REVIEW OF COMPUTER NETWORK TECHNIQUES: PRESENT AND FUTURE

D.B. McKay

Rapporteurs: Dr. T. Anderson
Mr. R.M. Evans
Dr. L. Waller

Abstract A review of the techniques and terminology used in the field of computers in communications, with particular emphasis on computer networks, is followed by a presentation of some research activity at Yorktown (in conjunction with ARPA network personnel) concerning two specific networks together with their operation protocols, and an examination of the problems arising in the design of a network machine.

Review of Telecommunications' Techniques

A basic component of any communications system is the communication link, of which there are a variety of types. On a simplex line, information traffic is in one direction only, perhaps from a data collection terminal. These lines are not very common. A half duplex line can carry data in either direction, but not both ways at the same time. All current IBM equipment uses half duplex communication lines. Full duplex lines permit two-way communication simultaneously.

In addition to these three modes of operation, there are many different speeds at which lines can transfer data. For instance, voice grade lines run at 2000 bits/sec. and are the principle lines in A.T. & T!s telephone network. Sub-voice lines, which may run at 150 bits/sec., are primarily used for interactive terminals; since terminals only generate data at about this rate, the line is used efficiently and therefore economically. A high speed line, often needed for analogue data, operates at 50k bits/sec. and is used whenever high data rates are required, as in the ARPA network.

Whether to have a dedicated, leased line or a dial-up, switched line depends on how heavily it is to be used. Roughly if a line is in use for more than 20 hours a month, it is cheaper to have a leased line, to increase band width and reduce errors. Against this, a switched line has the obvious advantage of flexibility.

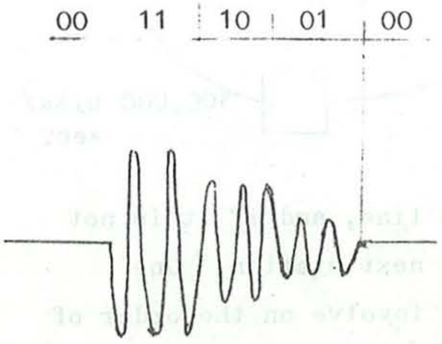
There are two different modes of transmission along a line, namely synchronous and asynchronous transmission. To achieve synchronous transmisssion both ends of the line must be synchronized and the data transmitted at a synchronous rate. Modems provide the synchronization of the data at both bit and character level. Large data transfers, such as files or from a remote job entry station, are commonly made over synchronous lines. Data concentrators are often used to combine a number of voice grade lines to obtain more efficient utilization of a synchronous line. This ensures a steady stream of data bits.

Asynchronous transmission involves preceding each data item by a pre-assigned 'start character', and following the item by an 'end character'. This is well suited to use by non-buffered terminals. The advantages and disadvantages of the two modes are summarized below.

	Asynchronous (Start-Stop)	Synchronous
Advantages	Little, if any data lost through lack of synchronization as each character is individually synchronized.	Good ratio of data to control bits (low redundancy).
Disadvantages	High rate of control information to data information (high redundancy).	Much data can be lost between synch pattern if devices become 'unsynched'.

Increasing use of more sophisticated terminals, which need much higher data rates, is leading to a corresponding decrease in the use of asynchronous lines.

The three types of modulation most frequently applied by modems ('modem' abbreviates modulation/demodulation) are frequency, phase and amplitude modulation of the signal. Modulation is needed to make effective use of the analogue communications network - in the U.S.A. this is A.T. & T.'s system of voice lines. There is a trade-off; more economic use of a line requires more densely packed data, but this incurs increased costs in encoding and decoding the data. The following diagram shows how amplitude modulation can be used to give four distinct levels of signal, and thus encode two bits within each information frame. Note that the word 'baud' refers to an information frame.

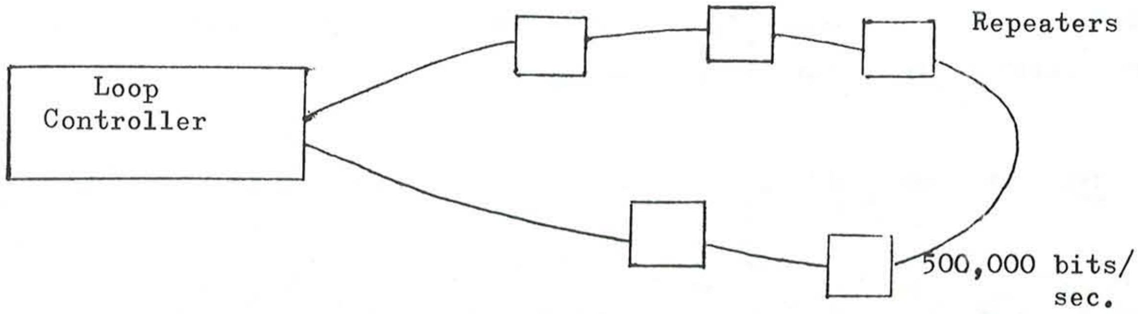


In this case, 1 baud/sec represents 2 bits/sec.

Error detection in the data communications area is the subject of many books, and from the viewpoint of university curricula, could provide a course in itself. To obtain increased effectiveness in detecting errors it is necessary to pay a cost in terms of diminished throughput of the original data along the line. Parity checking on each character transmitted is widely used, but statistics have been published indicating that on a 1200 baud line, 49% of the errors occurred within seven bits of another error. In consequence, approximately 30% of the errors would not be detected by parity checking only on characters. An additional check can be made by counting the number of '1' bits in a message, and transmitting this count after each message.

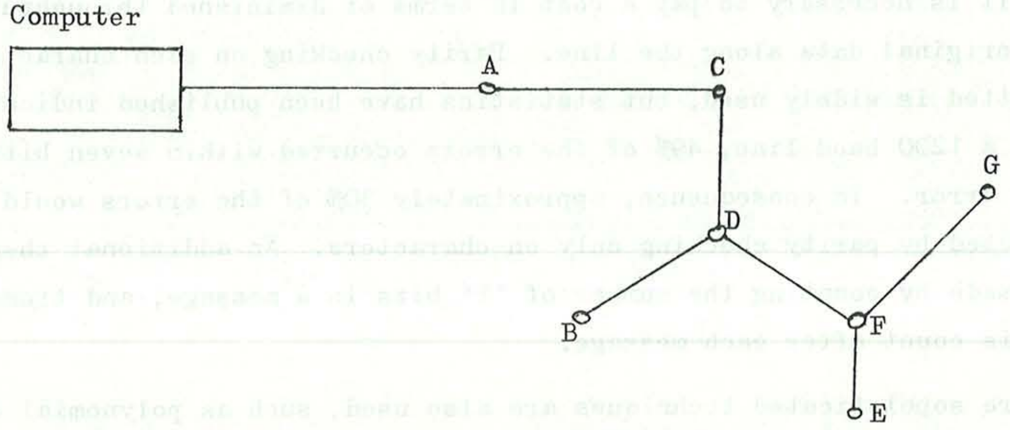
More sophisticated techniques are also used, such as polynomial codes, whereby a bit string is mapped to a polynomial, and the remainder under division by some fixed polynomial is used as a checking mechanism.

Communications networks have been constructed in a variety of configurations. The next diagram shows a single line, operating at high speed, connecting a number of nodes, one of which is designated as the loop controller.

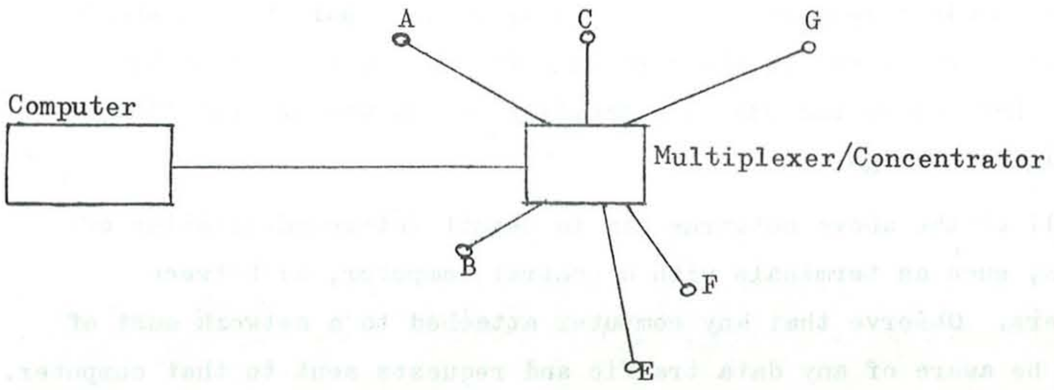


Each repeater station examines the data on the line, and if it is not intended for that station, passes it on to the next station. One application of the above type of network might involve on the order of 100 badge reader stations, usually within a single local plant.

