Laboratory for Computer Science
Computer Systems Research Division

Request for Comments 230
September 9, 1982

CSS and CSC Proposal to DARPA for 1983-85

by Jerome H. Saltzer, David D. Clark, David P. Reed and David K. Gifford

Attached are the various sections of the 1983-85 DARPA proposal that describe work to be done in the CSS and CSC groups.

---

6. EXTENDED REACH NETWORKS

6.1. INTRODUCTION AND OBJECTIVES

The principal objective of this research is to extend the reach of data communication networks: (1) to single-user computers through city-wide cable; (2) across administrative boundaries; and (3) to portable computers that can be only occasionally network-connected.

As advanced single-user personal computers become widely installed and used, it is necessary that data communication networks reach out to the many scattered places where these machines will show up, such as the office, the laboratory, and the home. Thus, a major goal of our work is to explore network technologies, such as cable television and subcarriers of radio broadcast stations, that have the potential of a broad reach at a cost that is commensurate to that of the computers themselves. These technologies introduce problems, such as assymetry in delay, bandwidth, or reliability between inbound and outbound communications, that require consideration in protocol design and in system integration.

Extending the reach of computer networks involves certain kinds of interconnections that cross administrative boundaries and pose policy control problems. Examples are links from government networks to non-government networks, from one company to another, and even from one country to another. Policy problems that arise include controlling where data is stored and processed, cost accounting, authorization of use, privacy, authentication of users, and permitting transit. Accordingly, an important goal of this research is, by actually implementing examples of such links, to discover ways of providing policy controls that are acceptable to the interconnected parties yet that minimize interference with the function and performance of the underlying data communication systems.

Finally, as single-user computers become more widely used and scattered, it seems inevitable that many of these machines, especially those that are portable, simply cannot remain always in contact with a network. Therefore, another goal of this research is to explore the possibility of occasional interconnection as a sufficient condition for allowing certain applications to be effectively carried out. Another way of describing this research goal is the development of algorithms for managing distributed data that is usually partitioned.

6.2. BACKGROUND AND TECHNOLOGICAL NEED

Extensive recent industrial work has created technologies for local area data communications that permit large numbers of computers to be connected to one another as long as they all are within the same building or in a group of nearby buildings. As soon as one leaves this comfortable environment, however, connection becomes difficult: Telephone lines have bandwidths two or three orders of magnitude smaller than the local area networks, yet at equal or higher connection costs. At the same time the installation of two-way cable television systems throughout major US cities provides us with the opportunity of taking advantage of a low-cost high technology physical plant for data communications.

Most work to date on the use of cable television as a data communication medium has concentrated on one-for-one replacement of telephone lines. That is, developments have concentrated on low data rate (64 kilobits/second and under) point-to-point communications tucked in to the cable television spectrum in convenient places. However, a television cable can easily handle the same megabit/second data rates as the local area data communications networks, as demonstrated principally by the Mitre Corporation. We believe that what is now needed is system integration, i.e., development of data communication systems specifically designed to be compatible with city-wide cable television systems, and gateway integration of that kind of local network, with the more common kinds of local networks that are already being widely installed.

As local area networks are installed at progressively more sites, questions are more and more commonly raised about the possibility of interconnecting such networks to each other. Technically, these interconnections are straightforward, and although some formidable problems remain, such considerations are not a primary focus of our research. When such links, however, cross administrative boundaries, either within an organization or between distinct organizations, requirements for control of the use of the links are often posed by the distinct administrations, or such requirements are belatedly discovered as a consequence of unfortunate experiences.

Several inter-organizational links already exist, and more are proposed. For example, at MIT, several MIT-owned local networks are strongly interconnected with one another, and with the ARPANET and Telenet on one side, as well as with local companies such as Symbolics, Inc., and Lisp Machine, Inc. on the other. Furthermore, additional links are proposed to CSNET, to BITNET, and to the IBM corporate network. Policy problems such as restricting ARPANET use to government purposes, accounting for use of Telenet, and restricting transit across the MIT network, appear daily in this environment.

Within the present state of the art, the technological basis for providing policy control is almost completely non-existent. Today, whenever a packet arrives at the boundary between organizations it is difficult for any person or program to discover its purpose, since that purpose is buried in several protocol layers, and the packet may be only one of many that are part of a single activity. Present approaches for policy control fall into one of three categories, none of which provides both satisfactory function and satisfactory control:

1. Allow the packet to cross, and depend on the end points to initiate communications that meet policy constraints. This technique fails, for example, if network B finds that it can be used as a transit network between stations on network A and stations on network C. In such a case, network B gets no chance to exert any policy control.

2. Require that all protocols terminate at each gateway between networks. For every application, place a program at the gateway to act as a monitor and relay. Since the protocol is terminated, the underlying purpose of the connection is visible to the monitor, which can more easily enforce policy constraints. This approach is analogous to making a telephone call in which each party can talk only to an intermediate operator, who relays the conversation. While acceptable for some applications, this approach exhibits delay and subtle function losses that may cripple certain applications.

3. Do not permit the connection in the first place. This approach provides conservative control, but is rather devastating from an application function point of view. Yet it is probably the most widespread technique used today, out of fear of alternative dangers.

Our proposed work will directly address the problem of policy control by trying approaches which permit a gateway to enforce policy constraints that are set elsewhere. It will also try to discover real policy requirements by examining several cases of interconnection that cross administrative boundaries. Finally, it will develop protocols that provide systematic policy control.

The proposed portability research project addresses two topics that are of current interest: Technological developments in the area of terminals, small computers, and human interface devices in general have proceeded faster than our ability to understand how to take advantage of them. Many apparently sophisticated portable devices have recently appeared on the market. On close inspection, most of them do not appear to be very useful. Accordingly, this project represents a serious attempt at integrating some of these new devices into an operational environment so that we may learn how to use them effectively.

A number of problems arise when computation is distributed among several machines, which must be understood if the concept of distributed computing is to become viable. One of the most difficult of these is the problem of *partitioned data*. One of the principal objectives of our distributed computing research is that, although parts of a system may fail, the user will still be able to proceed, using those parts of the system that remain operational. However, unless the partitioned data problem is solved, it will often be necessary to suspend a user's activities, because there is no way for coordinating that user's actions with simultaneous actions of users on other parts of the system who are temporarily isolated by communications failures within the system.

A general solution to this problem currently seems difficult, if not impossible. However, solution of the problem in specific application domains seems feasible and is expected to yield insights which may eventually lead us to a general solution. Accordingly, we believe that a pragmatic assault on this problem by means of specific projects is the best way of gaining the necessary understanding in this area.

6.3. ACCOMPLISHMENTS

During the last two years we have pursued several closely related research projects preparatory to the research now proposed. In the area of data communication network technology, we have designed, implemented, and installed a ten-megabit/second ring network in our Laboratory [6.1], [6.2]. We have worked closely with industry on this project, in order to transfer this technology outside MIT -- an objective that has been already realized: The design that we developed is now available as a commercial product from Proteon Assoc. in Waltham, MA. A similar design influenced by us was described in three papers published by the IBM Zurich Research Laboratory [6.3], [6.4], [6.5] and by the IBM Communication Products Division which has made a presentation on ring technology to the IEEE 802 local area network standards committee [6.6]. This project has convinced us that cooperative ventures with industry can be an effective technique in reducing the number of different skills required to carry out a research project. We intend to take the same approach on those parts of our proposed research that involve hardware development.

