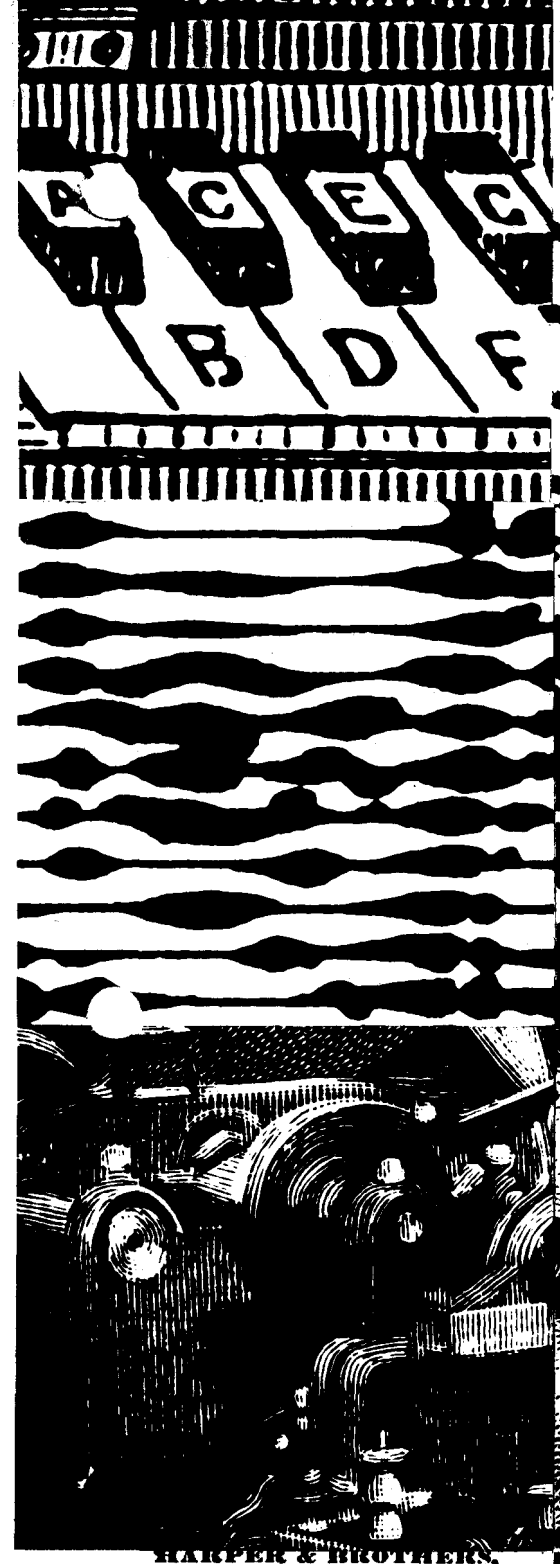
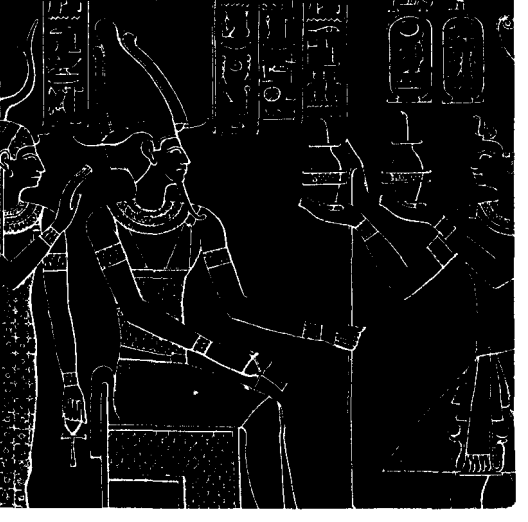


COMPUTER NETWORKS

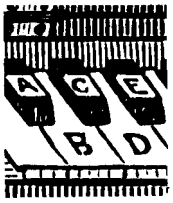
C. G. Bell
A. N. Habermann
J. McCredie
R. Rutledge
W. Wulf

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania



This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F 44620-67-C-0058) and is monitored by the Air Force Office of Scientific Research. This article is a condensed version of an article appearing in the Carnegie-Mellon University, Computer Science Review, 1969.

INTRODUCTION



Computer networks have the ability to bring the power of large machines to work on a single problem and to provide reliable computer services to large populations. They also may become an unmanageable structure that can cripple itself in a fashion akin to the great Northeast power failure in 1965. Imagine the following sequence: computer X does not have the sine subprogram but relies on computer Y for it; computer Y on the other hand solves the sine subprogram using the cosine subprogram which it doesn't have; computer Y therefore calls X for a cosine; X solves for cosine using sine which it asks Y for. . . . Of course, you say, no computer network would be so *simplistic*. But would you guarantee it could never happen for any set of computer resources among *N* computers—and that the network might not head for the buried recursive disaster like a lemming for a cliff?

Computer network research is just being defined as a separate area of computer science research. It overlaps the areas of operations research, operating systems, programming languages, and hardware systems. The commitment to network research is to both engineering and science. In the short term the network group at CMU intends to build a network, and from it gain direct experience of network operation. At the same time, we intend gradually to gain understanding which will allow us to build better computer structures—presumably in the form of networks. Thus network research is immediate (by building and operating a network) and on-going (by trying to achieve understanding).

It is natural to define a computer network to mean any configuration of two or more computers which communicate "meaningful" information to one another. This definition admits too broad a range of structures, both in the method of interconnection and in the nature of the information communicated. We will use "multiprocessor" for several processors sharing a common primary memory, and thus being extremely closely coupled. We will use "network" for the case of computers communicating by more indirect means, e.g., telephone lines, higher data rate links or even shared disks.

There are many reasons for an interest in computer networks; several of the more important are discussed in the first section. The CMU network group has focused on the rather broad goal of investigating network structures as an alternative to single computers for providing local computer power. This is only one of the ways networks can and will be used, but it is an extremely interesting one in terms of cost-effectiveness, reliability, computational power, and increasing total system longevity through gradual evolution (which we later define as rejuvenation). Secondary objectives include examining the more classical problems (e.g., communications) of other computer networks. That is, since the CMU network will not be distributed over very large areas, the communications costs are not the overriding factor that they might otherwise be. The single physical cluster of components, coupled with close coordination of the development groups, should allow experimentation, however, and make the study of more general networks possible. In fact, we do not want to predicate a network on the rather poor communications links that are currently available—we will, if necessary make a case for improving the links.