Another frequently used network configuration is known as a multidrop/multipoint network. Both features are shown in the diagram below, multipoint control being necessary at nodes D and F.



One important special case makes use of multiplexor or concentrator to combine the data signals from each node before transmission to the central computer.



A data concentrator performs a more complex function than a multiplexor, in that a concentrator can restructure and buffer messages from its inputs before sending them out, usually along a higher speed synchronous line.

Contention for a line occurs when two or more systems connected to the line wish to send data simultaneously. Some kind of protocol is needed to resolve contention for a line; for instance a priority scheme between the systems could be used. An alternative is to use roll call polling, which gives fast response to demands and does not require very much logic circuitry. As the name indicates, the method involves polling each system in the network in turn to determine whether the system is ready to send data.

Message switching systems are another way of obtaining better line utilization, so as to reduce costs. The penalty incurred is that the delay time for responses can vary. This could cause problems for some real time applications. The list below shows some of the functions a message switching system performs.

1. Accepts messages from terminals.
2. Determines determinations.
3. Checks priority.
4. Processes the message, and collects statistics.
5. Provides error detection.

It has been found that the implementation of a logical, end to end, acknowledgement (in addition to the physical acknowledgement between two intermediate message stations) is of great importance in keeping track of messages within the network. The ARPA message switching network has had to include this facility, which was not provided by the original design.

All of the above networks aim to permit intercommunication of systems, such as terminals with a central computer, or between computers. Observe that any computer attached to a network must of course be aware of any data traffic and requests sent to that computer. The requisite interaction with the network can be achieved by means of a 'communications control program', the basic functions of which are exemplified in Figures 1, 2 and 3, which briefly document a communications control program for an IBM System 3/10.

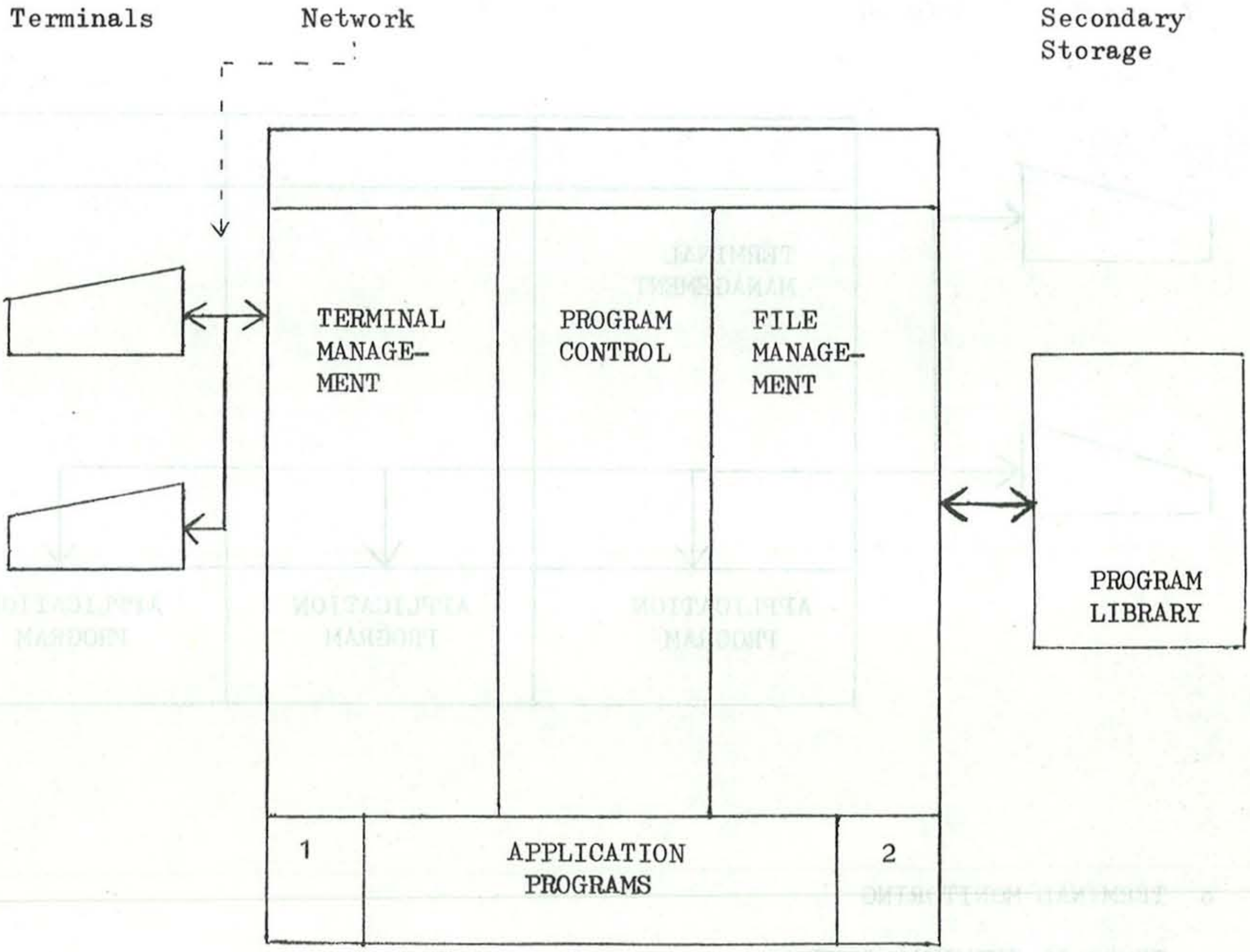
A communications subsystem is described to show the various components of a computer switching system within a network. It was built for a centralized computer network which will be described in more detail subsequently. This network was a star configuration of systems as shown in Figure 4.

The centrally located subsystem was designed to handle very large amounts of traffic. The line handler interfaced, via IBM's basic telecommunications access method (BTAM), to synchronous voice grade lines (of differing speeds). Error recovery over and above that supplied by BTAM was built into the system. The queues were managed using a round-robin dispatching algorithm on the blocks of the messages, thus ensuring that a single large message did not hold up the other short messages in the system.

This review has tried to summarize, or at least indicate, the many decisions and problems which have to be resolved when setting up a communications network. The speaker declared his prejudice for regarding network problems as extensions of computer problems.

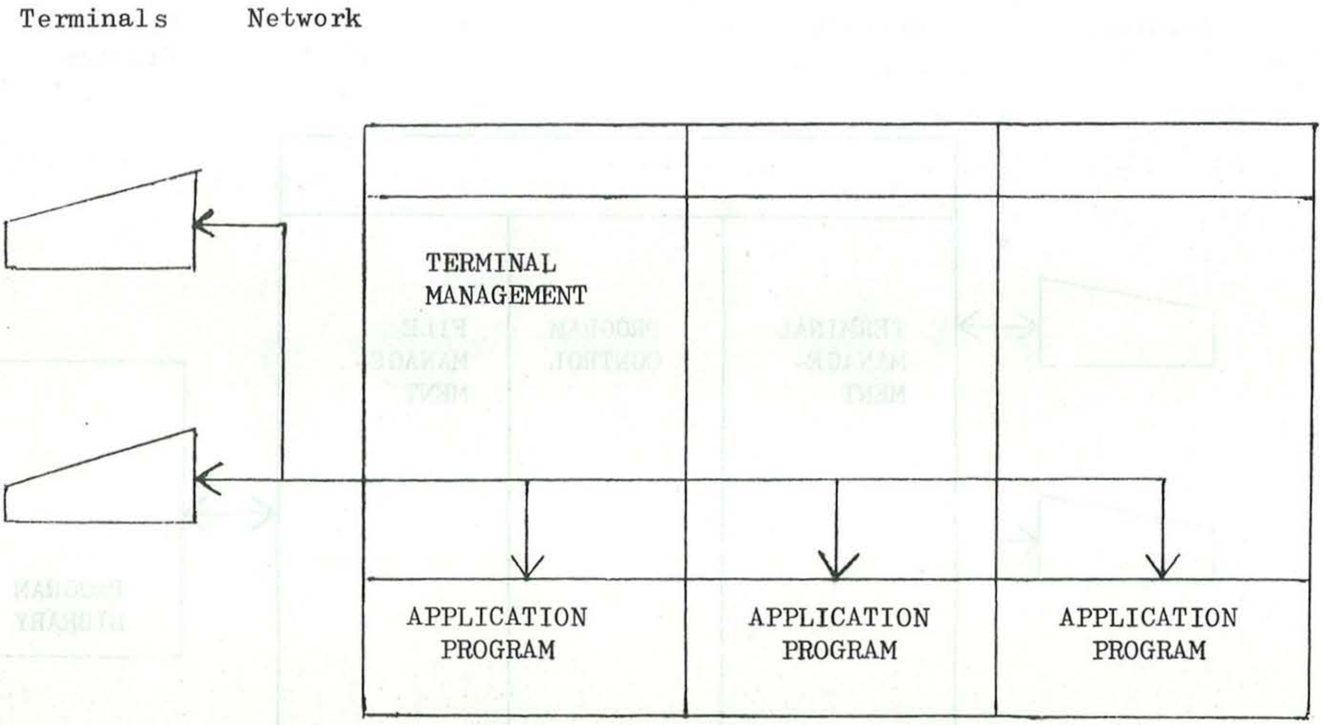
Professor Randell then asked for a quantification of the work involved in producing, and the amount of code produced, for the communications subsystem described above. Dr. McKay replied 'Three to four man-years of effort' and '50K bytes of storage'.