Our Laboratory has also been active in learning about inter-network interconnection, through its role as one of the DARPA groups leading in implementation of the TCP/IP protocol family. More specifically, experiences in installing gateways that allow packets to flow between two ring networks, the Chaosnet, an experimental Ethernet, and the ARPANET, has led us to certain conclusions concerning the architecture of such connections. In particular we have established that a hierarchical approach involving a centrally-administered higher-level backbone network is essential for maintaining a useful system because individual local networks are usually locally administered [6.7], [6.8]. This hierarchical approach was also explored in the design of a service that finds routes through a network that crosses several administrations [6.9]. A related architectural conclusion, drawn from this experience, is that on-the-fly translation of end-to-end protocols does not work very well except in special cases -- effective communication seems to take place only when participating parties agree on the protocol to be used [6.10]. This conclusion led us to the development and implementation of a multi-protocol forwarding gateway program, written in the C language for easier extensibility to additional protocols. It also provided us with insight on the problem of naming network destinations when inter-network connections are involved [6.11].

Finally, over the past two years we have carried out a number of preliminary studies for the portability project. Some of these studies involved an assessment of possible hardware configurations for portable devices. Once such configuration that was chosen for detailed study involved a small keyboard and a single line of LCD text display. This configuration was selected based on small pocket-sized computers which were coming on the market at the time of our research. We did a feasibility study of how this configuration could be used to facilitate rapid reading as might be appropriate for a computer mail package. Using a prototype implementation, as well as simulation on other display devices, we concluded that there did exist display patterns which made possible the reading of text at a reasonable rate. However, as we were studying this configuration we realized that the technology was evolving faster than we had initially expected. This observation led us to the conclusion that we should target our future research for a more sophisticated configuration, with several lines of display. In fact, it became clear that the only way we can develop a working system is to depend on whatever hardware is available at purchase time. For this reason, we intend to orient our proposed research more on the software and application issues rather than on the underlying hardware.

Another aspect of work to date has been a preliminary study of how the mail management application might be distributed between the portable device and a mailbox manager. On the basis of this study we feel that we now understand how to approach the problem of distributing the mail application: Our approach is consistent with the general position that we have taken on the semantics of distributed systems -- the knowledge of distribution, and of the partitions in this case must be jointly understood and controlled by the application and by the user. In particular, it is not possible to hide the distributed or partitioned nature of the database completely from the user. Instead, what must be done is to make the partition visible to the user in a clear, simple, and straightforward manner, so that the latter can understand exactly what is happening in the overall system.

6.4. PROPOSED EFFORT AND MILESTONES

In developing the technology of extended reach networks via cable television, we plan to engage in cooperative ventures with other organizations that can provide appropriate testbeds and hardware engineering, so that we, in turn, can concentrate on system integration issues such as gateway and protocol design. Several links with other organizations have already been established. The Newton, MA cable television system of Continental Cablevision of Massachusetts has indicated both willingness and interest in providing a testbed site, and the MIT educational cable television system is also available to us for such experiments. In addition Ungermann-Bass Corp. has proposed to use the MIT educational cable as a site for beta-test experiments for certain data communication products. Finally, the GTE-Telenet research laboratory, which has interests in data communications and support from GTE-Sylvania in cable television hardware, has expressed interest in a joint research project.

We plan to proceed on two parallel paths. First, using available, off-the-shelf hardware that operates at telephone-line speeds (9600 bits/second) we intend to place a gateway computer in the City of Newton and a telephone link to a gateway computer at MIT for the purpose of attaching Newton gateway to the Newton cable, along with several home computers already (or to be) located in the homes of MIT faculty and staff who reside in that city. This first effort has two immediate objectives: first, it establishes an operational hardware path, albeit a low-speed one, which in turn allows the software and protocol aspects of our proposed project to proceed expediently. Second, it provides some early experience with problems of noise ingress and accumulation, compatibility, and setup of reverse channels. All of these problems while more difficult at higher data rates, will certainly provide us with some early indications as to the limits that will be eventually encountered.

The second parallel path of our plan involves our cooperation with an appropriate firm for the development of hardware that operates at data rates of one Megabit/second or above over the cable television system. This effort is technologically more difficult and thus will begin by a campaign of careful measurement and analysis to characterize the environment of the available transmission channels on residential cable television systems.

Two milestones characterize these efforts: The first is when a home computer in the City of Newton can use the Internet Telnet and file transfer protocols to log in or exchange files with computers at MIT and elsewhere in the ARPANET. The second is when the same kind of interchange can be done using the high-data-rate communication system.

In the area of policy controls across administrative boundaries, we propose three specific projects: The first of these is to explore approaches that would allow a packet to carry its authorization for boundary crossing in a way that is easily verifiable by a gateway, in the same sense that a traveler's visa permits a border guard to verify a particular border crossing. The basic idea is to create a token with a public-key encryption system. This token would be added to the Internet packet header, and examined through decryption for validity at each boundary-crossing gateway. We plan to implement such a system and try it out first on the boundary between two local networks inside MIT. If a successful strategy is established, we will then propose to add this approach to the standard Internet Protocol, thereby causing IP gateways between organizations to require the presence of these visa-type tokens.

The second proposed project in this area is based on earlier research at our Laboratory to develop an authentication service based on encryption key distribution and a directory assistance service in the form of an on-line version of the MIT telephone directory. We propose to take these two services and a third, as yet undeveloped, service that would provide accounting functions, and wrap them together with a unified set of protocols that integrates naming, authentication, and accounting in a coordinated way. Although the exact outline of this project needs to be further developed, we can identify a functional milestone: the ability to send an electronic message out over, say, Telenet Telemail, with satisfactory checks on the authenticity of the sender and an automatic and appropriate charge to the sender's account.

The third proposed project on policy controls involves the addition of a corporate computer network. A candidate, with which unofficial discussion as to both technical and policy feasibility has been initiated , is the IBM Corporate Job Network. This candidate is particularly interesting because it probably represents an extreme case in terms of the stringency and accuracy of required policy controls. The milestone of this project is the following: Electronic mail between authorized parties can be originated either within the ARPANET or the IBM network and terminate at the other network; with satisfaction as to policy control expressed both by the Defense Communications Agency and by persons responsible for asset control within IBM.