We will first discuss the capabilities that networks propose to supply—the goals. The next section is devoted to the CMU network research, where we describe an initial, limited network. The final part deals with several research areas relevant to networks.

CAPABILITIES

In rationalizing computer networks—hence computer network research—we begin by giving those reasons for their existence that have the greatest long term effect, and then turn to reasons which might have immediate cost and performance payoff. In the long term, the concept of an information utility is appealing—perhaps also inevitable. An information utility is a network formed by users (at computer terminals), various computers, links among the computers, terminals, and users. This structure is tc be likened to present networks for distributing utilities (gas, electricity, etc.). The subscribers to the network are able to buy processing power, have information stored, and interact with other user (subscribers). In fact, we might expect to see different utilities depending on the community being served (e.g., the businesses, homes, government departments) and the function being carried out (e.g., making cash transactions, delivering the mail by printing letters at local sites, distributing news, doing library information retrieval, entertaining).

Such a utility network would hopefully allow load sharing by being able to pass information processing tasks to another machine either better qualified or in a better (momentary) position to handle the processing. (Computer utilities might take advantage of geographical time zones, just as electric utilities do, in order to supply peak noontime loads.)[1] We would expect various forms of load sharing to develop. The development will be limited by the costs of secondary memory, processing, and transshipment of information. With load sharing, one might transship programs or data, or both, to another computer for processing.

In the nearer future an existence of large specialized data bases will make program and file sharing desirable because of high shipping (communication) costs and redundant file storage costs. In this instance, we ship only a small amount of input data to a program and its data base. For example, we might expect to see specialized networks for libraries as a first step in information retrieval networks because of the undesirability of replicating information in multiple places. Indeed, computer networks may provide the only economical solution to the problem of maintaining and controlling the information in libraries. An-

[1] *This phenomena has already been observed by time-sharing service companies.*

other form of sharing within computer networks is the communication of results among its users. Just as a community of users within a single time-sharing system share programs, the network should enhance communications among all users of the network. Such a network would allow physically separated subcommunities engaged in a common research to communicate.

In the case of electricity distribution a network is organized, such that various parts of the network can be planned and constructed at different times. Construction of new sites is on the basis of growth, coupled with the availability of new equipment which reduces cost. We foresee having a structure which has this goal—we call this dynamic process *rejuvenation*. Depending on the cost of the new equipment, the cost to convert programs to the new equipment, and the attitudes of the user population, it may not be desirable to remove older equipment from service. Yet, because of load increases, newer facilities are needed. (Purchased second generation computer equipment costs less per operation than third generation equipment, at least for certain classes of problems.) In these cases, because the demand for service has increased, both the old and new equipment should remain in service. The operational policy is usually not to invest in software for the older system (because of the low payoff). However, the older equipment can still serve a useful function in the total network. In electric power networks older generators are usually kept for a long time and run to meet peak demands (daily or seasonal).To obtain new equipment just to cope with peak demands would oversupply the network and raise the cost of supply. A network should provide a framework for adding and removing equipment without subjecting users to the major transients associated with equipment configuration changes.

The last and most important reason for a network is to take advantage of functional specialization by matching the computer to the task. This appears to have immediate payoff, as we can easily identify possibilities for functional specialization and can indeed cite existing examples. There are at least two issues of specialization. First, is any computer available with the particular capabilities to solve a given problem (independent of cost)? Second, given a mix of problems, what is the right collection of computers such that this mix can be run to minimize the cost, turnaround, user convenience, etc.? An example of a highly specialized facility is the large array processor, ILLIAC IV, under construction at the University of Illinois. Others might be a computer with a large primary memory (e.g., 10^8 bits) or perhaps large virtually addressed primary memories (e.g., 10^{13} bits). Perhaps only a few such specialized facilities would exist in the country, and a network would make them available to a larger community than the immediate geographical area.

In Figure 1 we illustrate the case for functionally specialized computers by showing several tasks (in order of increased complexity), and plotting the cost to process the task. It shows what we would intuitively suspect—simple tasks cost less to do on simple computers. Also, the cost per task can be lowered for a task by building highly specialized hardware for that task.[2]

[2]*Almost as a separate issue, there are increasing numbers of small computers in most environments to meet reliability, response time, and interface constraints. Each of the small computers often requires access to a larger central facility; as such, a large collection of interconnected (computer) users are present and by our initial definition this structure is a network.*

The above instances of functional specialization are reasonably long term. Already computers are used as controls for card readers, line printers, and display scopes. In some instances the terminal computer processes the task directly (e.g., card-to-printer and simple Fortran jobs) instead of forwarding it to another computer. Display control computers are especially necessary and worthwhile because the added cost of a processor (above the memory cost) is small. Similarly, small computers are used to gather and pre-process data in real time experiments. The small computer stores the data on tape for later processing on a larger computer. With networks the small computer would send data to the large computer during the experiment; thus the experimenter can revise the experiment based on the analysis of results (using the larger computer).

A Simple Analytic Model Argument for Networks

A few simplified models of alternative system configurations will exemplify some of the more fundamental issues of network design. Two important criteria for information systems are response time and reliability. Response time is the elapsed time from submittal of a processing request until its completion. A good measure of reliability is the steady state probability that the system will not fail completely.

As a specific example consider the following highly simplified problem. To achieve an overall processing power, P, a system could have one computer (complete with all necessary input-output devices) with capacity P instructions per unit time, or two units each with capacity P/2 instructions per unit time. Furthermore, the two computers may be connected in a rather simple network, or be completely disjoint. Assume that service requests arrive at the system with an average rate of λ jobs per unit time for each configuration. Figure 2 illustrates these potential arrangements.