COMMUNICATION BASED SYSTEM



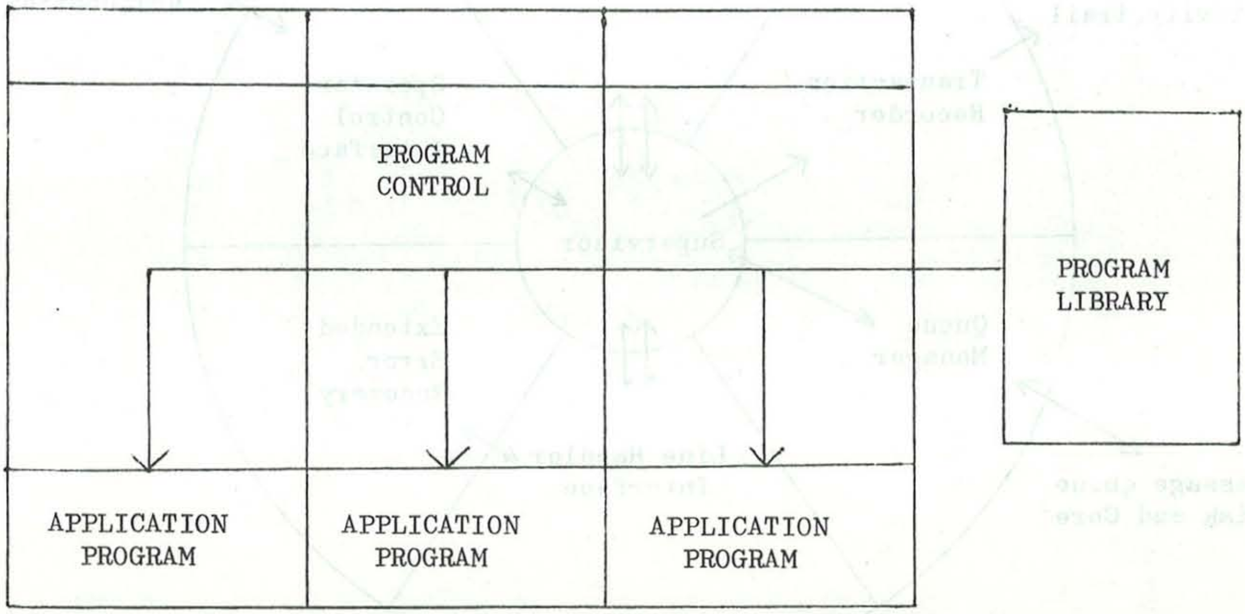
- o TERMINAL MONITORING
- o PROGRAM FETCH ON TERMINAL OPERATOR REQUEST
- o STORAGE MANAGEMENT
- o CONCURRENT PROGRAM EXECUTION

Figure 1 Communications control program



- o TERMINAL MONITORING
- o SYMBOLIC TERMINAL CONTROL
- o DATA TRANSFER AND TRANSLATION
- o OPERATOR FACILITIES

Figure 2 Communications management



- STORAGE ALLOCATION
- APPLICATION PROGRAM FETCH
- MULTIPROGRAM CONTROL
- APPLICATION PROGRAM PURGE

Figure 3 Program management

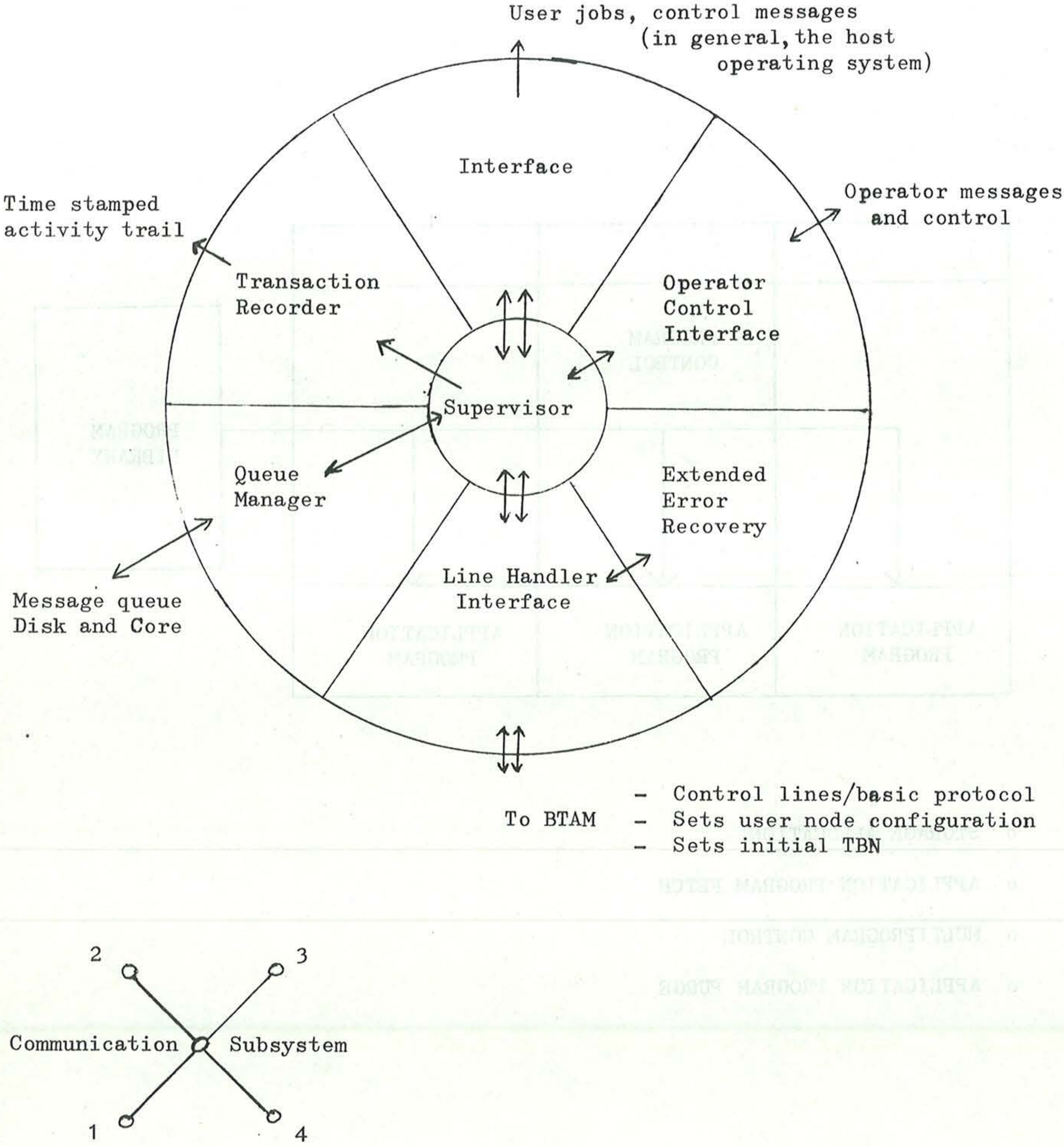


Figure 4 Communication subsystem

Dr. Fraser asked what the bandwidth of the system was. Dr. McKay pointed out that this depended on the number of lines being handled, which in turn was limited by the host computer. The initial implementation on an IBM 360/50 supported 3 x 50K baud lines and roughly 32 voice grade lines; transfer to a 360/91 increased the possible bandwidth considerably.

