computer programming. On one hand, it has to give immediate answers to simple requests; on the other hand, it has to provide powerful programming capabilities.

A milestone in the development of interactive systems was the Culler-Fried-System, which strongly influenced the early AMTRAN development. Prof. Culler's system represents a highly powerful multi-console system. A disadvantage is that it does not stay close to the mathematical notation, and it is not simple and easy to learn.[7]

CPS is a conversational subset of PL/1. It has a calculator mode and a program mode and is a useful conversational system although it does not have AMTRAN's flexibility and power in array and function handling.

Iverson's language APL (A Programming Language) is a more formal approach to application programming. It is particularly useful for classical mathematical applications, and it has been implemented as a powerful interactive time-sharing system. The language has features such as array arithmetic, programming capabilities, and a large set of primitive operators including matrix handling operators. An extensive set of special symbols is used instead of keywords. Thus, a special typewriter is necessary. The proponents of APL claim that its source code is more efficient per statement than that of any other programming language. On the other hand, it is less readable. One has to learn special symbols instead of using mnemonics. For example, the quad $\square$ in APL is less informative as an output operator than the TYPE in AMTRAN.

Major disadvantages are that APL does not follow classical mathematical notation, there is no hierarchy among operators, and the order of execution of statements is from right to left. This means the mathematician and scientific nonprogrammer must convert his formulas, written in normal textbook format, into the APL-notation, and the programmer experienced in any other language is even more confused. APL is a language which requires both care and training for simple applications.

CONCLUSIONS

AMTRAN, as described in this paper, can be implemented on a small computer. Such a small computing system is a serious alternative to a console of a commercial time-sharing system. The present developments on the hardware market—a decrease in the price of small computers—make the outlook for problem-solving systems like AMTRAN, particularly bright.

ACKNOWLEDGMENT

BIBLIOGRAPHY

1 J REINFELDS   L A FLENKER   R N SEITZ
  P L CLEM
  Proceedings of the Annual Meeting of the ACM p 469
  Thompson Press New York 1966
2 G KRATKY   H KOPETZ
  The semantics of a mathematically oriented computer language
  Proceedings of the ACM National Conference San Francisco
  August 1969
3 G KRATKY
  The definition of AMTRAN
  Computer Center AMTRAN Report University of
  Georgia 1969-4
4 B D FRIED
  On the users point of view
  Interactive Systems for Experimental Applied Mathematics
  M Klerer and J Reinfelds eds Academic Press New York
  1968
5 P NAUR et al
  Revised report on the algorithmic language ALGOL 60
  Communications of the ACM Vol 6 pp 1–17 January 1963
6 N ESKELSON   H KOPETZ
  AMTRAN-manual for the IBM 130 computer
  To be Published 1969
7 W D ORR
  Conversational computers
  John Wiley and Sons Inc 1968

Computer network development to achieve resource sharing

by LAWRENCE G. ROBERTS and BARRY D. WESSLER

Advanced Research Projects Agency
Washington, D.C.

INTRODUCTION

In this paper a computer network is defined to be a set of autonomous, independent computer systems, interconnected so as to permit interactive resource sharing between any pair of systems. An overview of the need for a computer network, the requirements of a computer communication system, a description of the properties of the communication system chosen, and the potential uses of such a network are described in this paper.