The proposed work on portability extensions involves two parallel activities. Since the spectrum of available technology for portable terminals and computers has advanced rapidly over the last year, we find it necessary for us to carry out a revised technology study, for the purpose of establishing an appropriate experimental testbed. In parallel with this work, we intend to pursue software prototype development for a distributed mail package. This software project consists of three parts. First, we will develop a mailbox server, i.e., a machine, connected to the internet, which will act as the user's agent for sending and receiving mail. This machine will implement the standard internet mail sending protocols, SMTP, and will provide a reliable and stable storage of mail items using the UNIX file system for its storage. Second, we will implement a user interface program for UNIX to provide client access to this mailbox server. The two programs will be structured to communicate with each other using network protocols, so that they can be on different physical machines. Our intention here is that the client program, although initially used on a time-sharing version of UNIX, will be shortly moved to the user's personal computer. Third, we propose to develop a second user interface program that provides access to the mailbox server and is a prototype of the program that would run on a portable device. The above two user interfaces are similar in several respects: Each must have knowledge of the fact that the other exists, and must deal with the mailbox server in a consistent way. Also, both will present, to the extent possible, the same set of user commands for the display and manipulation of mail. However, the program intended for operation of the portable device will contain a special protocol for communication with the mailbox server, which is intended to operate properly in the face of repeated and constant communications failures.

The preliminary implementation of this portable version of the user interface will be carried out during 1983 on a modest-size desktop personal computer, such as the IBM Personal Computer. When this software has been demonstrated to work well in this environment, we will move the portable version of the interface software onto a portable device, which will operate in a mode where it is normally partitioned from the mailbox server. Under this configuration, we expect that the user will connect his portable device to the mail system once or twice a day to allow the version of the mail database in the portable machine to be brought up to date, and to transfer back to the mailbox server any requests that the user has made for the disposition of mail that has been already read.

The next phase of this project, which will occur after we have put a prototype mail system into operation, will be to identify an additional application for which we can devise a suitable human interface via the portable device. One possible application, which would provide a different view of partitioned data, would be a calendar management system such as the one described in Section 12.

The timing of projects that involve additional applications for portable devices is dependent on our initial experience with the mail application. It is our intention that by the end of 1985 we will be in a position to make substantive general statements about the partitioned database problem and about the use of portable computers as serious tools in a distributed computing environment.

6.5. RELEVANCE TO DARPA/DOD AND TECHNOLOGY TRANSFER

These three projects are relevant to DARPA in a number of ways: the exploration of new network technologies besides its inherent importance provides valuable feedback on protocol design. In particular, communication over the proposed paths will be achieved using the Internet Protocol family. This will provide valuable information about the applicability and performance of these protocols in a broad set of new circumstances. Asymmetry in the communications paths, for example in route or bandwidth, will stress the protocols in new ways. We expect that these protocols will work well in this new context, thereby supporting their generality claims.

The study of inter-enterprise communication will explore some low-level (per packet) access control mechanisms, as well as some application level mechanisms, that are of general relevance to the Internet project, where control of what traffic goes where is a current problem of high priority.

Our Laboratory is in an excellent position to effect the transfer of relevant results to DARPA, through the role of Dr. D. Clark, who is Principal Research Scientist at MIT/LCS, as protocol architect for the DARPA Internet project.

7. DISTRIBUTED SYSTEMS TESTBED

7.1. INTRODUCTION AND OBJECTIVES

During the past two years, we have investigated certain techniques suitable for the construction and integration of distributed systems. These techniques include an effective set of building blocks that comprise subsystems common to many distributed applications, such as the SWALLOW distributed storage system, tools useful in the construction and maintenance of applications such as debugging programs for a distributed environment, and communications architectures that make possible the construction of applications in a decentralized manner through the effective and efficient use of high bandwidth communications technology, such as local area networks, and in the future, wide bandwidth satellite communications.

We propose to fuse several threads of research in these areas to produce an advanced environment or testbed for these distributed systems ideas. This testbed will consist of new high technology network capabilities, such as local area networks, wide band packet satellite communications, advanced personal computer nodes, and specialized server nodes for the storage of data, authentication, message transport, and name resolution.

The evolving new computer technologies, such as high performance workstations with interactive displays and new communications technologies that emphasize high bandwidth communication at very low cost, offer significant potential advantages for the construction of new classes of applications. These advantages include more effective resource utilization, increased availability of computation to the user on demand, simpler protection models, and natural extensibility. Achieving these advantages through use of these new hardware technologies calls for the construction of systems in a completely new way: The availability of computational cycles close to the user make it possible to construct more highly interactive applications, and the availability of high bandwidth communications technologies make it possible for the computer to mediate access between more effective data storage and communications than was possible over low bandwidth telephone lines. However, as a direct result of these technological improvements, the system and the applications built on it face a much more dynamic environment -- including concurrency, complex failures, changing system hardware configuration and availability, on-line installation of software, and changing protection constraints -- than is normally found in a centralized application of the traditional kind. These considerations make construction of distributed applications qualitatively different than construction of centralized applications. Furthermore, the availability of substantially higher bandwidth network resources calls for completely different kinds of network protocols for full utilization of network bandwidths upon demand by the application. To fully realize the potential of the new hardware technologies, and the new computing configurations made possible by distributed, decentralized computing, a new generation of systems software and protocols will be needed. Accordingly, our objective in this research is to construct a prototype of this new generation system, evaluate its effectiveness, and transfer the results of these investigations into longer term projects such as the ARGUS language research described in Section 8 of this proposal.

7.2. BACKGROUND AND TECHNOLOGICAL NEED

Our proposed effort can be divided into four areas: (1) a coherent distributed system operating system; (2) a decentralized data storage system (SWALLOW); (3) exploitation of high bandwidth networks; and (4) protocol design and implementation for high performance. Though divided into these four areas, our work is unified, in the sense that we expect to produce a single distributed system prototype that combines these ideas into one coherent system. The following subsections discuss the background for these four areas respectively.

A High Performance Kernel System

Our earlier research has been concerned with the effective implementation of communications protocols, the effective implementation of a distributed data storage architecture described in the next section that allows the construction of object-oriented data structures and stable storage of these data structures, and the implementation of distributed systems services provided by specialized server nodes. One result of this research was the realization that most traditional operating systems did not efficiently support the techniques needed to implement these designs.

Communications protocols based on packet communication do not effectively map into traditional operating system approaches to I/O, which are based on stream-oriented connections, and which, except in the case of disks, work best with low bandwidth devices such as 10-50 Kbit/second serial communication lines.

The SWALLOW object-oriented approach to data storage conflicts with a traditional operating system notion of a file, a large clumsy object with no standard data representation and no way to link information together between files.

Furthermore, the use of input devices such as mice, joysticks and pointing displays, and interactive output devices such as bitmapped black and white or color monitors directly attached to the memory of the processor, are not adequately supported by the stream view of I/O devices.

Finally, extensive use of servers located across a network for such tasks as name resolution, authentication, data storage, and route selection also taxes the stream I/O architecture available to the operating system.

Our common experience in these areas suggests that a new approach to operating system design, oriented around fast processes sharing an address space, packet-oriented communications, and data abstraction, would be more effective in supporting our objectives.

Prior work that we intend to build on includes certain aspects of the Xerox PARC Alto operating system [7.1], the Mesa runtime system [7.2], and our Laboratory's TRIX 1.0 operating system [7.3] and the Cambridge University work on distributed systems.