Dr. Wheeler wondered if any statistics by types of error had been collected. Dr. Fraser answered that among others, the ARPA network and A.T. & T. had assembled a lot of data of this nature.

Networks and their Operational Protocols

The way in which networks operate, achieve their interconnection and establish the ability of users to communicate with other systems in the network is to be described. In particular detailed descriptions are to be given of two networks, the ARPA network and the network 440 with which the speaker was involved.

ARPA originally started up with 16 host systems in two accumulations, one of which was on the west coast, the other on the east. In Figure 5 the nodes represent the interconnections of the IMPs. An IMP is a front end to the host and has at least two 50 kilobaud lines coming out of it. An IMP has an interconnection facility of up to 7 lines and a substantial amount of analysis has been performed upon the topology of the interconnection of the IMPs. ARPA now has upwards of 35 sites including a satellite link to Europe.

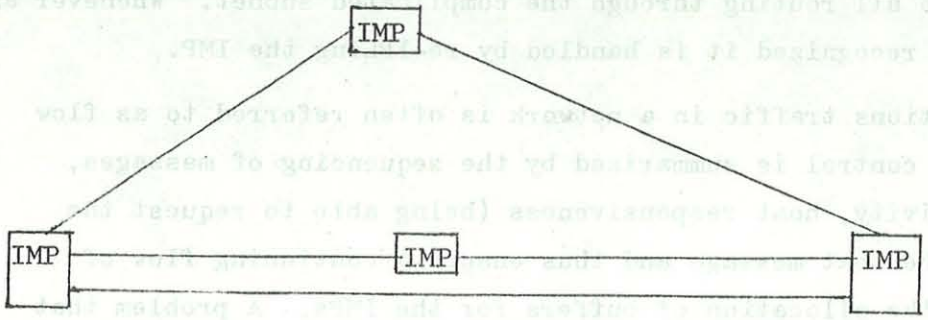


Figure 5 Interconnection between IMP's

The system was conceived in 1968 and three different levels of protocol were designed. These protocols, the subnet protocol, the host to host protocol, and the users' protocol, are independent in that each has a dedicated purpose.

The subnet protocol comprises the intercommunications between the IMPS, that is the communications subnet of the ARPA network. A primary consideration was the reliable transfer of bits - it was felt that this could be done better than by going through the dial network of A.T. & T. The average transit time should be less than half a second because interactive computers being used in an interactive environment were involved. The subnet-operation should be completely autonomous. (The speaker felt, however, that this was a serious design deficiency in that a lot of information could be gained from knowing what was happening in the communications net; such information could be valuable to the mode of operation of a host in that the host might wish to purge or bring some new function into play.) Another objective was that the host-to-host protocol should be separate from the subnet design.

The communications subnet protocol is seen in Figure 6 as a packet-switching transfer of data between the IMPs. A logical message consists of 8 units of transfer (or packets). When a host has a logical message to send it transfers 8 packets of information into the service IMP. The IMP breaks the message into packets and starts sending them. Each packet received by a receiving IMP is acknowledged and when all packets have been reassembled by the receiving host a RFNM (request for next message) is sent back to the original host, indicating that another logical message can be sent. Primarily the communications subnet handles interrupts and resets buffers in the IMPs for the host channels. The IMPs handle all routing through the complicated subnet. Whenever an IMP failure is recognized it is handled by re-IPLing the IMP.

Communications traffic in a network is often referred to as flow control. Flow control is summarized by the sequencing of messages, subnet connectivity, host responsiveness (being able to request the host to send the next message and thus ensure a continuing flow of traffic), and the allocation of buffers for the IMPs. A problem that occurred early on in the ARPA network was that if for some reason a logical message does not completely arrive at the destination IMP, then one buffer can get partly filled with the message and buffers can become

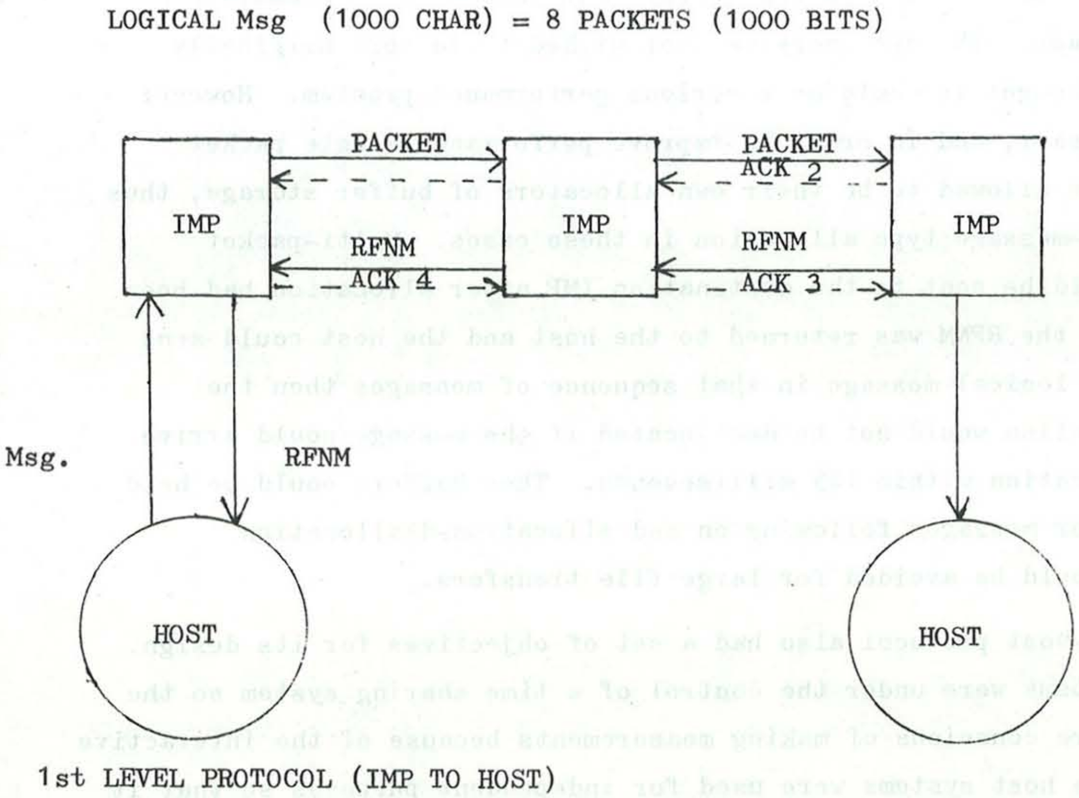


Figure 6 Communications Subnet

tied up. Consequently the IMPs do not release their buffers; this can happen at various points in the network and other messages can be held up because of the congestion, thus the network can grind to a halt. To counter this problem the following action was taken: buffer storage for a message was allocated for its transfer through the network; IMP 1 would request IMP 2 to allocate enough storage within IMP 2 to accumulate a logical message. The designers had not wished to do this originally since they thought it would be a serious performance problem. However it was necessary, and in order to improve performance single packet messages were allowed to be their own allocators of buffer storage, thus avoiding two-message type allocation in these cases. Multi-packet messages would be sent to the destination IMP after allocation had been made, and if the RFNM was returned to the host and the host could send out the next logical message in that sequence of messages then the buffer allocation would not be deallocated if the message could arrive at the destination within 125 milliseconds. Thus buffers could be held in reserve for messages following on and allocation-deallocation procedures could be avoided for large file transfers.

Host-to-host protocol also had a set of objectives for its design. The host systems were under the control of a time sharing system so the designers were conscious of making measurements because of the interactive traffic. The host systems were used for independent purposes so that it was necessary to preserve independent administrative control. Since experienced programmers were using the network there should be no restrictions upon programming languages and the character set. The design should be open ended so that the protocol could be changed, and there should be a minimum effect upon the host operating systems. This second level protocol is achieved by the NCP, the Network Control Program. The NCP must establish connections, be able to break connections and control the flow. It does this by receiving messages, for example, that an IMP is ready for a new logical message. If a user makes a request to the user protocol then this request comes into the NCP and a request for connection is sent out together with a connection number (sent number). This is received by a logger (a task running on the NCP on each of the servo systems), and assigned a received number in conjunction with the sent number. Thus two addresses are established and hence the logical link between the two systems is achieved.