The goal of the computer network is for each computer to make every local resource available to any computer in the net in such a way that any program available to local users can be used remotely without degradation. That is, any program should be able to call on the resources of other computers much as it would call a subroutine. The resources which can be shared in this way include software and data, as well as hardware. Within a local community, time-sharing systems already permit the sharing of software resources. An effective network would eliminate the size and distance limitations on such communities. Currently, each computer center in the country is forced to recreate all of the software and data files it wishes to utilize. In many cases this involves complete reprogramming of software or reformatting the data files. This duplication is extremely costly and has led to considerable pressure for both very restrictive language standards and the use of identical hardware systems. With a successful network, the core problem of sharing resources would be severely reduced, thus eliminating the need for stifling language standards. The basic technology necessary to construct a resource sharing computer network has been available since the advent of time-sharing. For example, a time-sharing system makes all its resources available to a number of users at remote consoles. By splicing two systems together as remote users of each other and permitting user programs to interact with two consoles (the human user and the remote computer), the basic characteristics of a network connection are obtained. Such an experiment was made between the TX–2 computer at Lincoln Lab and the Q–32 computer at SDC in 1966 in order to test the philosophy.[1] Logically, such an interconnection is quite powerful and one can tap all the resources of the other system. Practically, however, the interconnection of pairs of computers with console grade communication service is virtually useless. First, the value of a network to a user is directly proportional to the number of other workers on the net who are creating potentially useful resources. A net involving only two systems is therefore far less valuable than one incorporating twenty systems. Second, the degradation in response caused by using telegraph or voice grade communication lines for network connections is significant enough to discourage most users. Third, the cost to fully interconnect computers nation-wide either with direct leased lines or dial-up facilities is prohibitive. All three problems are a direct result of the inadequacy of the available communication services.

DESIGN OF A NETWORK COMMUNICATIONS SERVICE

After the Lincoln-SDC network experiments, it was clear that a completely new communications service was required in order to make an effective, useful resource-sharing computer network. The communication pipelines offered by the carriers would probably have to be a component of that service but were clearly inadequate by themselves. What was needed was a message service where any computer could submit a message destined for another computer and be sure it would be delivered promptly and correctly. Each interactive

543

conversation or link between two computers would have messages flowing back and forth similar to the type of traffic between a user console and a computer. Message sizes of from one character to 1000 characters are characteristic of man-machine interactions and this should also be true for that network traffic where a man is the end consumer of the information being exchanged. Besides having a heavy bias toward short messages, network traffic will also be diverse. With twenty computers, each with dozens of time-shared users, there might be, at peak times, one or more conversations between all 190 pairs of computers.

Reliability

Communications systems, being designed to carry very redundant information for direct human consumption, have, for computers, unacceptably high downtime and an excessively high error rate. The line errors can easily be fixed through error detection and retransmission; however, this does require the use of some computation and storage at both ends of each communication line. To protect against total line failures, there should be at least two physically separate paths to route each message. Otherwise the service will appear to be far too unreliable to count on and users will continue to duplicate remote resources rather than access them through the net.

Responsiveness

In those cases where a user is making more or less direct use of a complete remote software system, the network must not cause the total round-trip delay to exceed the human short-term memory span of one to two seconds. Since the time-sharing systems probably introduce at least a one-second delay, the network's end-to-end delay should be less than ½ second. The network response should also be comparable, if possible, to using a remote display console over a private voice grade line where a 50 character line of text (400 bits) can be sent in 200 ms. Further, if interactive graphics are to be available, the network should be able to send a complete new display page requiring about 20 kilobits of information within a second and permit interrupts (10–100) to get through very quickly, hopefully within 30–90 ms. Where two programs are interacting without a human user being directly involved, the job will obviously get through sooner, the shorter the message delay. There is no clear critical point here, but if the communications system substantially slows up the job, the user will probably choose to duplicate the remote process or data at his site. For such cases, a reasonable measure by which to compare communications systems is the "effective bandwidth" (data block length for the job/end-to-end transmission delay).

Capacity

The capacity required is proportional to the number and variety of services available from the network. As the number of nodes increase, the traffic is expected to increase more than linearly, until new nodes merely duplicate available network resources. The number of nodes in the experimental network was chosen to: (1) involve as many computer researchers as possible to develop network protocol and operating procedures, (2) involve special facilities, such as the ILLIAC, to distribute its resources to a wider community, (3) involve as many disciplines of science as possible to measure the effect of the network on those disciplines, and (4) involve many different kinds of computers and systems to prove the generality of the techniques developed. The nodes of the network were generally limited to: (1) those centers for which the network would truly provide a cost benefit, (2) government-funded projects because of the use of special rate government-furnished communications, and (3) ARPA-funded projects where the problems of intercomputer accounting could be deferred until the network was in stable operation. The size of the experimental network was chosen to be approximately 20 nodes nation-wide. It was felt that this would be large and diverse enough to be a useful utility and to provide enough traffic to adequately test the network communication system.