Distributed Data Storage

For the past two years we have been working on a new approach to managing decentralized data. We have embodied this approach in the prototype system called SWALLOW summarized next.

In many applications where the data involved is naturally decentralized, there is still a strong need for coordination among the nodes. This coordination is needed, for example, to control concurrent transactions involving data managed by multiple nodes, and to orchestrate recovery from failures that affect such transactions. In addition, aggregates of interconnected advanced nodes exhibit a need for shared data storage servers that provide reliable coordination and long term storage of information.

Although there have been many suggested mechanisms for coordinating synchronization and recovery in a decentralized environment, these mechanisms usually apply only to very rigid structured databases, and often take advantage of the fact that the entire data configuration is known at the time that the application is constructed.

The SWALLOW system takes the approach that all stably stored data in the system is managed so that coordination of synchronization and recovery can be done. The basic approach proposed by Reed [7.4] involves a technique for construction of atomic actions on stable storage in a distributed system that minimizes implementation information in the interface between different connected systems. Each system knows as little as possible about the data stored on other systems. In particular, each system is free to physically organize its data in whatever manner seems appropriate, subject only to the constraint that it provide, as an external interface, a set of actions that satisfy Reed's model.

Our work on SWALLOW led us to the realization that, at least for the near term, the economics of secondary storage devices make feasible reliable, low-cost, long term storage as a shared resource. This led to the concept of *repository* nodes, i.e., special nodes whose job it is to provide reliable, low-cost, long term storage of data.

The SWALLOW data storage system was explicitly designed to support data structures consisting of linked collections of small objects. In order to support such data structures efficiently, an architecture based on "append-only storage" was developed, which had the side benefit of working well on optical storage media, such as write-once videodisk. This append-only storage model also works well with incremental garbage collection algorithms that have been proposed for storage management [7.5] [7.6].

High Bandwidth Networks

We believe that in a few years almost all data communications will be based on high bandwidth packet networks. Major components of such systems will be local area networks, such as Ethernet or the MIT ringnet, long distance satellite networks such as the DARPA Wideband Experimental Packet Satellite, and high performance store and forward networks composed of linked connections of these and possibly other technologies. A common thread that unites these network technologies is that bandwidth is relatively inexpensive. Work to date on data communications architectures and protocols have assumed that communication consists of long hold time connections with relatively constant low bandwidth data communications (the so-called telephone line model). We believe that computer-to-computer communications are driving the network design space to a different model: Most computer to computer communications are characterized by short bursts of very high activity between a pair of nodes followed by long periods in which there is no communication between that pair of nodes. Consider, for example, the transferring of a file from one computer to another with minimal delay. Optimizing such a transfer requires that the file be transmitted as fast as possible over the network and stored at the receiver. Once the file has been transferred, there is no longer any need for communication between the sender and receiver. Unlike a telephone call, the sender-receiver connection does not exist for a fixed duration with low, but constant, bandwidth; instead, the connection's duration drops as the usable bandwidth increases. Accordingly, efficient file transfers require a method for dynamically allocating large bandwidths to a sender, for a short duration. Even more extreme are the remote procedure call protocols, such as those proposed by Xerox [7.7].

The new high bandwidth technology for communications also offers the opportunity to construct systems that trade bandwidth for protocol simplicity. In the case of local area networks, for example, it is often easier to buy more bandwidth at the hardware level than to optimize the application (for example, its flow control methods) around bandwidth limitations in the communications system. We believe that it is important to investigate this design space trade-off more thoroughly.

In the development of this new communications technology, there are still some unanswered questions: Bridges or gateways that can forward packets rapidly from a local area network to a local area network or from a local area network to a satellite are still not capable of very high transfer rates of packet forwarding, being typically limited to transferring only one or two hundred packets per second. To achieve better transfer rates requires a rethinking of the architecture of the gateways, optimizing the performance of the software, and possibly changing the architecture of the machines that serve as gateways. In such a context, a desirable goal is a gateway that can handle the maximum data rates of all attached networks simultaneously.

Protocol Effectiveness

Use of high bandwidth networks is often impeded by bottlenecks in the host's ability to access the network. Protocol implementations on systems such as UNIX [7.8], are quite limited in the data transfer rates that they can achieve. Typical experimental data for file transfers on a local area network with 10 megabits per second aggregate bandwidth shows that the hosts are limited to about 50 to 100 Kbits per second. Thus, the "host bottleneck" seems to be about two to three orders of magnitude away from the speeds offered by the underlying communications technology.

Effective exploitation of the new hardware communications technologies and high bandwidth networks requires a new way of thinking about protocol implementation and protocol design. We identify three areas along which this problem needs to be attacked: First, the protocol implementation techniques must be refined so that protocol implementations are small in number of lines of code, and fast, by optimization of their algorithms. Second, efficient means of managing I/O devices (traditionally mediated by operating systems) must be devised to cope with these high speeds. Third, new protocols have to be designed that take advantage of the end-to-end application needs for which the protocol is being used. The first area, *effective implementation techniques* is a continuation of our past research on protocol effectiveness. The second, *efficient kernel I/O*, is a goal of the high performance kernel mentioned above. The third area, *design of new protocols*, is based on our earlier work in thinking about protocol design, particularly the so-called "end-to-end argument" [7.9], and the design of protocols for accessing various servers.

7.3. ACCOMPLISHMENTS

The work proposed here represents a continuation and integration of past efforts in the network area and in building blocks for distributed systems. Over the last two years our progress in these areas has been as follows:

The SWALLOW repository was designed and constructed and is now operational on a specially modified Alto computer with 300 megabit disk drive. We have demonstrated the ability of the repository to perform multi-site atomic actions in a test configuration. The repository also demonstrates our append-only storage model in that the design can be directly transported to a system with write-once optical disk instead of magnetic disk. We have also done some performance evaluations on this repository which have led to its redesign and reimplementation [7.10].

Another component of the SWALLOW system, the broker, which is a software module on each client node that provides the interface to data object storage, has been designed and is ready to be implemented on the single-user VAX workstations acquired in July 1982. The design of this broker incorporates a number of novel features, in particular, an approach to caching the contents of stable storage at the local system that works correctly with the atomic action implementation so that stable storage is nearly as efficient as volatile storage. This local caching design is an extension of the append-only storage model used for the repository. Another innovation in the broker design involves structuring the access to memory so that the transfer of updates to stable storage can proceed concurrently with the execution of the atomic action, thereby eliminating much of the delay associated with classical "two phase commit" protocols [7.11].

Over the past two years, our understanding of how to build effective protocols has materially increased as a result of our studies and experiments. Our first project was a careful analysis of the implementation of IP and TCP on MULTICS. This study made quite clear that the encountered performance problems were not due to protocol structure but rather to scheduling overhead in the operating system. Furthermore, we discovered that the complexity of the code was not primarily due to the details of the protocol, but to the client interface which each protocol was required to provide to the layers above it. The next phase of our study involved the implementation and evaluation and IP on the Xerox Alto computer. The operating system provided for the Alto is entirely different from that on MULTICS, and was specifically designed to support network protocols. Experiments in this environment conclusively proved that given the right support, protocol implementation can be done in an efficient and compact manner. Following the conclusion that we reached in the MULTICS case (that the client interface of a protocol layer could well be the most complex part of the implementation) we carried out an implementation which was specialized through one particular client, the user telnet protocol. The resulting implementation of TCP was substantially less complicated than any others with which we are familiar, consisting of less than 200 lines of source code which compiled into approximately 1,000 machine instructions.