The user's protocol utilizes a TELNET which will log a user onto a site of his choosing. It generates a proper set of control characters so that the system onto which the user is logging thinks it is dealing with a local system. Thus people (throughout the network) with different types of hardware terminals interconnected into their own hosts or TIPs (terminal IMPs) can establish connections to other systems. Participation in other systems can thus be achieved in a flexible manner. The TIPs can support 64 low speed (2400 baud) lines and have dial-up facilities through which access can be made to any computer in the network.

The ARPA community decided to add to their network an Illiac - a big array machine - and also a PDP 10 which is going to be solely dedicated as a data computer in that it will accept a data description language and manage a trillion bits of storage. This large store is to be used partly for a climate study. Ten years of weather data has been accumulated - enough to fill up one-tenth of the store - and weather prediction studies are to be based upon it. At Santa Barbara a "simple minded file system" has been developed and it allows 2314 disc storage online to a 360/75 to be used as a temporary hold facility for bit streams from users. A file transfer protocol, allowing the creation, opening, copying and deletion of files, has also been set up. The ARPA community is also trying two other experiments. Their graphic users are attempting to establish a standard protocol for transferring graphic data between various sites. Data conversion, or data reconfiguration is also of interest. By this technique the bit representation of data may be changed from one machine to another. This conversion, performed on an IMP or a TIP, operates upon the data as a bit stream and requires only parameters from the user in order to specify the format of the input and output streams.

At Yorktown it was felt that netting would permit the use of shared programs and data; existing facilities and software ought to be utilized and all the functions of a network ought to be available to a user in a transparent manner. Furthermore such a system ought to be general purpose, and a batch type orientation on top of an interactive capability was envisaged. The network 440 was set up as in Figure 7. The central computer was required to perform two functions - act as a communications subsystem and also as a network controller to coordinate the activity from users. The user's system communication interface is analogous to the NCP on the Arpanet, but it has another function in that if dynamic storage is available it could be allocated dynamically to the user's

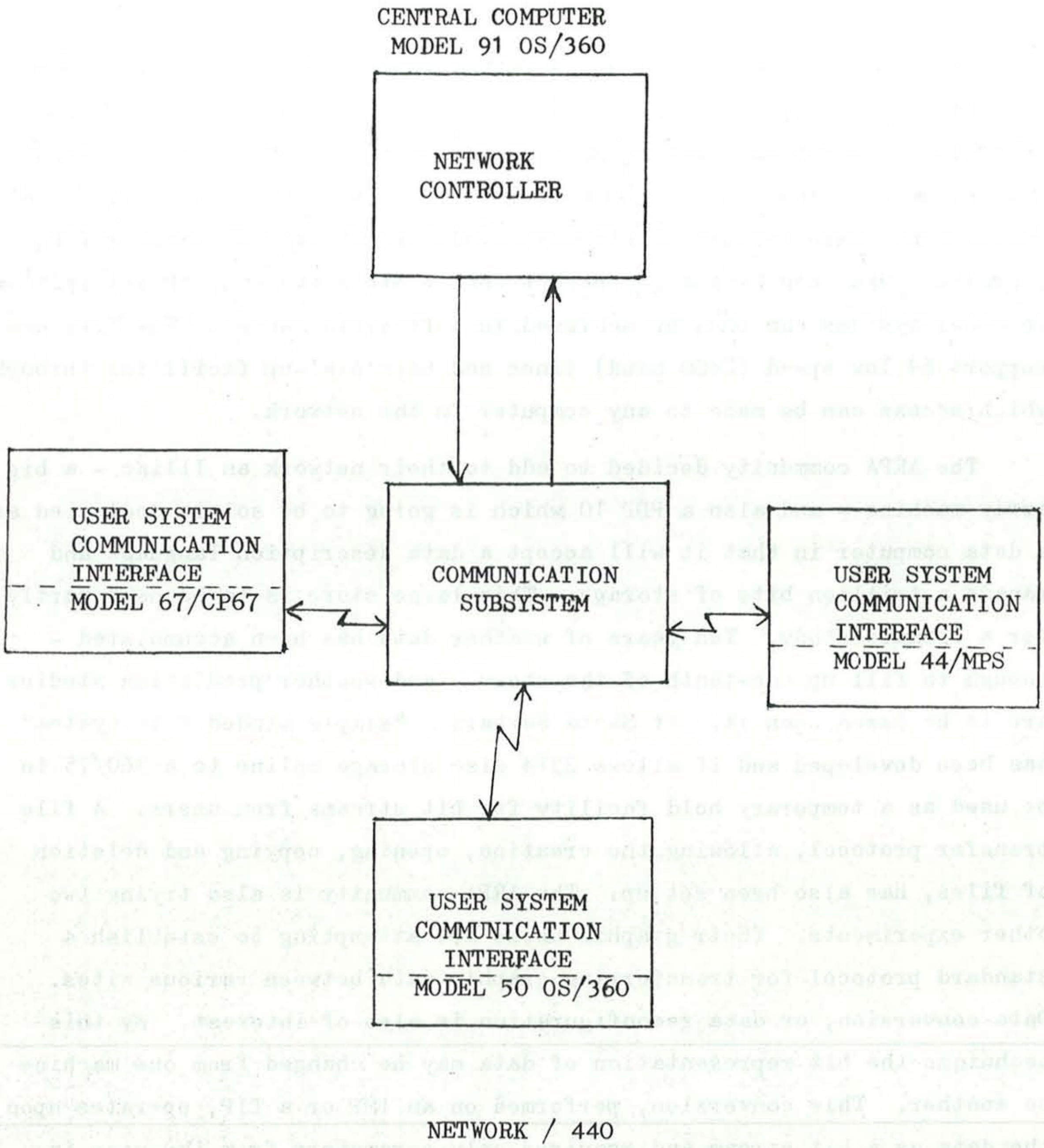


Figure 7 **Integrated computer network in star configuration**

interface. Initially the network control subsystem was run on the model 50 but it was later moved to a dedicated 300K region on the model 91 at Yorktown.

An objective that the designers kept in mind was that the network was really a multiprocessing system and should appear that way to the user. They were primarily interested in establishing a sequence of programming events for controlling communications among network users. The network controller was to set up the users' network jobs (or sessions). The user system interface allows communication between user systems and central systems. The communications subsystem - very similar to the first level protocol in the ARPA network - allows switching and provides temporary storage for messages.

The whole concept of the network 440 was that users would wish to specify a set of commands (or functions) for the network to perform. As an example consider the situation depicted by Figure 8.

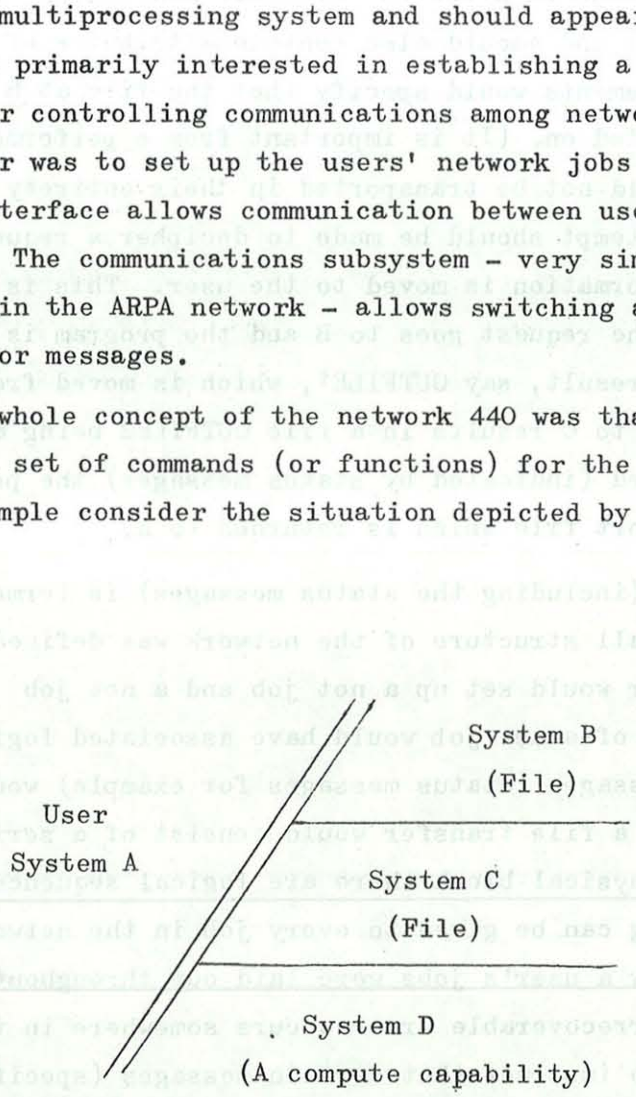


Figure 8 Illustration of the use of facilities of different systems

The user at A wishes to take information from the files in systems B and C, have it processed by a program in system D, resulting in a report which is to be returned to A. In the network 440 this could be don with a network job using definition statements and command statements. The definition statements allow the symbolic addressing of the various nodes (A,B,C etc.) and should also contain attributes of the file(s). The command statements would specify that the file at B should be moved to D and be operated on. (It is important from a performance standpoint that files should not be transported in their entirety through the systems - rather an attempt should be made to decipher a request so that only the relevant information is moved to the user. This is however a non-trivial problem.) The request goes to B and the program is executed against the file giving a result, say OUTFILE1, which is moved from B to D. The same procedure applied to C results in a file OUTFILE2 being moved to D. When all the files are moved (indicated by status messages) the program at D is executed giving a report file which is returned to A.