For a 20 node network, the total traffic by mid-1971 at peak hours is estimated to be 200-800 KB (kilobits per second). This corresponds to an average outgoing traffic per node of 10–40 KB or an average of 0.5–2 KB traffic both ways between each pair of nodes. Traffic between individual node-pairs, however, will vary considerably, from zero to 10 KB. The total traffic per node will also vary widely, perhaps from 5–50 KB. Variations of these magnitudes will occur in both space and time and, unless the connunications system can reallocate capacity rapidly (seconds), the users will find either the delay or cost excessive. However, it is expected that the total capacity required for all 20 nodes will be fairly stable, smoothed out by having hundreds of active network users spread out across four time zones.

Cost

To be a useful utility, it was felt that communications costs for the network should be less than 25% of the

computing costs of the systems connected through the network. This is in contrast to the rising costs of remote access communications which often cost as much as the computing equipment.

If we examine why communications usually cost so much we find that it is not the communications channel per se, but our inefficient use of them, the switching costs, or the operations cost. To obtain a perspective on the price we commonly pay for communications let us evaluate a few methods. As an example, let us use a distance of 1400 miles since that is the average distance between pairs of nodes in the projected ARPA Network. A useful measure of communications cost is the cost to move one million bits of information, cents/megabit. In the table below this is computed for each media. It is assumed for leased equipment and data set rental that the usage is eight hours per working day.

TABLE 1—Cost per Megabit for Various Communication Media 1400 Mile Distance

Media

Telegram	$3300.00	For 100 words at 30 bits/wd, daytime
Night Letter	565.00	For 100 words at 30 bits/wd, overnight delivery
Computer Console	374.00	18 baud avg. use[2], 300 baud DDD service line & data sets only
TELEX	204.00	50 baud teletype service
DDD (103A)	22.50	300 baud data sets, DDD daytime service
AUTODIN	8.20	2400 baud message service, full use during working hours
DDD (202)	3.45	2000 baud data sets
Letter	3.30	Airmail, 4 pages, 250 wds/pg, 30 bits/wd
W. U. Broadband	2.03	2400 baud service, full duplex
WATS	1.54	2000 baud, used 8 hrs/working day
Leased Line (201)	.57	2000 baud, commercial, full duplex
Data 50	.47	50 KB dial service, utilized full duplex
Leased Line (303)	.23	50 KB, commercial, full duplex
Mail DEC Tape	.20	2.5 megabit tape, airmail
Mail IBM Tape	.034	100 megabit tape, airmail

Special care has also been taken to minimize the cost of the multiplexor or switch. Previous store and forward systems like DoD's AUTODIN system, have had such complex, expensive switches that over 95% of the total communications service cost was for the switches. Other switch services adding to the system's cost, deemed superfluous in a computer network, were: long term message storage, multi-address messages and individual message accounting.

The final cost criteria was to minimize the communications software development cost required at each node site. If the network software could be generated centrally, not only would the cost be significantly reduced, but also the reliability would be significantly enhanced.

THE ARPA NETWORK

Three classes of communications systems were investigated as candidates for the ARPA Network: fully interconnected point to point leased lines, line switched (dial-up) service, and message switched (store and forward) service. For the kind of service required, it was decided and later verified that the message switched service provided the greater flexibility, higher effective bandwidth, and lower cost than the other two systems.

The standard message switched service uses a large central switch with all the nodes connected to the switch via communication lines; this configuration is generally referred to as a Star. Star systems perform satisfactorily for large blocks of traffic (greater than 100 kilobits per message), but the central switch saturates very quickly for small message sizes. This phenomenon adds significant delay to the delivery of the message. Also, a Star design has inherently poor reliability since a single line failure can isolate a node and the failure of the central switch is catastrophic.

An alternative to the Star, suggested by the Rand study "On Distributed Communications"[3], is a fully distributed message switched system. Such a system has a switch or store and forward center at every node in the network. Each node has a few transmission lines to other nodes; messages are therefore routed from node to node until reaching their destination. Each transmission line thereby multiplexes messages from a large number of source-destination pairs of nodes. The distributed store and forward system was chosen, after careful study, as the ARPA Network communications system. The properties of such a communication system are described below and compared with other systems.