This implementation led us to take a closer look at the question of how TCP interfaces to its clients. As normally conceived, TCP can have a number of clients, with markedly different requirements for throughput and delay. The requirement for dealing with these clients leads to a complex and general buffering strategy which we felt could be sidestepped through a different approach to modularizing protocols. A first experiment in this area was done with the Tripos system, and a different family of protocols in use at the Computer Laboratory in Cambridge, England. The resulting protocol implementation, in comparison to the one that had been previously done for that system, ran ten times as fast and was approximately one-fourth in length. Our success with this implementation suggested a final experiment, which was to use the structure that we had devised for the Tripos programs for implementing a new version of TCP and IP for UNIX. UNIX is a somewhat more complicated operating system than Tripos, and TCP and IP are somewhat more complicated than the protocols used in the Tripos experiment. For this reason, a final test involving UNIX is needed to provide a more convincing demonstration of our ideas. The design for this UNIX system is essentially complete, and its implementation should be completed by the end of 1982.

These studies yielded insights which are directly relevant to the general implementation of TCP and IP. Accordingly, we are proceeding with a report of these ideas to the internet community.

Another aspect of this project which we call "effective protocols," addresses the question of designing new sorts of protocols which are capable of intrinsically higher performance under certain specific circumstances. These protocols are based on the general idea that by making them to some degree application dependent, the protocol level requirements for reliable sequence delivery of the data can be relaxed. The resulting protocols are simple, and run at high speed. Furthermore, because they are simple, the burden of implementing them is not substantial. Examples of these protocols include the SWALLOW message protocol, a high-speed file transfer protocol called "Blast," and a remote bit map terminal update protocol called "Blink." These protocols have established that the bottleneck to performance is not communications related but rather rests on the applications' ability to generate and consume data.

These two projects, effective implementation of protocols and the design of effective protocols give us great confidence that we can indeed build protocols with the performance required for the system that we propose to build.

Several projects have contributed to our knowledge of efficient operating system design. The TRIX system, developed by the Real Time Systems Group, suggested an approach to modularity which captures the structure needed for cooperating asynchronous tasks. In this structure the ideas of process and address space are decoupled, so that several processes, each with its own stack, can share an address space in which the state variables of the computation are stored.

By decoupling address space from process, a substantial step is taken in reducing the required overhead of task switching. Efficient task switching permits new sorts of program structures, so to experiment with how such programs might look, we developed three different task management packages to try, in BCPL for Tripos, in C for UNIX, and in CLU. Programming with the aid of these packages has given us considerable confidence that our ideas for our operating system can be made to work.

7.4. PROPOSED EFFORT AND MILESTONES

The overall goal of our effort over the next three years is to produce a unified system, SWIFT, based on four converging thrusts discussed below. Although each of these efforts could be pursued on its own, much of the value of these approaches is derived from interaction with other efforts. For example, to effectively utilize the high bandwidth network and high bandwidth gateway technology that we intend to develop we will have to develop new protocols based on our work on effective application-oriented protocols and protocol effectiveness. Similarly, the SWALLOW distributed data storage system also requires effective protocols and naturally fits into the high performance kernel.

High Bandwidth Networks

Our primary goal in this area is to develop the interconnection technology needed to make possible data transfers at peak rates of greater than one megabit per second between any pair of hosts. Since the network under discussion will be composed of local area networks such as the Xerox Ethernet [7.12] and the MIT RingNet and longer distance links, such as the expected DARPA Wideband Experimental Packet Satellite, and the cable experiment (of Section 6), we need gateways that connect these networks together and that are capable of transmitting data at an aggregate bandwidth *equal to the sum of the attached networks*. Designing these gateways is both a hardware task and a software task. In 1983 we intend to study the problem of implementing small but fast gateways out of available hardware components. To this end, we would like to determine bottlenecks in our current gateway design that prevent us from achieving high gateway bandwidth. We believe that a change in the mechanism used for routing and other network interconnection functions will be needed. We intend to investigate an alternative approach by implementing source routing as proposed by Saltzer, Clark, and Reed, [6.8] as a way of implementing much higher bandwidth packet switching in gateways. Tò do source routing, we need to construct source routing gateways, but more importantly we need to construct a source routing service consisting of replicated route servers throughout our local network testbed. By the end of 1982, we expect to have constructed a network based on source routing that can be used for performance experiments. This network testbed may also be used for experimenting with our inter-enterprise communication research discussed in Section 6.

Protocol Effectiveness

We expect that most of the work on protocol effectiveness will be complete by early 1983. As a further test of our theories, we intend to carry out an implementation of TCP for the SWIFT kernel. This work should confirm that the kernel is properly structured to provide the right support for this sort of protocol.

We expect further that most of our proposed effort will be oriented toward specialized protocols, running on top of IP, for such purposes as support of the SWALLOW system.

Finally, we expect that, the SWIFT project will continue to teach us new considerations that relate to good protocol design.

SWALLOW Distributed Data Storage System

Our main efforts on SWALLOW are aimed at making the system a reliable service, and at exploring the construction of applications that use SWALLOW. During 1983 we plan to focus on two activities. The first involves integration of the SWALLOW broker into the SWIFT operating system kernel. Our intention here is that SWALLOW will manage all of the storage in the SWIFT system. Since SWALLOW is a low-level storage manager, providing objects but neither files nor naming, we will need to construct a variety of more traditional storage services on top of SWALLOW. Primary among these is the storage of text files, and naming contexts (directories). The SWALLOW broker, then, would become a central storage component of the SWIFT distributed environment.

We expect that the workstations in the SWIFT environment will not have stable local storage. To provide stable storage in the SWIFT environment, we will use the SWALLOW repository as the stable storage repository. We must take the current SWALLOW repository designed for the Alto and transport it to a more appropriate host on our local network environment. Such a host will probably be a VAX with additional data storage, and possibly optical storage if such storage devices become commercially available during the term of our proposed research.

By the end of 1983, we expect that a version of the SWALLOW broker will have been constructed in CLU as part of the SWIFT kernel operating system, and that a version of the SWALLOW repository will be running on a VAX, having been rewritten in CLU. During 1984 we expect to test the performance of the SWALLOW environment.

Also during 1984, we expect to begin development of applications, such as a distributed program library system, or other sorts of distributed database systems, based on the SWALLOW approach. As these applications are developed, we expect to learn about the effectiveness of the SWALLOW object management approach by observing the performance of these applications. By early 1985, we expect simple applications to exist and more complex ones to be underway. As a result of experience with these applications, and the performance measurement activity we are likely to devote a portion of 1985 to the tuning and reworking of SWALLOW.