Each of the messages (including the status messages) is termed a logical message. The overall structure of the network was defined so that the network controller would set up a net job and a net job identification. Each step of a net job would have associated logical messages. Most logical messages (status messages for example) would consist of a single block, whereas a file transfer would consist of a series of blocks; so that within a physical block there are logical sequence numbers. Hence a logical structuring can be given to every job in the network, and a map exists of exactly how a user's jobs were laid out throughout the entire network. If some irrecoverable error occurs somewhere in the network then it is possible to instruct that certain messages (specified by their net job identifier) are to be purged; thus selective purging to any desired level can be performed.

In initialization, a network job establishes a connection to the system(s) in the network. The user systems have four possible responses to the initialization message. These are:

1) Ready to receive and transmit.
2) Cannot receive data but can receive control messages.
3) Physically connected but cannot receive traffic from the network.
4) Unable to handle new requests, but handling traffic for jobs in progress.

All systems have a resume message and because of the header mechanism and logical structure they can give a selective start-up to any job in the network. These were deliberately designed-in hooks in order to provide flexibility. As an example, suppose user node 1 is sending a logical message consisting of four blocks to user node 2 via node 3, and node 2 goes down. Node 3 may be in the process of sending blocks to node 2 through the grid node, in which case the grid node may reply 'I don't wish to take logical messages D from you' or may do nothing. In the first case node 3 would hold on to the messages pending (D2, D3, D4) whilst in the latter the grid node would accumulate D2, D3, D4 in its buffers and hold them until system 2 came back. There is thus a choice of whether to keep messages at the central site or at a user's site. In an unreliable network one might wish to move data as far and as quickly through the network as possible.

A Network Machine - its Architecture and Control

Introduction

This section is concerned with the design of a network machine, in particular with that type of network machine which has a separate Network Controller which interfaces with the user. Not all networks have a separate controller, for example in the ARPA network the user has to get right into the network itself. The network controller's function is to discover exactly what the user wishes to do and then to take advantage of all the potential resources of the network to accomplish the user's aims. Naturally there are many problems to be overcome in designing such a network controller.

When considering a network of any sort - in this case an actual network of machines with each site being a computer or system of computers - it is almost always true that the systems are dissimilar and autonomous, thus there are many problems in bringing the systems together. In particular there are value representation problems and problems associated with having different command languages. In designing a computer network it is necessary to sit down and attack these problems right from the start by answering questions such as 'How is the user going to do this?', 'Should this be transparent to the user?', 'How are we going to control this huge mass of resources?'.

Figure 9 is an architectural review of a network. This method of viewing a network is to see it as a combination of all the systems in the network; each system looks identical, that is, it doesn't necessarily have the same complement of functions as every other system but the identity of the system is the same.

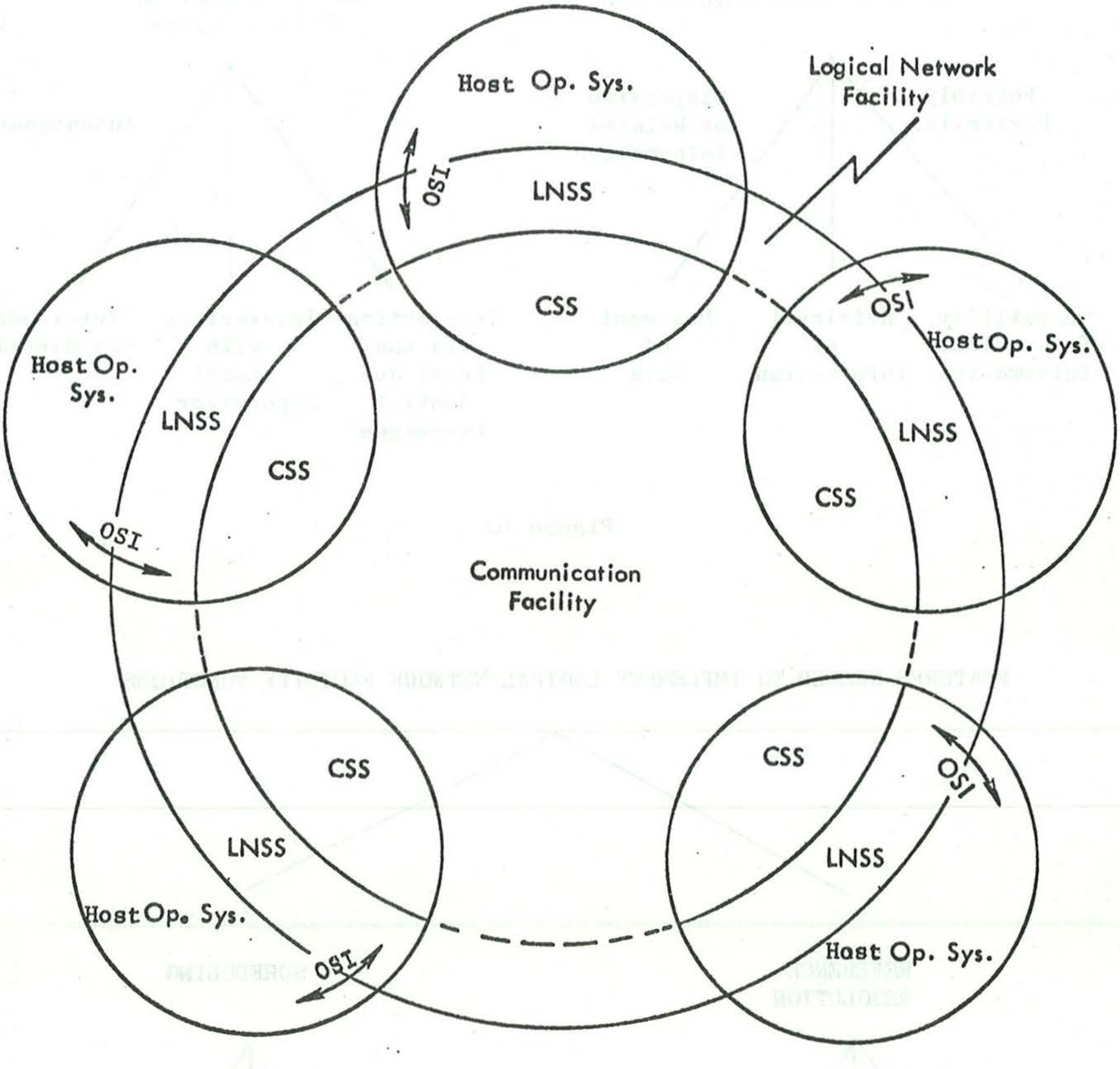
The Communications Subsystem can be any physical arrangement of the systems, for example, the ARPA communications subnet. The Logical Network Subsystem basically contains all the functions necessary to operate the network and co-ordinate the user jobs. By means of intercommunication the logical network subsystems configure themselves into a logically centralized multiprocessor system. The summation of all the individual logical network subsystems is referred to as the Logical Network Facility. It is this logical network facility and the functions it must contain that will be discussed here.

Logical network facility

The logical network facility has two basic functions. The first, and most important, is to interact with the user and carry out his specified tasks. Secondly, because it is made up of many logical network subsystems, it has to set up and co-ordinate the control of the various functions needed to carry out the users tasks.