A more complete description of the implementation, optimization, and initial use of the network can be found in a series of five papers, of which this is the first. The second paper by Heart, et al[4] describes the design, implementation and performance characteristics of the message switch. The third paper by Kleinrock[5] derives procedures for optimizing the capacity of the trans-

mission facility in order to minimize cost and average message delay. The fourth paper by Frank, et al[6] describes the procedure for finding optimized network topologies under various constraints. The last paper by Carr, et al[7] is concerned with the system software required to allow the network computers to talk to one another. This final paper describes a first attempt at intercomputer protocol, which is expected to grow and mature as we gain experience in computer networking.

Network properties

The switching centers use small general purpose computers called Interface Message Processors (IMPs) to route messages, to error check the transmission lines and to provide asynchronous digital interface to the main (HOST) computer. The IMPs are connected together via 50 Kbps data transmission facilities using common carrier (ATT) point to point leased lines. The topology of the network transmission lines was selected to minimize cost, maximize growth potential, and yet satisfy all the design criteria.

Reliability

The network specification requires that the delivered message error rates be matched with computer characteristics, and that the down-time of the communications system be extrememly small. Three steps have been taken to insure these reliability characteristics: (1) at least two transmission paths exist between any two modes, (2) a 24 bit cyclic check sum is provided for each 1000 bit block of data, and (3) the IMP is ruggedized against external environmental conditions and its operation is independent of any electromechanical devices (except fans). The down-time of the transmission facility is estimated at 10–12 hours per year (no figures are currently available from ATT). The duplication of paths should result in average down-time between any pair of nodes, due to transmission failure, of approximately 30 seconds per year. The cyclic check sum was chosen based on the performance characteristics of the transmission facility; it is designed to detect long burst errors. The code is used for error detection only, with retransmission on an error. This check reduces the undetected bit error rate to one in 10^{12} or about one undetected error per year in the entire network.

The ruggedized IMP is expected to have a mean time to failure of 10,000 hours; less than one failure per year. The elimination of mass storage devices from the IMP results in lower cost, less down-time, and greater throughput performance of the IMP, but im-

plies no long term message storage and no message accounting by the IMP. If these functions are later needed they can be added by establishing a special node in the network. This node would accept accounting information from all the IMPs and also could be routed all the traffic destined for HOSTs which are down. We do not believe these functions are necessary, but the network design is capable of providing them.

Responsiveness

The target goal for responsiveness was .5 seconds transit time from any node to any other, for a 1000 bit (or less) block of information. The simulations of the network show the transit time of a 1 kilobit block of .1 seconds until the network begins to saturate. After saturation the transit time rises quickly because of excessive queuing delays. However, saturation will hopefully be avoided by the net acting to choke off the inputs for short periods of time, reducing the buffer queues while not significantly increasing the delay.

Capacity

The capacity of the network is the throughput rate at which saturation occurs. The saturation level is a function of the topology and capacity of the transmission lines, the traffic distribution between pairs of nodes (traffic matrix) and the average size of the blocks sent over the transmission lines. The analysis of capacity was performed by Network Analysis Corporation during the optimization of the network topology. As the analysis shows, the network has the ability to flexibly increase its capacity by adding additional transmission lines. The use of 108 and 230.4 KB communication services, where appropriate, considerably improves the cost-performance of the network.
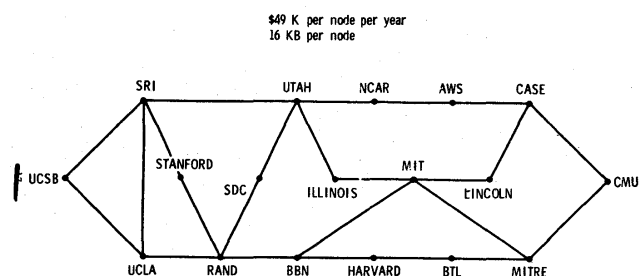