We expect that during the next three years commercial high density writable optical storage media will become available. Since our implementation of SWALLOW was developed with the goal of storing data on these media, we hope to be able to try out these ideas on experimentally optical media. Our best estimate of when this will happen is sometime in 1984. Scheduling for such experiments will, by necessity, be dictated by availability of these devices, and our ability to purchase them.

The SWIFT High Performance Kernel and Servers

Our major goal in developing the SWIFT high performance kernel and supporting servers is to unify several different projects into a coherent testbed. In order to do this, we will develop a new operating system kernel with the primary goals of supporting the object model for storage assumed by SWALLOW, the high performance access to I/O devices, and the low-cost processes required for effective protocol implementation and highly interactive applications.

Our first goal is to produce a design for a simple operating system kernel for the CLU language, which provides for extremely cheap processes that share the same address space and object-oriented SWALLOW storage. We choose to base this operating system on the CLU language for several reasons. First, we have developed a transportable CLU implementation. Second, this CLU implementation is easily altered to provide compile time support for certain of the key ideas that we intend to incorporate within the SWIFT system. Third, CLU's data abstractions provide a very convenient way to define the interfaces of our operating system as it is built. We recognize that by choosing a high-level language such as CLU, rather than a language such as C or Bliss, we are going against certain traditions in operating system construction. It has often been argued that it is impossible to construct appropriately efficient code in high-level languages. In fact, we have found that CLU is amenable to construction of very efficient programs, and more importantly, that the data abstraction and storage management facilities provided by CLU should eliminate many of the headaches often encountered by operating system implementation projects.

In this project, our goal is to spend six months on the kernel design, followed by a six-month implementation effort resulting in a demonstrable but somewhat simplified system by the end of 1983, roughly at the same time that we expect the SWALLOW broker to be usable. This work will be carried out on the single-user VAX, augmented by a number of additional workstation machines as the project proceeds. We expect that five to ten machines will be needed as workstations by our researchers as the development proceeds.

During 1984, we expect to begin construction of applications based on the SWIFT operating system kernel and the network level services by such servers as the SWALLOW repository, name look-up servers, authentication servers, route servers, and debugging servers. Such applications will include the distributed program library mentioned above, and other Laboratory applications such as calendar and message systems, and personal database systems.

The ARGUS language [7.13] described in Section 8 will initially be implemented on top of UNIX on the VAXs. It is clear that the long term effectiveness of the ARGUS implementation depends on efficient implementations of many of the functions provided by the SWIFT system. Accordingly, we expect that during 1984 we will explore merging the SWIFT project with the ARGUS project. The likely outcome of this exploration will be an effort during 1985 to build a system that supports the ARGUS language directly, using the implementation ideas explored in SWIFT. To this end, we expect to track closely and interact actively with the ARGUS research effort.

Computing and Network Support Needs

Our ability to do the proposed research depends on the availability of a critical mass of computing resources dedicated to this development. During early 1983 the four in house single-user VAX workstations should be adequate. As implementation proceeds in the latter part of 1983, three to four additional VAXes will be needed as servers, and another three to four as workstations. In 1984 and 1985 we foresee a need for ten to fifteen workstations to develop and test applications. These workstations will be SUVAXes or Nu's. Network resources needed for this research include network attachments for all participating machines, LSI-11's for gateways, and a link to the Wideband Experimental Packet Satellite.

In addition, we plan to use several large capacity secondary storage devices for our SWALLOW work which will be either large magnetic disk drives, or write-once optical disk drives.

7.5. RELEVANCE TO DARPA/DOD AND TECHNOLOGY TRANSFER

Within the DOD, decentralized systems have a natural application due to the decentralized nature of the organization and its special data and computing requirements. New hardware advances such as the development of powerful advanced computing nodes, high performance local area networks, and high bandwidth packet networks have all enhanced the opportunity and need for such decentralization.

Yet the capability to construct decentralized systems has not kept pace with these hardware advances. Since there are many applications within the DOD that require or will require decentralized solutions, it is important that tools and subsystems be developed that aid in the construction of such applications. The coherent distributed testbed that we propose here will provide an opportunity to evaluate a number of ideas that have been collected in distributed systems research at MIT so far. We expect that this system, will contribute to the understanding of how such decentralized systems can be constructed to be efficient, failure resistant, and expandable.

As in the case of Section 6, the components of this work associated with network protocol investigations, in particular the high bandwidth networks, high performance gateways, and effective protocol design efforts will have a direct influence on the DARPA internet project through Dr. Clark, who is serving as a technical coordinator to the DARPA internet experiments. The development of these ideas at MIT allow for exploration of approaches that are too risky to attempt directly upon the internet project.

The SWALLOW, SWIFT, and effective protocol design development efforts will most likely influence the DOD by serving as a model for future decentralized system architectures that will be built to serve DOD's specific needs.

11. THE COMMUNITY INFORMATION SYSTEM

11.1. INTRODUCTION AND OBJECTIVES

The primary goal of this research is to develop new ideas and techniques that will allow people to communicate in qualitatively better ways with each other and with information resources of common interest. These approaches will be implemented and tested within a single experimental framework -- a community information system. The information system that we propose will provide members of our immediate research community with information of common interest (e.g., news and local events) and with inter-communication capability.

This system has certain characteristics that are important to our research. First, it models within our own specific community many of the inter-communication opportunities that are present in future communities of people and machines both within and outside of the DOD. Second, it entails systems research which is motivated by an application that sets people-to-people and people-to-machine inter-communication as its principal goal -- this represents a research direction that differs from and complements our systems-to-applications direction taken in Part B of the proposal. In effect, this work may be regarded as focusing on a "thin" vertical slice that cuts across our research activities in advanced nodes, networks, and integrated language/systems -- in that sense it is expected to benefit from the results of that work and to provide additional valuable experience in the opposite direction. Third, in accordance with our policy to minimize sequential inter-dependence of different LCS research projects, the community information system project will begin with a modest scale system based on available hardware and software and on a few of the well developed technologies discussed in Part B of the proposal. As the proposed project evolves and as our other research projects mature it will use progressively more of their resultant systems and techniques.

Our plan is to start by building a modest base system that deals with the problems of information collection, communication, and consumption within a limited data domain. Thus, once this base is established, we will be able to expand it to deal with a larger data domain, and we will be able to use it to test our ideas on how information systems should be organized to improve the quality of communication. We plan to use the community information system application to pursue basic research in four major areas: user interfaces, databases, data communications, and programming languages as discussed next.

1. The user interface research will focus on techniques for displaying, updating, browsing, and querying databases. We will start with existing bit-map color display technology, and use a mouse as a pointing device.

With this hardware base, our emphasis will be on presenting a friendly and reactive user interface that is heavily based on graphics. For example, we would like to allow users to formulate queries with icons, to point at maps to indicate areas that they are interested in learning more about, or perhaps to use a combination of graphical techniques to specify their interest profiles. An important goal here is to make database technology accessible to the casual non-professional user for both data entry and consumption.