These three systems correspond to standard queuing models, having simple analytical expressions for steady state solution. It is necessary to assume (1) the arrival process is Poisson and (2) the service requests are random variables drawn from a negative exponential distribution. The first assumption means that if one labels the arrival times of jobs as $a_1, a_2, \ldots, a_n, \ldots$, then the interarrival times ($t_i = a_i - a_{i-1}$) are distributed exponentially. Then, if the interarrival times t_i have a mean value of $1/\lambda$, the following equation describes the input process.

$$(1)\quad \text{probability}\,(t_i \leq T) = \begin{cases} 1 - e^{-\lambda T} & T \geq 0 \\ 0 & T < 0 \end{cases} \quad \text{for all } i$$

The second assumption states that work requests (e.g., number of instructions) $s_1, s_2 \ldots s_n \ldots$ are exponential. For a mean of $1/\mu$ instruction per job the following equation describes the service process.

$$(2)\quad \text{probability}\,(s_i \leq S) = \begin{cases} 1 - e^{-\mu S} & S \geq 0 \\ 0 & S < 0 \end{cases} \quad \text{for all } i$$

The time request i spends in a computer is then s_i/P, where P is the processing rate of that computer. We shall examine some re-

sults for the systems of Figure 2 for which simple analytic solutions exist. Queuing theory tests present complex series solutions for more complex cases, consisting of an arbitrary number of processors.

For purposes of comparison suppose we are interested in the steady state average value of the response time, R, through the system (i.e., R = the elapsed time from job arrival to job completion; $\bar{R}$ is the average value of R). Steady state solutions exist only for systems where the average processing time is less than the average interarrival time ($1/\mu P < 1/\lambda$ or alternatively $\lambda < \mu P$). Otherwise there simply is not enough total processing time to do all the jobs. The following table displays the results under the above assumptions.

System	$\bar{R}$ = *Average value of R*
A) Single Computer	$\bar{R}_A = 1/(\mu P - \lambda)$
B) 2 Connected Computers	$\bar{R}_B = 2[1/(\mu P - \lambda)] \cdot [\mu P/(\mu + \lambda)]$
C) 2 Disjoint Computers	$\bar{R}_C = 2/(\mu P - \lambda)$

System C has an average response time exactly twice that of system A, independent of the system load. ($\bar{R}_C/\bar{R}_A = 2$) The comparison with system B is not quite so simple. The ratio $\bar{R}_B/\bar{R}_A$ has the following value:

$$(3)\quad \bar{R}_B / \bar{R}_A = 2\mu P/(\mu P + \lambda) > 1$$

The inequality follows from the fact that $\lambda < \mu P$ for a steady state to exist. Figure 3 displays this ratio of average response times for various system loadings. As λ, the arrival rate, approaches μP, the processing rate, the systems become saturated. Note that in the normal design range ($.7 < \lambda/\mu P < .9$) the differences in average response times are quite small.

Let probability (fail (i, T)) denote the probability that system i will fail completely during time T. Assume that the probability of an individual computer failing during T is independent of its capacity (P or P/2) and equal to f(T). The reliability (1-probability [fail (i, T)]) of the three systems is presented in the following table:

System	*Reliability*	
A	1 - f(T)	
B	1 - f(T)•f(T) -	assumes system is operational if either computer is operational
C	1 - f(T)•f(T) -	assumes jobs can be moved among the two computers. Otherwise the reliability is worse, namely, 1 - 2f(T).

Systems B and C achieve a higher reliability measure ($f(T) < 1$ and therefore for n computers, $f(T)^n < f(T)$) because a failure of one of the processors does not cause the entire system to crash. A general characteristic of many networks is that one node may fail without impairing the total system. Obviously this "fail soft" characteristic must be carefully designed into the system to avoid massive failures. Note that although the overall system reliability is the same for systems B and C, a failure in one processor of system C is a total failure for that subpopulation it serves. Only a failure in the switch of system B could cause the total system to fail, and these elements may be designed with higher reliability than larger general purpose computers.

Figure 4 presents a different aspect of reliability and power by comparing a single, large computer with a two computer network, each of which is one-half the size. In this comparison, we assume the probability of failure for the large and small computers are f(T) and f(T)/2, respectively. The probability of obtaining P units of power in both cases is 1 - f(T). (For the two computer case this is 1 - 2X(F(T)/2), since both computers must operate to provide the power, P.) The interesting case, providing the "fail soft" operation, occurs when the probability of providing at least P/2 processing power is $1 - f(T)^2/4$; that is, there is a very high probability of having at least P/2 power. We have treated systems B and C above as the same, and taking the total user population as a whole, they are identical. If, however, we consider the P/2 power load case, then it is important to be able to have the switch system B provides so that all the users can be switched to the operational computer. (As a secondary benefit, one of the systems can be maintained (preventative) while there is a load of P/2.)

THE CMU NETWORK PROJECT RESEARCH

The CMU network project is organized as a collection of quasi-independent subprojects whose final goal is the construction and operation of a cost-effective, reliable, eminently usable network system. Along the way we expect to identify a number of fundamental aspects and to seek basic solutions to them that extend well beyond the particular networks we will construct. On the following pages some of the subprojects are described. The discussion is not meant to be complete, but only to give an idea of some aspects of the research.

PLN (Purposely Limited Network)

A rudimentary network is under construction in order to gain experience and to provide data for other phases of the project. It is not intended that this network will be suitable for general use, but it will be adequate for testing some hardware interface designs and for studying system behavior under artificial loads. The experience includes modifying the various monitors and writing software which can be used to measure various performance characteristics. These will be used as inputs to successive models.

The PLN is constrained by equipment currently available and will most probably consist of two PDP-10[3] computers (C10), a hybrid computer (Ch),a PDP-8 computer (C8) used as a message concentrator, and a TSS/8 time-shared PDP-8 (C8TS). The TSS/8 has a small file (about 1.5 million characters)which is

[3]*The PDP-10 was selected as the main processor because it is large enough to serve about 50 online users, yet small enough to have multiple nodes in a network of reasonable cost.*

used for program swapping and for local storage of files. Alternatively, if the disk is not used, C8TS can be used as a message concentrator.

