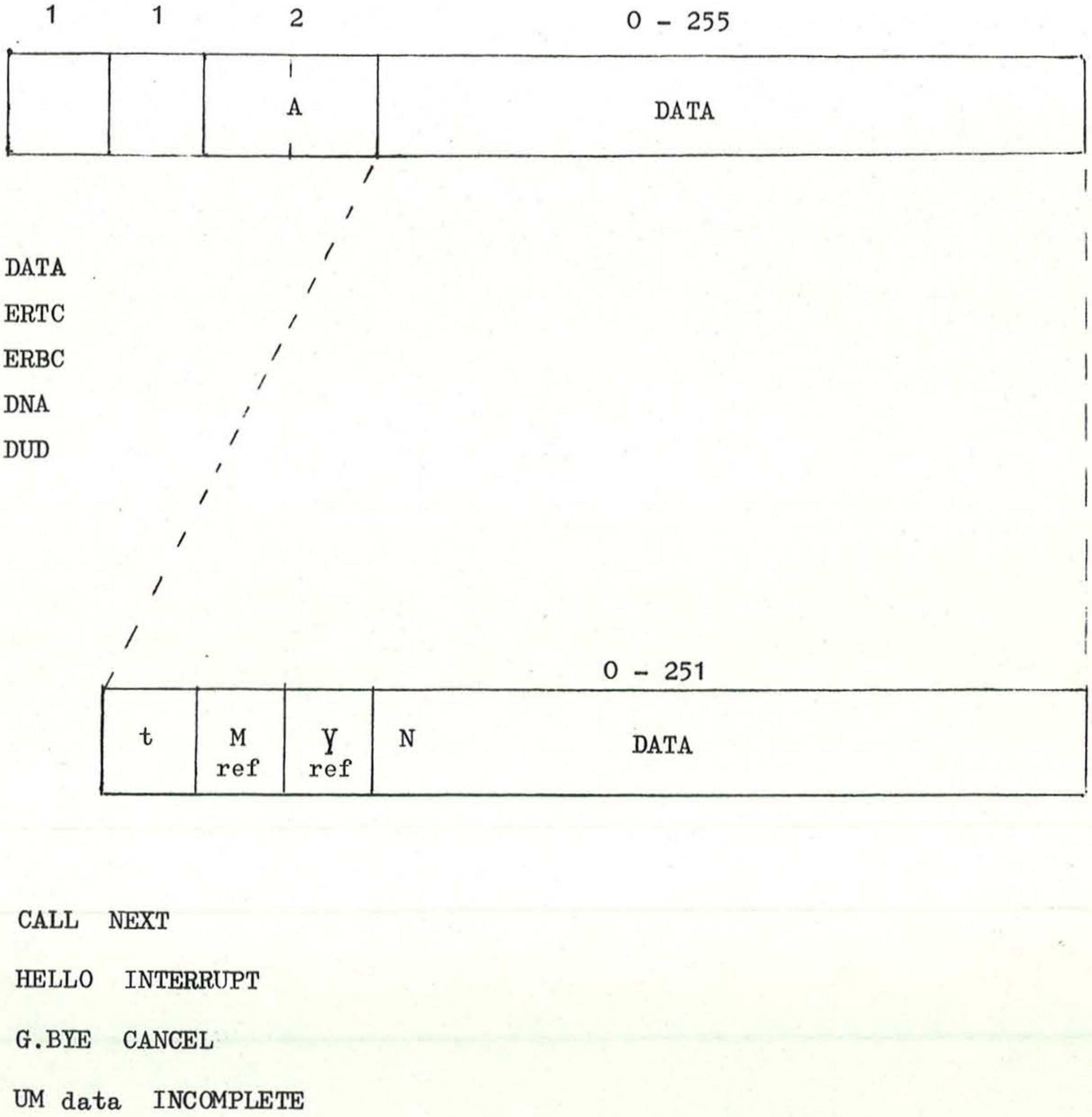

Figure 8 PROTOCOL USED IN NPL PACKET SWITCHING NETWORK