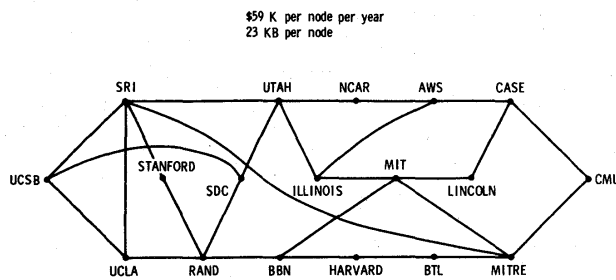Figure 1—ARPA network initial topology

Figure 2—ARPA network expanded topology

Configuration

Initial configuration of the ARPA Network is currently planned as shown in Figure 1. The communications circuits for this network will cost $49K per node per year and the network can support an average traffic of 16 KB per node. If the traffic builds up, additional communication lines can be added to expand the capacity as required. For example, if 23 KB per node is desired, the network can be expanded to the configuration shown in Figure 2 for an increase of only $10K per node per year. Expansion can be continued on this basis until a capacity of about 60 KB per node is achieved, at which point the IMPs would tend to saturate.

COMPARISON WITH ALTERNATIVE NETWORK COMMUNICATIONS SYSTEMS DESIGNS

For the purpose of this comparison the capacity required was set at 500 baud to 1 KB per node-pair. A minimal buffer for error checking and retransmission at every node is included in the cost of the systems.

Two comparisons are made between the systems: the cost per megabit as a function of the delay and the effective bandwidth as a function of the block size of the data. Several other functions were plotted and compared; the two chosen were deemed the most informative. The latter graph is particularly informative in showing the effect of using the network for short, interactive message traffic.

The systems chosen for the comparison were fully interconnected 2.4 KB and 19 KB leased line systems, Data-50 the dial-up 50 KB service, DDD the standard 2 KB voice grade dial-up system, Star networks using 19 KB and 50 KB leased lines into a central switch, and the ARPA Network using 50 KB leased lines.

The graph in Figure 3 shows the cost per megabit versus delay. The rectangle outlines the variation caused by a block size variation of 1 to 10 Kilobits and
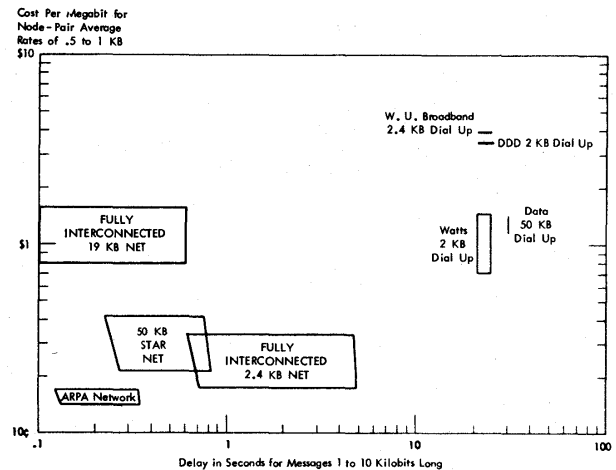

Figure 3—Cost vs delay for potential 20 node network designs

capacity requirement variation of 500 to 1000 baud. The dial-up systems were used in a way to minimize the line charges while keeping the delay as low as possible. The technique is to dial a system, then transmit the data accumulated during the dial-up (20 seconds for DDD, 30 seconds for Data-50). The dial-up systems are still very expensive and slow as compared with other alternatives. The costs of the ARPA Network are for optimally designed topologies. The 19 KB Star was eliminated because the system saturated just below 1 KB per node-pair which did not provide adequate growth potential though the cost was comparable to the ARPA Network. For the 50 KB Star network, the switch is assumed to be an average distance of 1300 miles from every node.