The physical interconnection is shown in Figure 5. (Actually this connection is common to all three proposed structures, Figures 6 to 8.) Four of the computers are physically connected to each other through every possible path with the exceptions that the link between the two PDP-8's is missing and that Ch is only connected to one C10. The principal reason for all the physical links is to be able to study the various structures, and in the event of failure in any of the nodes, transform the network into another working structure. For this reason, it is necessary that both PDP-10's be able to access the file system. Figures 6a and 6b show a non-hierarchical organization with both C10 and Ch operating independently.

In the case of the first organization, Figure 6a, there are essentially just two large computers and Ch, since the C8's are serving the role of message concentrators (the file on TSS/8 is not being used). There is asymmetry, since only one C10 holds the system files. In this structure both large computers would function autonomously, each with its own clientele, and the only communication would be for files and occasionally perhaps longer jobs which had been set up to run in a batch queue would be transferred from one computer to the other for load sharing. In the second system, Figure 6b, there are four functionally independent computers; each of the four computers operates with its own set of users. With this structure it would also be possible for any user to be physically attached to a given computer, but then be routed to the appropriate computer for actual execution. Thus, each computer would function in a message switching mode. For example, a user whose messages were switched via the C8 might eventually be run on the C8TS.

In the next structure, Figure 7, one of the large computers is serving the role of switch, central file, and also performing some computation. The lower C10 would only perform computation when directed, and all I/O and file traffic would go through the upper (master) C10. The middle C8 would function as a message concentrator for the master C10, and the C8TS would serve either as a message switch, or alternatively, if the capability were allowed, doing any task for which it were unnecessary to use the larger computers C10. The decision for the processing would probably be made at the master C10. Finally, if we extend the idea of a hierarchy, and yet distribute the control, we get the structure of Figure 8. Hierarchies would normally have more members at each level, of course, but with this structure the ideas could still be tested. Each level would be responsible for a certain class of tasks (see the following section on dominance). Only if the task could not be completed at a level, would it be passed to a higher level. Here the link between the two PDP-8's is needed. Note, just as is the case with other hierarchical organizations, that there are times when a message has to be passed through all four computers.

NETWORK INFORMATION TRANSMISSION RESEARCH

The nature of information flow in a computer network has similarities to flows in other networks. The network is a structure which provides paths (alternatively called branches, links and arcs) for transferring a commodity from one place (network node) to another place. We will not consider transmitting a commodity to many nodes in parallel (i.e., broadcasting), though this is a possibility open to information flow networks which is unavailable for other types (e.g., material flow networks). The simplest network provides for the flow of a single anonymous commodity (e.g., electricity distribution, water and oil pipelines). Other networks carry anonymous commodities in a discrete fashion (e.g., oil distribution system using tankers or tank cars).

Information networks carry a discrete, non-anonymous commodity. Labels identify the information so that it may be routed to specific destinations, the labels also being information. Similar networks include roads and highways, trucking, postal systems, and telephone systems. These networks carry physically large entities (e.g., letters, boxes, cards) compared with the routing information. The nature of information suggests that almost all of these networks carry information, i.e., messages. Although the media used to carry information differs among postal and computer networks, they are both moving information.

Telephone networks and proposed computer networks differ significantly in several ways. The telephone network is several orders of magnitude larger. A significant part of a telephone network is concerned with switching and routing. Computer networks often are fixed structures with the computers routing (or switching) the information themselves. Another difference between the telephone and computer network is the wider range of information transmission rates required for the computer.[1] Terminal and information carrying links operate over a wide range of data transmission rates. This range requires either multiple paths or large capacity paths between node pairs. The information carrying rates of the more common terminals and links are given in Table 1.

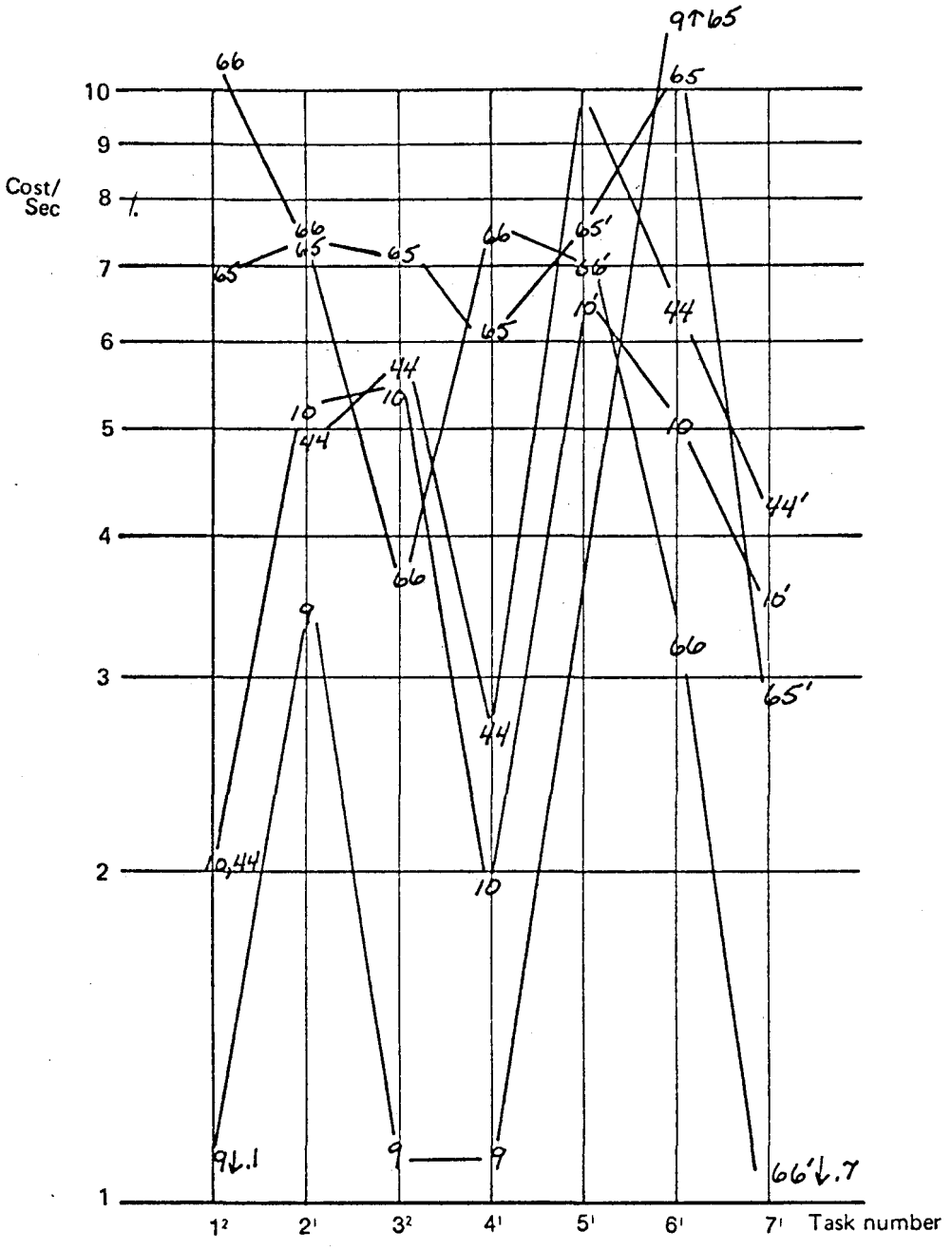
Terminal	Data-rate (in bits/sec)
typewriter	110 ~ 300
scope	1200 ~ 2000
line-printer/card-reader	1400 ~ 10,000
Link	
asynchronous (very low speed)	0 ~ 300
asynchronous (low speed)	300 ~ 2000
low speed synchronous	2400
medium speed synchronous	2400 ~ 10,000
high speed synchronous	40.8, 50, 230, 460 x 10^3
very high speed synchronous	1.5 x 10^6