It is important to realise that, whereas in network 440 control was exercised by a single computer, in this network control originates from many points in the network. Similarly, it is not necessary for the functions which support interaction with the user to be present on every system in the network; provided these functions are located on at least one of the systems, they can be dynamically brought in to support the users as the user population grows. Consequently the logical network facility must have the 'intelligence', and connections between the logical network subystems, such that it can direct the user to the system where the required function is located without the user being aware that any of this is happening.

Figure 10 gives a review of the types of activities that are going on in the network to support the user in carrying out his specified tasks. These activities fall into two distinct areas. Firstly, there is the collection and preparation of all information associated with a given task. This involves collecting together associated information from all



Network Machine

LNSS - Logical Network Subsystem

CSS - Communications subsystem

OSI - Operating system interface

Figure 9

LOGICAL NETWORK FACILITY FUNCTIONS

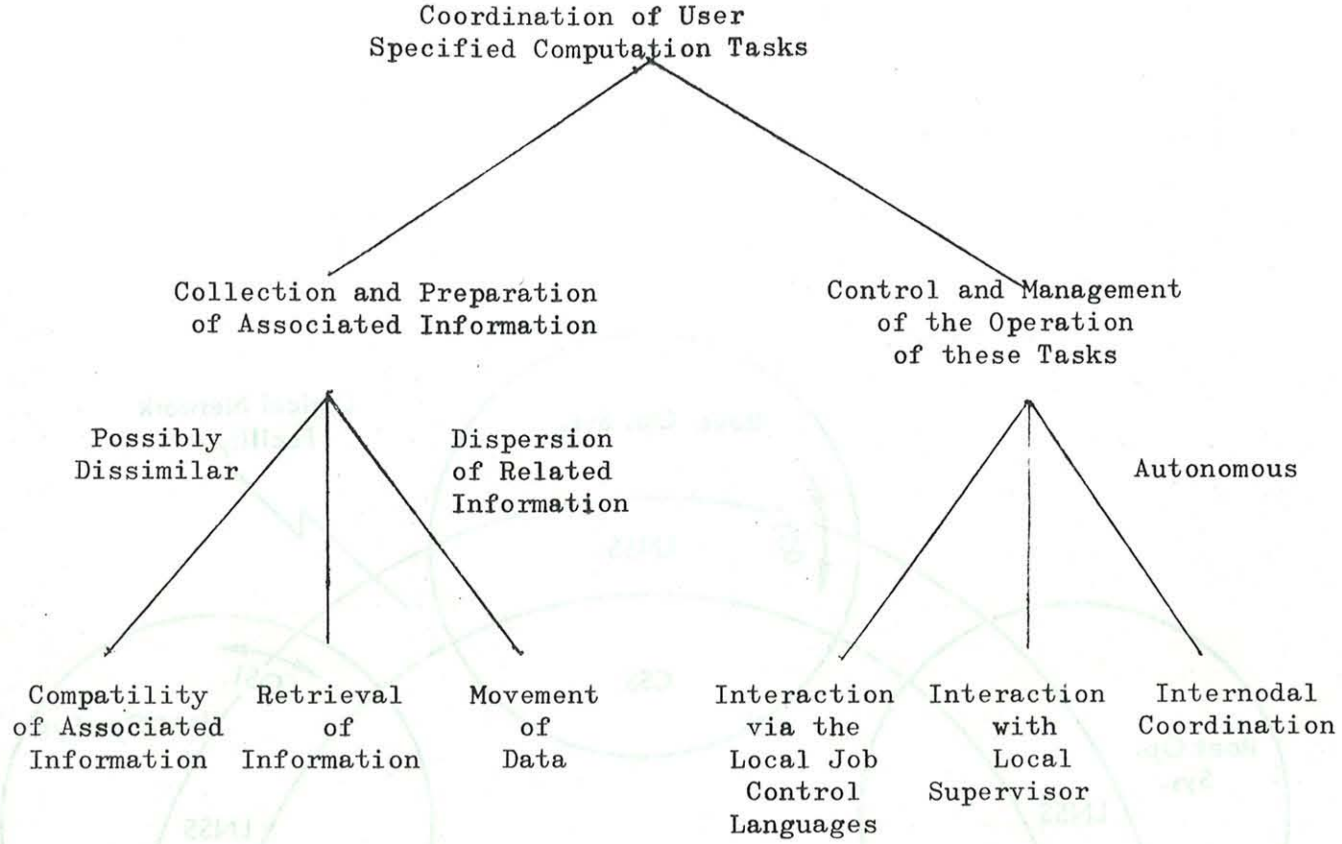


Figure 10

FEATURES NEEDED TO IMPLEMENT LOGICAL NETWORK FACILITY FUNCTIONS

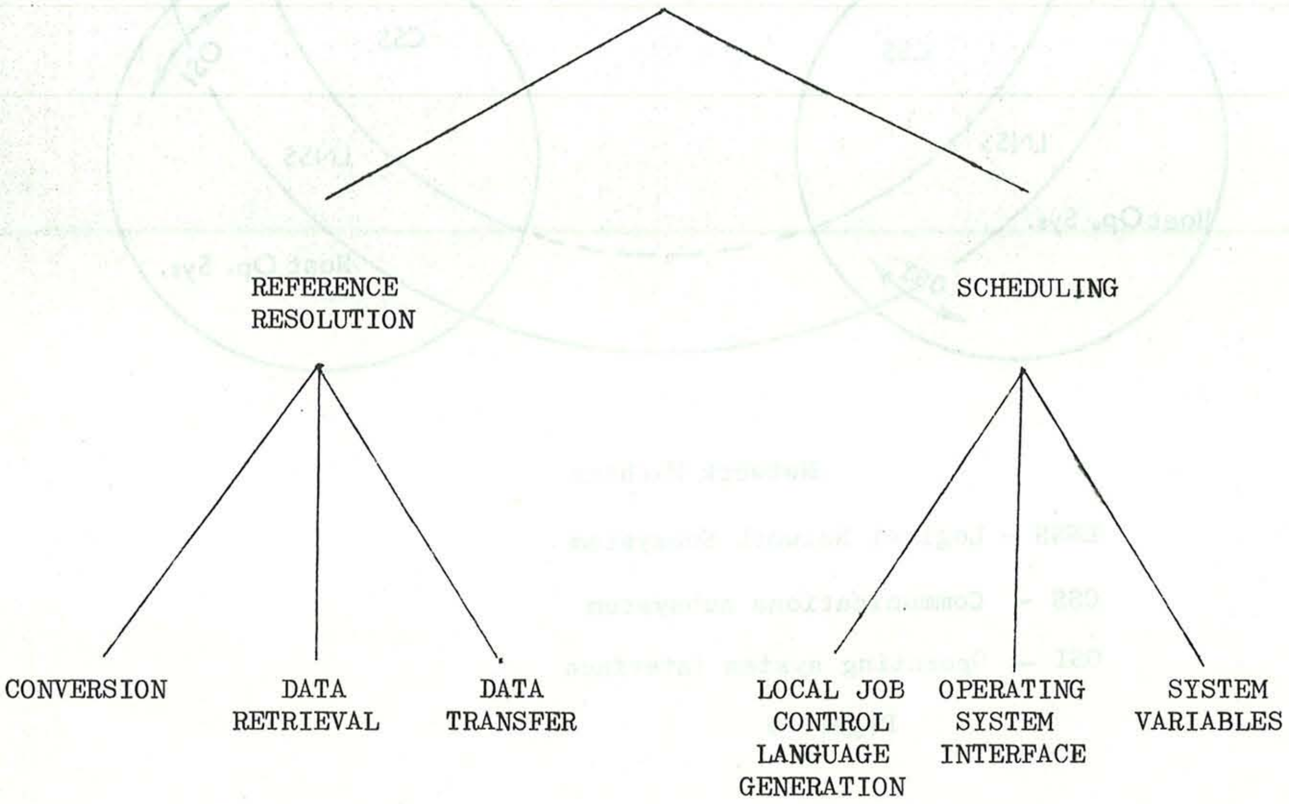


Figure 11

over the network, retrieving any other information that may be needed and physically moving data around the network. Secondly there is the control and management of the network operation to see that the specified task is carried out. This involves interacting, either through local JCL into a job stream, or through a local supervisor, with autonomous systems so as to take advantage of the resources of the various systems in the network. This co-ordination and control of tasks involves an operations protocol for obtaining use of functions on various systems.