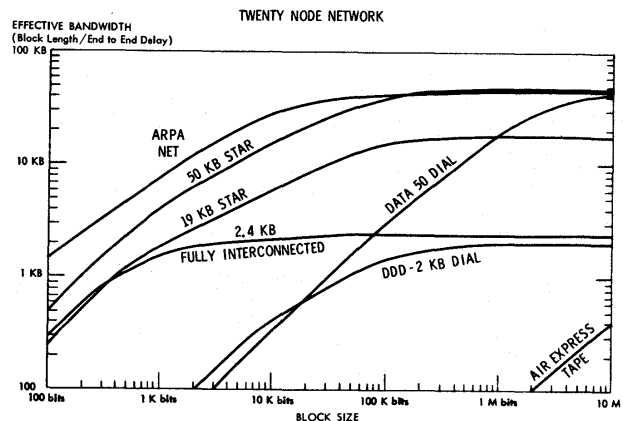The graph in Figure 4 shows the effective bandwidth

Figure 4—Effective Bandwidth vs block size

versus the block size of the data input to the network. The curves for the various systems are estimated for traffic rates of 500 to 1000 baud. The comparison shows the ARPA Net does very well at small block size where most of the traffic is expected.

NETWORK PLANS

Use of the Network is broken into two successive phases: (1) Initial Research and Experimental Use, and (2) External Research Community Use. These phases are closely related to our plans for Network implementation. The first phase, started in September 1969, involves the connection of 14 sites involved principally in computer research. These sites are current ARPA contractors who are working in the areas of Computer System Architecture, Information System Design, Information Handling, Computer Augmented Problem Solving, Intelligent Systems, as well as Computer Networks. This phase should be fully implemented by November 1970. The second phase involves the extension of the number of sites to about 20 to include ARPA-supported research disciplines.

Initial research and experimental use

During Phase One, the community of users will number approximately 2000 people. This community is involved primarily in computer science research and all have ARPA-funded on-going research. The major use they will make of the network is the sharing of software resources and the educational experience of using a wider variety of systems than previously possible. The software resources available to the Network include: advanced user programs such as MATHLAB at MIT, Theorem Provers at SRI, Natural Language Processors at BBN, etc., and new system software and languages such as LEAP, a graphic language at Lincoln Lab, LC^2, an interactive ALGOL system at Carnegie, etc.

Another major use of the Network will be for accessing the Network Information Center (NIC). The NIC is being established at SRI as the repository of information about all systems connected into the Network. The NIC will maintain, update and distribute hard copy information to all users. It will also provide file space and a system for accessing and updating (through the net) dynamic information about the systems, such as system modifications, new resources available, etc.

The final major use of the Net during Phase One is for measurement and experimentation on the Network itself. The primary sites involved in this are BBN, who

has responsibility for system development and system maintenance, and UCLA, who has responsibility for the Net measurement and modeling. All the sites will also be involved in the generation of intercomputer protocol, the language the systems use to talk to one another.

External research community use

During the time period after November 1970, additional nodes will be installed to take advantage of the Network in three other ARPA-funded research disciplines: Behavioral Science, Climate Dynamics and Seismology. The use of the Network at these nodes will be oriented more toward the distribution and sharing of stored data, and in the latter two fields the use of the ILLIAC IV at the University of Illinois.

The data sharing between data management systems or data retrieval systems will begin an important phase in the use of the Network. The concept of distributed data bases and distributed access to the data is one of the most powerful and useful applications of the network for the general data processing community. As described above, if the Network is responsive in the human time frame, data bases can be stored and maintained at a remote location rather than duplicating them at each site the data is needed. Not only can the data be accessed as if the user were local, but also as a Network user he can write programs on his own machine to collect data from a number of locations for comparison, merging or further analysis.

Because of widespread use of the ILLIAC IV, it will undoubtably be the single most demanding node in the Network. Users will not only be sending requests for service but will also send very large quantities of input and output data, e.g., a 10^6 bit weather map, over the Net. Projected uses of the ILLIAC include weather and climate modeling, picture processing, linear programming, matrix manipulations, and extensive work in other areas of simulation and modeling.

In addition to the ILLIAC, the University of Illinois will also have a trillion bit mass store. An experiment is being planned to use 10% of the storage (100 billion bits) as archival storage for all the nodes on the Net. This kind of capability may help reduce the number of tape drives and/or data cells in the Network.

FUTURE

There are many applications of computers for which current communications technology is not adequate. One such application is the specialized customer service computer systems in existence or envisioned for the