Table 1

Information Carrying Capacities of Terminals and Links

Since each of these terminals and links deals in information quanta, called messages, the important characteristics are the message length and its corresponding delays. Figure 9 plots the message transmission time versus message length for various data rate links,and shows acceptable performance values for

[1] *The telephone system also transmits video broadcast information for the TV industry and it by default transmits information among computers. The differences between audio (speech) transmission (via the telephone network) and computer information transmission may eventually require separate networks.*



Tasks: 1—Computer off
2—add instruction
3—"average" instructions
4, 5—AZTEC Kernel
6, 7—Cooley-Tukey Kernel
¹Cost-task data taken from Cox, 1968
²Cost-task data taken from Roberts, 1969

Figure 1 Cost/sec *xk* for various tasks Computers: 9-small; 10, 44-medium; 65, 66-large; primed ' computers are large versions of other machine.

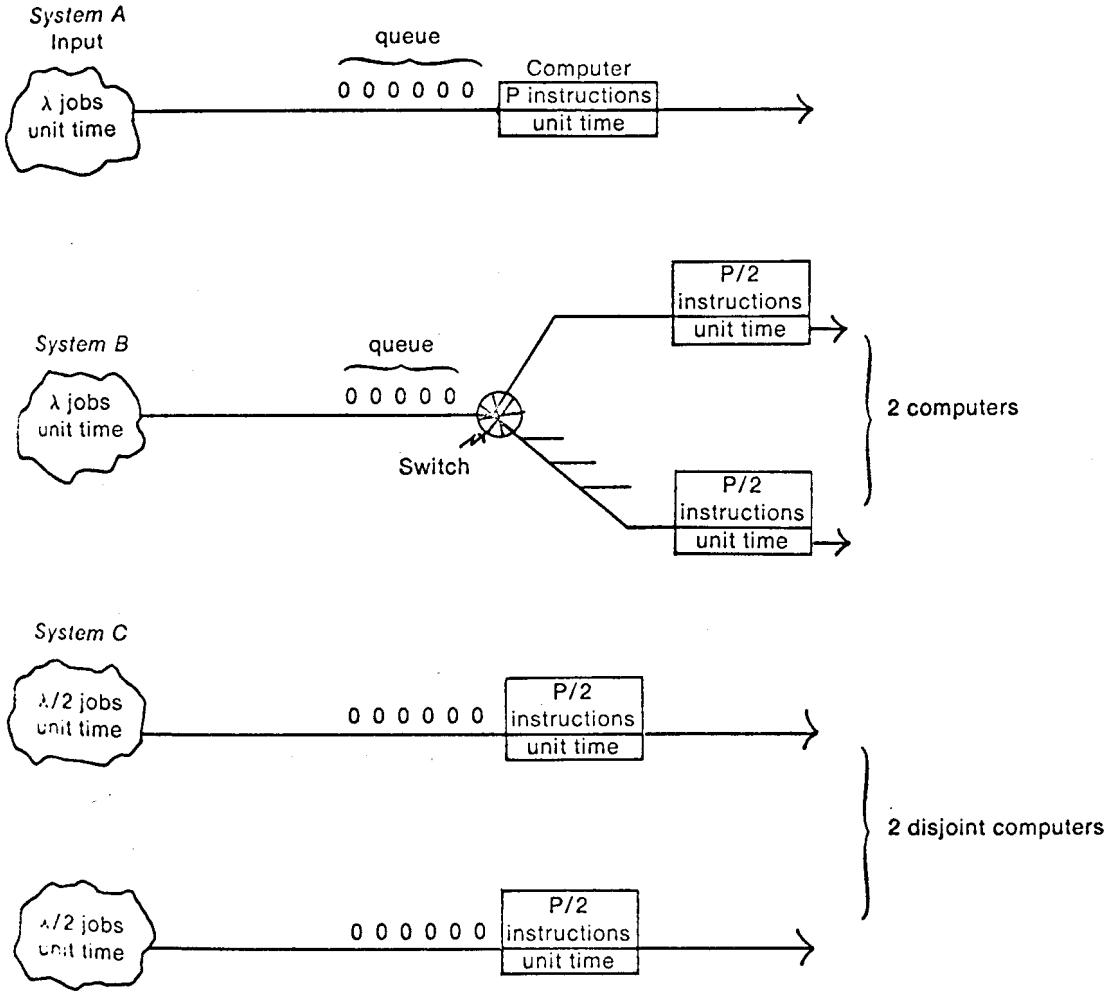


Figure 2

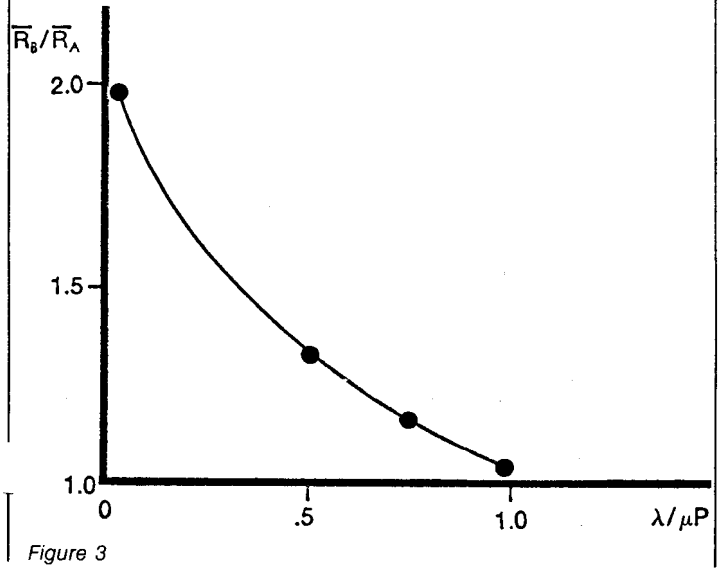


Figure 3

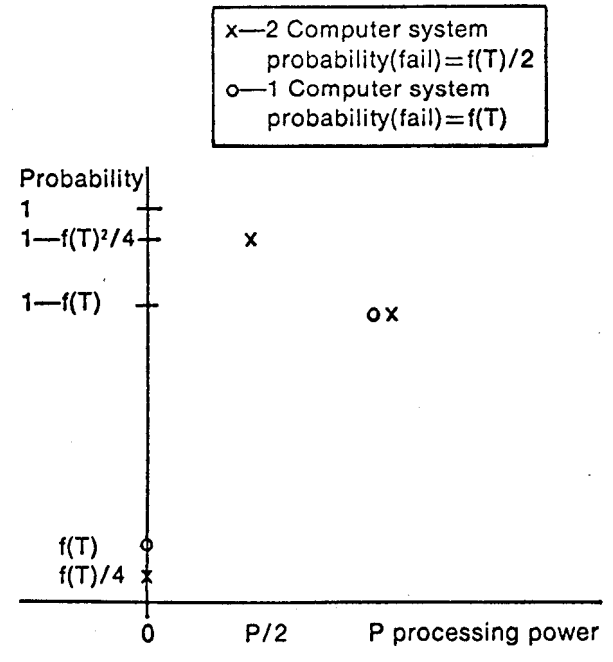


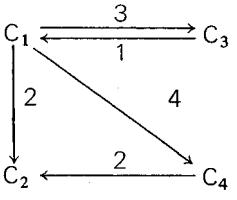
Figure 4 Probability of obtaining at least a given processing power.

various tasks. For instance, many computer networks would like to buy their transmissions from the telephone system for very short intervals (milliseconds), at very high data rates, and with short switching time (milliseconds), i.e., bursts. Switching time and pricing policies within the telephone system conspire to make this difficult to do. The telephone switching networks are often electromechanical and switching delays of 20 seconds are possible. Thus, with networks, links and switches become important components of the structure.