2. The data model research will take the perspective that one can not have arbitrarily large amounts of knowledge in one computer, and thus an information system must be composed of hundreds, or thousands, of databases that are integrated with one another. This view has served us well in traditional media where we maintain a personal library of books and papers that are full of references to one another. Starting with relational database technology, we will explore entity-value-relationship data models [11.1], methods of establishing and maintaining inter-database references, and ways of collecting high quality information for a database system with forms-fill-out terminals.

3. The programming language research will develop a simple interpretive language for communication. Programs written in this language will be transmitted to users and stored in their databases. Thus, a user's query could result in a running program. The program might pop new buttons up on the screen, present an intelligent form to the user, or perform transformations on the user's database. The intent here is to make the medium as useful as possible by transmitting data in a way that not only informs users, but actively interacts with them as well. One aspect of this research will examine how such programs might be created by naive users.

4. The data communication research will explore the use of a mixture of broadcast and duplex communication channels in an inter-network. Our goal is to be able to use both types of channels for a single service in a transparent way. The nodes of this inter-network will be single-user computers which will communicate with one another and with information services.

11.2. BACKGROUND AND TECHNOLOGICAL NEED

As many successful systems have demonstrated, computers are well adapted to providing people with valuable information. However, most people are already deluged with more information than they can possibly assimilate. Our experience with electronic mail and information services leads us to believe that straightforward applications of technology may only exacerbate the information overload placed on people.

This research proposes to integrate a unique set of technologies: low-cost personal computers with sophisticated graphics capabilities, contemporary ideas about database systems, and mass digital communication. This set of technologies will allow us to explore a new type of system architecture. Providing users with a personal computer will allow us to build a personal information assistant to help them cope with the information they consume every day. In addition to moving information from place to place, the assistant will rearrange, filter, remember, index, track, and process information according to the interests and needs of its owner [11.2]. With such a personal information assistant, the cost of many common information related tasks will be reduced to the point that a large change will take place in what one person can accomplish. For example, it is much easier for a computer to listen to digital broadcasts and compare items with one's interest profile than it is for one to attempt to read the broadcast data (e.g., news) every day. A personal computer can also keep track of other kinds of data such as the current state of the public transportation system -- busses, subways, freeways, and so on -- that can help community members in their daily life.

Perhaps the most exciting opportunity before us is that once in digital form, information can be heavily cross indexed. Thus, in addition to telling us about an event, the computer could tell us about the sponsoring organization or a method of reaching the event, based on the current state of the public transit system. This cross indexing, or information integration, allows dynamic information such as news and events to contain references to static information such as maps and background material.

The technology we plan to use for our mass digital communication presents another interesting set of possibilities. The broadcast channel will be used to "trickle-change" the databases in personal information assistants by sending new information and programs to users. There can, of course, be an unlimited number of users that listen to broadcast information services. Initially we will use a dial-up packet switch to allow users to access information that was not broadcast or to allow them to originate information.

11.3. ACCOMPLISHMENTS

The following work has already been completed to lay the foundation for this project:

User Interface Prototype. A prototype of a user interface that might be used for our system was completed. The prototype was built using the Cedar programming environment at the Xerox Palo Alto Research Center. The prototype was based on an entity-value-relationship database system and a window oriented browser. Scanned images, such as pictures and maps, were added to the database system for our prototype. This allowed one to browse through city maps, the pictures of people in a group, and so on. A videotape of the prototype was made and shown to several research groups.

Industrial Support. We plan to use commercially available hardware and software wherever possible in our work, and thus it is important for us to establish working relationships with computer companies. Effective contacts have already been made with research groups at Apple Computer, Atari Computer, Digital Equipment Corporation, and the Xerox Palo Alto Research Center.

Information Suppliers. To test our information management system we plan to obtain information every day from outside information sources. To this end, we are currently negotiating with newspapers and wire services.

11.4. PROPOSED EFFORT AND MILESTONES

The project will test a set of ideas drawn from our four research areas by operating a community information system in the Boston metropolitan area. We will begin by constructing the base system starting in January 1983. By 1985 we plan to have a prototype system running, and by 1986 plan to have a complete demonstration system operational. Below we have included a list of project milestones with scheduled completion dates.

The base system will provide comprehensive international news and information on events in the Boston area. In addition, it will index the documents and messages of community members. The base system will be vertically integrated, in the sense that it will deal with collection, communication, and consumption of information, as opposed to only one of these aspects. Thus many problems will be explored within the scope of the base system, with the expectation that experience with the system will allow us to generalize its solutions. The base system has been chosen to be sufficiently small to enable the project to rapidly build critical mass.

Figure 11-1 is a block diagram of the base system. External information sources will forward information to MIT through dedicated communication channels as shown. Information will also be obtained from community members operating at single-user workstations. Once collected, information will be tagged and put into a database system at MIT. The database system will provide basic storage and indexing services for information such as documents, mail, and news. Starting with