Figure 11 indicates what is happening on each of these two support sides. Reference resolution is concerned with locating jobs, locating data, extracting pieces of data, getting data transferred, and conversion of the data to the correct format for the machine it is to be used on. This data conversion problem has been tackled in two different ways on present systems. The ARPA network has a data-reconfiguration service which takes an arbitrary input bit-stream together with input parameters describing how the bit-stream is broken down, and output parameters describing how the output bit-stream is to be made up. Network 440 performs conversions directly on various entities of data, for example, it can convert a floating point number in IBM 360 format to one in CDC 6600 format. The possibility of losing accuracy in the conversion process is tackled by making various alternatives available to the user.

On the control side the main problem is scheduling, which, in a network, can be a very interesting problem. If the user is given a degree of specification of what he wants to do, to the point where he can define the resources that must be available to him, then the scheduler can do an effective job of matching the environment the user wants for his job to one of the many environments available in the network. When a user enters a job into the network he does not want to become involved in the ambiguities and differences of local JCLs. What he really wants to be able to do is to specify in a general way what he wants done and then let the network controller generate the relevant local JCL commands, interface with the local operating systems and pass control blocks between the logical network subsystems.

<u>Network catalogue</u>

One of the main facilities that is needed in a network of this sort is a <u>Network Catalogue</u> to provide access to the constantly changing mass of information available in the network. Since this catalogue is to be stored in the network, and since network control is a function that has to be carried out by the network, just like any user function, it is logical to allow the network catalogue to be distributed throughout the network. Several different types of entry may occur in this catalogue.

The most obvious type of entry is the Data Name. This enables users to reference their data sets symbolically and not be concerned with the actual location of the data sets within the network.

A second type of entry is the Descriptor. This is a data set which can describe the logical configuration of the network, or the logical configuration of a particular data set. Thus a user will have several of these to describe his various data sets. Since descriptors are merely a special type of data set it is logical to name them, and hence allow them to be applied to any other data set of the same logical configuration.

A third type of entry is the Group Association. This is a type of assignment on top of the users data, which is concerned with the various group associations within the structures of his own data files. It can also deal with such things as authorization information, that is, who can access a particular file. Since the user community is operating in the network environment it is obvious that certain data sets will be made available for public use, and that other data sets will be made available to a restricted group of users. Hence a network user is allowed to give group assignments to networks and network data, thereby giving a global or subglobal access capability to that data. Storing this information in the network catalogue allows the data management facility (discussed later) to control this availability of data.

Also within the structure of the catalogue is a Network Catalogue Entry. Since the network catalogue is distributed around the network it is necessary to have some means of knowing where each piece of the catalogue is. It is the network catalogue entry in the network catalogue which manages this network catalogue data.

Network control language

To enable the user to initiate and control his tasks there has to be a means of communication between the user and the network controller. This communication is carried out by means of a Network Control Language. This language should be an interactive type of control language and not merely a specification language.

The network control language must have a data description capability, that is, there must be a data description language either as a separate function or included within the network control language. Also the user must be given some means by which he can describe, with sufficient information, the programs he wished to have executed. For example he may say 'Here is my Fortran program, execute it for me, and I don't care where you do it', or he may wish to restrict it to a certain class of machines because of some dependencies on a particular compiler or piece of hardware. This allows the network controller to pick any machine, or any machine of a particular type, according to some scheduling algorithm which attempts to best match the resources available to the users demands. Similarly the user should be able to initiate as many jobs as he wishes without having to wait for one to be completed before starting the next. It is then the network controllers responsibility to schedule these jobs on the various machines, to collect and hold the results, and, if requested, to notify the user upon completion of a specified job.

Logical network subsystem

Each logical network subsystem is made up of several distinct functions. The first of these is the Primary Controller which initiates the network jobs and is the first thing that the user comes into contact with when he attaches himself to the network.

Next there is the Data Management Facility which, basically, implements resource sharing. The most important aspect of resource sharing is data transfer. Consequently the manipulation and description of the data within the network is vital to enable the user, with the minimum effort on his part, to locate and extract the pieces of data that are of interest to him. As well as carrying out the users specified jobs, there will also be network control jobs to be executed. The functions necessary to carry out these network control jobs are catalogued and stored in the network, and the data management facility has to keep track of these functions as well as the functions directly related to the users job.

There is also a Delegated Control Capability. This is the logical interface which recognizes and accepts that control is being delegated either into or out of a particular node and manages this transfer of control.

To co-ordinate all these functions there is a Common Controller. As relevant information becomes available in the network it is broadcast to all these common controllers so that they can update their stored information. The other functions then reference the information in their local common controller whenever they wish to make decisions involving such things as scheduling or assignment.

An example

This example assumes the same conditions as that described in Figure 8, that is, four machines with two files, one program and the user all on different machines. The example is intended to illustrate the ease with which the user specifies what he wishes to do, and the fact that he is not required to know the location of the files or the program but merely their names, say FILEA, FILEB, RPT respectively.

The user informs the network controller that FILEA consists of ten records, each containing ten characters and that he wishes this format to be known by the descriptor DES1 by saying something like 'FILEA((10REC(10CHAR))N.DES1)'. He can then state that FILEB has the same structure by saying 'FILEB(DES1)'.

The network controller can discover the actual location of these two files by referencing the network catalogue. It is then the network controllers job, via the data management facility, to move these files or parts of files around the network as needed to carry out the users wishes.

The user could now decide to assign the name A to the sixth record of FILEA by saying 'A←FILEA [6]'. He can concatenate records by saying 'T←FILEB [7] || A' and so forth. In this manner he builds up his library of catalogued information about his data and what he wants to do with it, until finally he has described his data environment to the network.

Now to generate his desired report the user merely says 'R←RPT(T)', where he wishes the program RPT to operate upon the data T to generate his report which will be stored under the name R.

This type of functional capability is what the user really wants from the network. Although there is a lot of work and a lot of problems involved in providing this capability, it is all worthwhile if it produces a network that people will actually find useful and easy to use.

Proposals for a general data sharing capability on the ARPA network

One of the aims of the ARPA network is to provide a large storage capability on the ILLIAC, managed by a PDP10 data computer. Hence there is a large data resource in the ARPA network and users require techniques and protocols to enable them to take full advantage of this resource. A data management committee, set up by the speaker, came up with the following proposed stages of development in the ARPA network to provide these capabilities.

The first requirement is a Transmission Facility. At the present time if a user wants to move a file from one system to another he has to log on to the system on which the file is stored, request it to transfer the file and then wait while it does so. The transmission facility would recognize the users request for file transfer, wherever he is in the network, and then carry out that transfer for him. This transmission facility would initiate file transfer, handle all messages about the status of the file in transit and generally act as a buffer for the user.

Once this transmission facility is working the next step is to start collecting information about the names and locations of all the files in the network and to construct a Location Catalogue. This catalogue will enable users to refer to their files by name and not have to remember in which system and whereabouts in that system each of their files is at the present time.

The next stage is to construct a Description Catalogue. Within the data computer there is a language which allows users to structure and define the logical makeup of the data they are storing on the ILLIAC. This description information also has to be stored, so the idea is to name these files of description information, reference them by name and allow them to be applied to other data that is being stored on the data computer.

Stage four consists of providing Data Conversion Modules. The ARPA network already has an experimental data reconfiguration service which works on arbitrary bit streams with input and output parameters. Network 440 has a conversion module which allows machine representation conversion between IBM 360, CDC6600, B5500 and PDP10, and is designed to cope with six other systems as well. It would be interesting to place the Network 440 module in the ARPA network and compare the two different mechanisms side by side under the control of the data management facility. From this comparison could come the design for the best type of conversion module for the ARPA network.

At present whenever the user wishes to access a piece of data, whether it be systems data or user data, he first has to log on to the system containing that data and then make his request in the local command language. To remove this restriction a File Access Command Interface is needed. The user could then make his request from anywhere in the network in a generalized manner. The data management facility would take this request, construct the appropriate task on the system containing the data and return the data to the user.

The sixth stage is to provide a Data Access Module which is an extended version of the data description language already present in the data computer. This will provide a standard method of naming and referring to data anywhere in the network, thus providing the user with an additional functional capability. This module will use the file access command interface to translate the users request into the correct sequence of actions to recover the relevant data. Also within the data access module will be the capability of using descriptors, at present used only on the data computer, throughout the network. This will enable a user to structure all his files and give him the capability of extracting pieces of data from out of his files without retrieving the whole file.

The final stage is to build a Data Management Interface to allow all the individual local data management systems to communicate with each other. This will provide a general user control, or network control, capability in the network. Although this stage involves a considerable amount of experimentation it would appear to be a useful method of attempting to test out the ideas of the logical network facility described earlier.