Since a network provides the physical structure for transmitting information, it determines whether the message transmission requirements can be met. Here we are not concerned with *why* information should flow among the nodes, nor how the information flow is controlled (switching and routing policy). The following attributes of a network are relevant to the analysis of information flow: the traffic it must carry (in essence the information transmitted between node pairs); the network's components and their interconnection structure; and finally, the policy of information transmission. We will discuss the above attributes in turn.

A significant constraint of a network is the traffic it must carry. In effect, the traffic already suggests a particular network. We can call it the logical network. It is the network formed by positing a direct link from the transmitting node to the receiving (and using) node, independent of whether there is a physical link between the nodes.

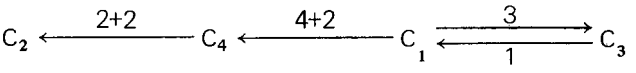
The following example will show the difference between the logical network and various physical implementations that can realize it. Consider the traffic flow in the logical network among four computers: C_1, C_2, C_3 and C_4.



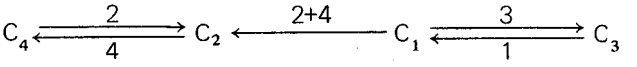
The flow F_{ij} is from computer i to computer j; these flows (in units of kilobits/sec, say) are: $F_{12}=2$, $F_{13}=3$, $F_{14}=4$. $F_{31}=1$, and $F_{42}=2$. In the network figure the flows are marked along the directed links. The same information can be represented in the following flow matrix:

from \ to	C_1	C_2	C_3	C_4
C_1	-	2	3	4
C_2	0	-	0	0
C_3	1	0	-	0
C_4	0	2	0	-

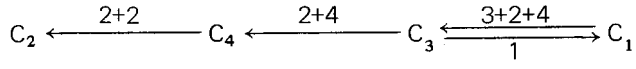
There are many physical networks which satisfy the information flow constraints imposed by the logical network. For example,



or

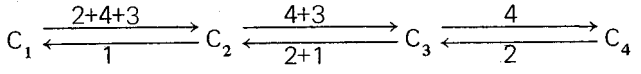


The initial logical network inherently has a path of length one from each source node to each destination node, whereas in the two cases above, information sometimes passes through two links and one node to reach the final destination. For the following structure,

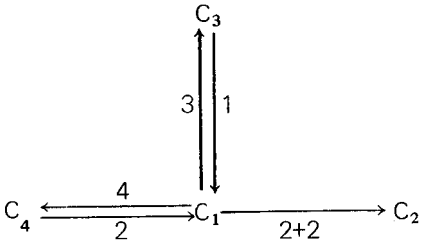


three nodes and two links have to be traversed in order for C_1 to communicate with C_2.

The most expensive physical network is:



Obviously, many structures which are not in the form of a chain are possible, e.g.,



The physical component properties of the paths (links) form the most rigid constraints for a design since they constrain the information flow to a maximum rate. The individual nodes also constrain the structure, since only a certain number of bits can connect to a given node. Also, the flow into a node usually implies the node's capacity (to process) will be degraded. The logical network and message constraints are less rigid because they can be adjusted if necessary to provide less capability for the network. Thus, a physical network structure is to be optimized relative to the logical network objectives. The objectives are flow, reliability, and cost. The problem is thus:

Given: What is to be transmitted (messages); where it is to be transmitted (logical network); and by whom it is to be transmitted (components).

Determine: The structure of components which meets this function.

Since a network structure may not be static over a several year period, the residual information carrying capacity left in the network is also important. Planning for future structures can take several forms. The future traffic can be the initial constraints, thereby insuring an inherently costly initial design. Each local phase of the network can be optimally designed with new components replacing older components at each change of phase. Or, the network can be optimally designed to operate over a long period of time such that each transition includes the old components of the network. The last scheme takes the final state of the network as given, and intermediate planned states are obtained for the short term which are derived from the final design. However, it must cope with substantive uncertainties as to the nature of the final state.

Control Structure Research

The purpose of the network control structure is to realize execution of a set of well-defined functions on the given nodes. There are the major functions such as "run a FORTRAN job" and the internal functions that accompany the execution of such jobs, such as "load the compiler". Normally, in a network a function can be executed on several nodes. The execution of a function on a single node is called a process. Thus, the active network consists of a set of processes, each of which resides in reality in one of the physical nodes. A function then determines a class of input-output equivalent processes. However, the performance of a function by the different processes of an equivalence class at a given instant of time may vary in cost. The variance is determined by the particular process and the node this process resides in, and by time dependent factors, such as current load and location of files. The execution of a job requires in general the performance of several functions, i.e., the activity of several processes. A process is activated by means of a command and we shall say that a process, P_i, *dominates* a process, P_j, if P_i is permitted to issue a command to P_j. Note that it is individual processes, not computers, that dominate and are subject to domination. Care must be taken that the rules of issuing commands and the established dominance relations do not allow a circular command stream (as in our sine-cosine example at the beginning).

The notion of dominance is conveniently expressed by a directed graph, the nodes of which are processes. An arc from process P_i to process P_j represents the non-empty subset of commands P_i may issue to P_j. Consequently, the space of possible control structures is precisely the space of such directed graphs. An equivalent representation of these directed graphs is the connection matrix of the graph. A connection matrix, A, is a square array of elements, a_{ij}, whose values are the weights along the arc from node i to node j (or zero if the arc does not exist). Thus, the space of control structures among n processes may also be characterized as that of all n × n matrices whose elements assume values from the set of possible commands.

The interesting research aspects of the network control structures deal with the questions of communication between processes, the space of control structures when the physical nodes with their capabilities are specified, and the selection of a viable structure out of a given space of control structures. The latter may be particularly hard to answer because of conflicting criteria, such as reliability (which requires redundancy), cost performance and functional specialization of nodes.

As an example of a control structure, imagine a simple network consisting of a master computer (C_1) and two slave computers (C_2 and C_3). Further assume that editing can occur on the master (C_1) and on one of the slaves (C_2), that either of the slaves can run FORTRAN jobs, and that all files are accessed (or stored) only through the master. Denote the processes as

CLI_1	command language interpreter (on C_1 only)
Ftn_2 and Ftn_3	Fortran (on computers 2 and 3)
Ed_1 and Ed_2	Editors (on computers 1 and 2)
Fet	Fetch a file (on C_1)
Sto	Store a file (on C_1)

The relevant commands are run Fortran, RF(file); run editor, RE(file); fetch a file, F(file); and store a file, S(file). A possible control structure for this simple network might be as shown below in Figure 10, which is also represented by the following connection matrix (note the partitioning, indicated by the dotted lines, which denote the association between processes and nodes):

	CLI_1	Ed_1	Fet_1	Sto_1	Ed_2	Ftn_2	Ftn_3
CLI_1	0	RE	0	0	RE	RF	RF
Ed_1	0	0	F	S	0	0	0
Fet_1	0	0	0	0	0	0	0
Sto_1	0	0	0	0	0	0	0
Ed_2	0	0	F	S	0	0	0
Ftn_2	0	0	F	S	0	0	0
Ftn_3	0	0	F	S	0	0	0

The connection matrix representation is an especially convenient one for testing whether a particular control structure satisfies continuity and redundancy constraints. For example, one can easily see that for this structure the entire network fails if CLI becomes inoperative (even if its host node continues to function). Of course, for such a simple network the same fact was obvious from the English description. However, similar problems would not be quite so obvious in larger networks with distributed control.

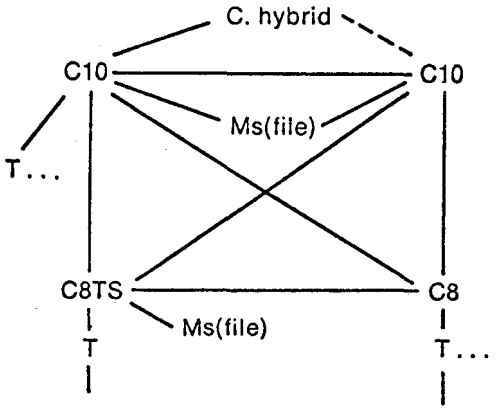


Figure 5 Interconnection of computers.

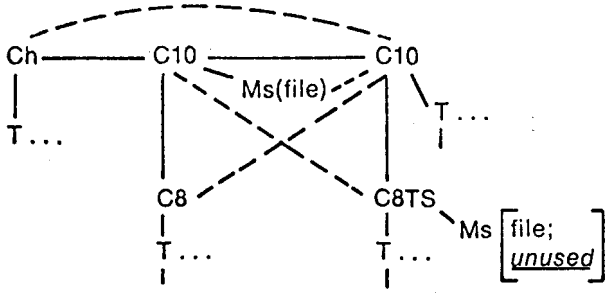


Figure 6a Three equi-level

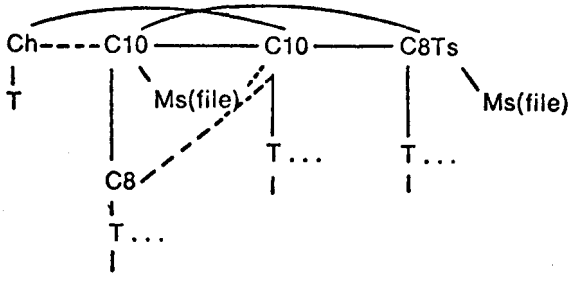


Figure 6b Four equi-level

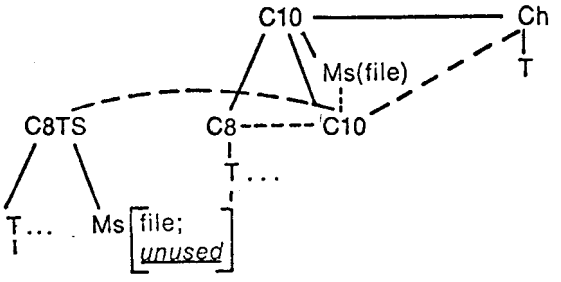


Figure 7 Two level structure with central filing, switching and computation; pre-processing may occur at C8TS; Ch is normally operated independently.

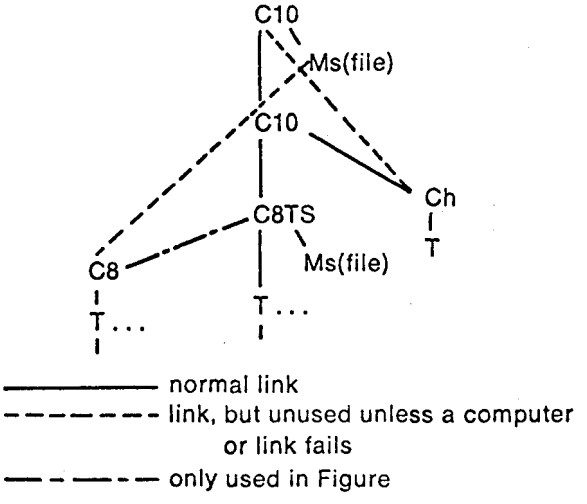


Figure 8 Hierarchical structure. Each level handles processing appropriate to the level.

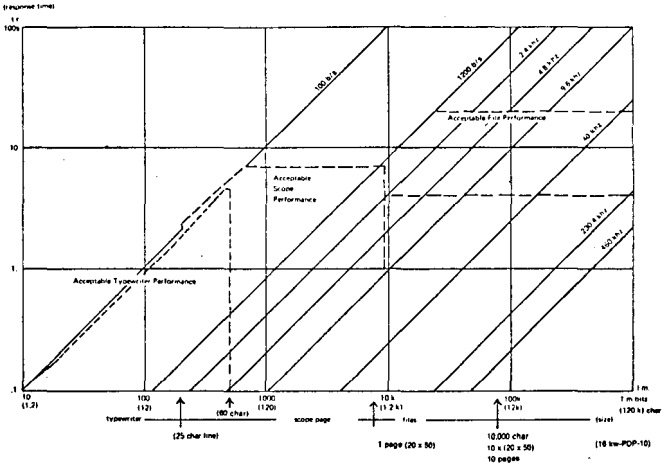


Figure 9 Response time (t.r.) vs. Information Transferred (i.m.) for various Speed lines (r.i.).

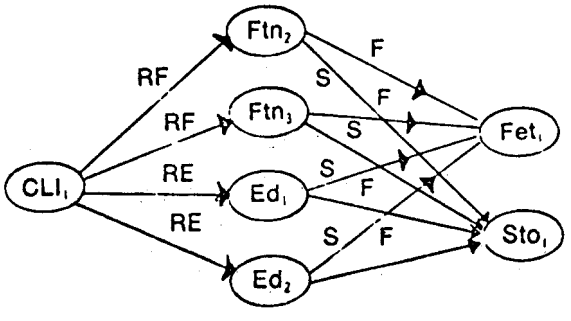


Figure 10

CONCLUSION

We hope we have given the reader an insight into our feelings as to why it is important to research networks. The discussion is not meant to be exhaustive, but merely illustrative about some of the areas. Although network research should be ongoing for the foreseeable future, a short-term goal is the design of several operating networks. Most of the research areas are not unique to networks, but networks do provide a generalization of the questions that occur in classical areas of computer systems and systems programming. These questions can be identified with the appropriate areas as follows:

Operations research. What machines are necessary? What links are necessary? How are messages routed to meet given performance criteria?

Performance measurement and accounting. How is the system being used? What is the efficiency (according to given criteria) of the system? What measurements must be taken for planning of new systems?

Hardware and physical components structure. What links can be connected among the computers? How are the links connected? How much redundant information must be included to operate at a given reliability? How are the various components interconnected? How should future networks be structured?

Operating system and control structure. What is the structure of an individual operating systems? How are the various operating systems coordinated? How are new systems with different structures interfaced to the network? Where does control for tasks reside? In one computer? Distributed over many? Is parallel computing possible?

Command (control) language. Is there a separate, interactive language to control tasks? Can tasks write programs to control tasks?

Systems programming. In what language is the system written?[5] What system structure will allow the programs to be written clearly and correctly?

Future generation networks. What should the next generation computers and their operating systems be in order to operate in a network environment? What should the communications link environment be to provide for various network forms? What kind of network will be needed when homes have terminal computers?

Project organization. How are subprojects organized? How can a system be experimental and still operate with a realistic computing load? How ambitious should the various intermediate networks be?

We leave the reader with an initial list of questions which we hope the next few years of research will begin to answer. The concrete output of the research will be several specific networks, but these networks in turn will generate further questions—and the desire for better networks and computers.

---

[5] *Although not discussed herein, the language BLISS is being used as the implementation language in which to write the system. The first task of BLISS was as a language for BLISS's implementation.*