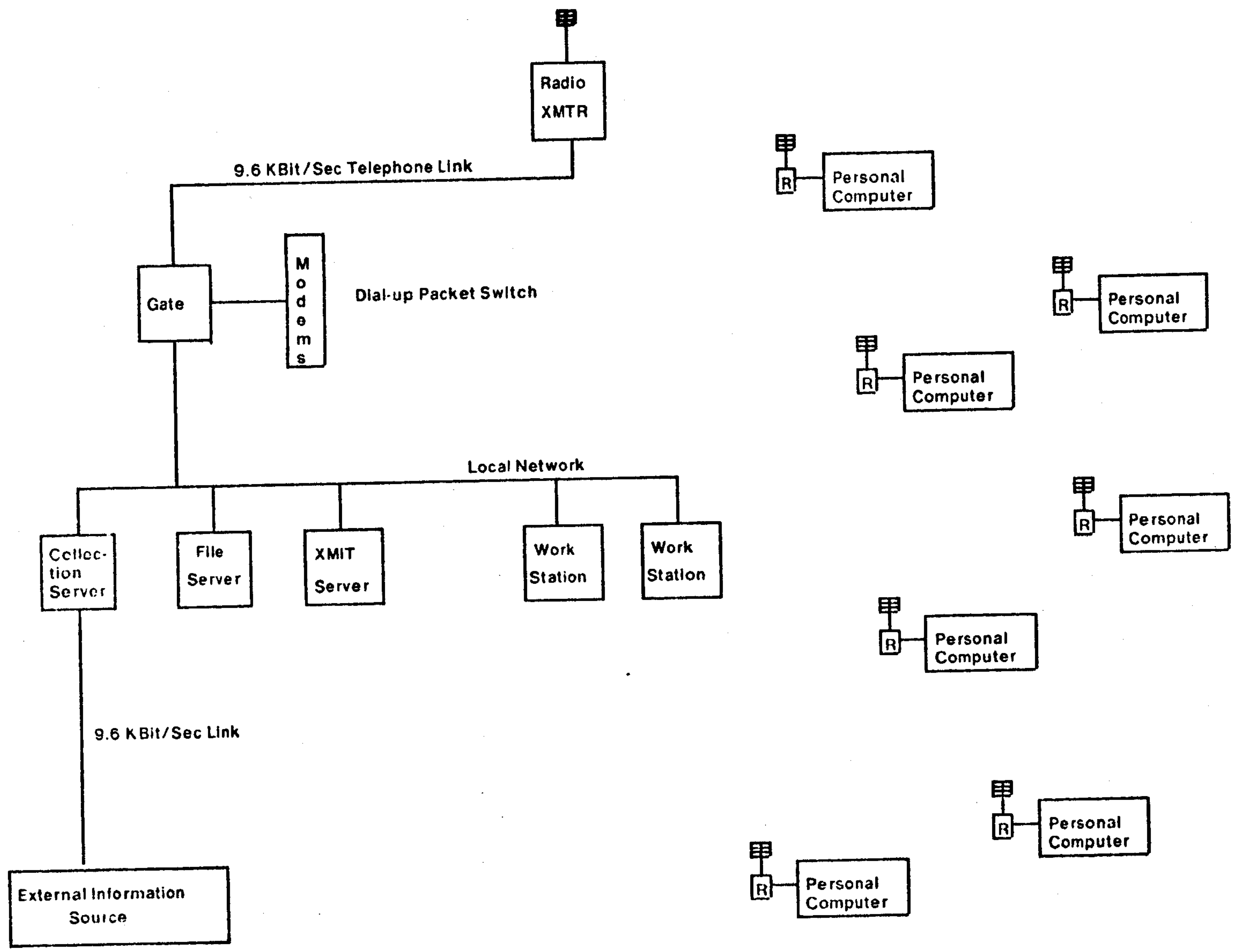

Figure 11-1: Base System Block Diagram

well understood database technology, we plan to work on inter-database references to facilitate information integration, free text searches for information, and methods of collecting and automatically indexing information that is provided by outside sources.

Information in the database will be communicated to personal computers via continuous digital broadcast [11.3] (e.g., the MIT FM station and the Newton cable system; see also Section 6) and a dial-up packet switch. In addition to broadcasting the full text of event descriptions, references will also be transmitted. References are intended to be presented to the MIT system via the dial-up communication channels to obtain further information, or to acquire missed broadcast information. The dial-up packet switch will use simple data compression techniques to improve the capacity of telephone lines by doing such things as compressing packet headers when possible.

Drawing from digital broadcasts and direct connection with the MIT packet switch, personal computers will maintain local databases that reflect their owner's interests. In addition to being able to browse and query both local and remote data bases, a user will be able to maintain standing orders that certain types of information should be brought to his attention. Because these personal computers will not always be connected to the system at MIT we will examine the problems of self-contained databases that can be disconnected from a network and later reconnected.

Our user interface research will use the database system as its foundation to create an environment where it is simple to experiment with new ideas. To create a new user interface one will write a program that uses our standard display manager and database system. Because the details of data storage and display management will be hidden, these programs will be relatively short and simple, allowing us to try many different information presentation techniques. Some of the user interface approaches we plan to experiment with include query by example, spatial queries (for example, indicating an area of interest by pointing to it on a map), using icons to compose queries, and facilities for allowing a user to refine a request until he finds what he is looking for. As these approaches suggest, we are trying to reduce the amount of knowledge needed to operate a sophisticated database system to make the technology accessible to non-professional users.

The user interface software will be developed on two types of workstations. The first will be a professional workstation with sophisticated user interface software, such as a Symbolics Lisp Machine with a color display. This professional version would be practical for business use, and will be used to test advanced user interface concepts. The second type of workstation will be a personal computer, such as a Motorola 68000-based system, with somewhat less sophisticated user interface software, but with essentially the same database capabilities. This version would be used by selected LCS researchers at home.

To support this system there will be a variety of server machines at MIT. The gateway server will implement a dial-up TCP packet switch, and it will forward packets to the digital radio for broadcast. The collection server will receive information from outside sources, perform initial classification, and store the information in the database system. The file server will serve as the repository for the database. Finally, the transmit server will schedule selected new database entries and indices for broadcast.

The following is a list of major milestones:

January 1983:

1. Begin constructing digital radio plant;

2. Begin database design for personal information assistant;

3. Receive two professional workstations for user interface prototyping;

4. Begin prototyping user interfaces on professional workstation.

July 1983:

1. Receive personal computers with graphics support and window manager from commercial source. These computers are based on the Motorola 68000 microprocessor, and should retail in 1983 for a few thousand dollars;

2. Begin design of data communication language;

3. The dedicated data link to the information source is operational, and information is received and recorded daily.

January 1984:

1. The broadcast digital radio channel is completely integrated into the MIT TCP Inter-network, and it is possible to broadcast packets from MIT to the Cambridge area;

2. The dedicated data link to the information source is operational, and information is received and recorded daily;

3. Receive two additional professional workstations for data base system prototyping;

4. Database system fully specified. Begin implementing a prototype of database system on the professional workstations.

July 1984:

1. Prototype implementation of database system usable on professional workstation;

2. Begin design of database system for personal information assistant.

January 1985:

1. Begin regular digital broadcast of data. Data are broadcast in the data communication language;

2. Professional workstation able to browse database.

January 1986:

1. Personal information assistant is able to receive data from broadcast channel, store it in its database, and support window oriented browser.

11.5. RELEVANCE TO DARPA/DOD AND TECHNOLOGY TRANSFER

The problem of collecting and communicating real-time information over a large geographical area is directly applicable to many DOD situations. Our interest in getting information to the attention of appropriate people by using inexpensive hardware technology, straightforward user interfaces, information filters, and personal information assistants may also be of use in similar situations. For example, the SAFE project [11.4] is investigating related advanced information tools for the intelligence analyst. Moreover, once operational, the Community Information Project can be charged with different data and used in a DOD context, such as command and control, logistics, or intelligence.

The network concepts that we propose to explore are relevant to DOD needs. The idea of using broadcast radio technology to communicate with remote database systems will be applicable to certain types of command and control applications. The goal of utilizing high bandwidth outgoing channels (e.g., digital broadcast) and low bandwidth incoming channels (e.g., telephone lines) within an integrated packet network will be useful in many DOD situations.

Our work on a specific database application should provide insights on how to structure distributed information systems. Our work on a simple programming language for communication may allow these systems to be used by naive users for sophisticated tasks.

In addition, our application will motivate basic work on security. Other groups in the Laboratory are concerned with controlling access to inter-network resources such as a digital packet transmitter. We are interested in using cryptographic sealing [11.5] to secure our packet network and database systems.

We expect that technology transfer will occur in two ways: (1) through the reports and papers that our group will author; and (2) by direct transfer to other DARPA sites. An important precondition for technology transfer is that we demonstrate the capabilities of our system, and we plan to do this by using it in our daily work. Direct transfer of the system to another site would be straightforward since we plan to use commercially available hardware.

REFERENCES

6.1 Saltzer, J., Clark, D. and Pogran, K. "Why a Ring," *Proceedings of the 7th Data Communications Symposium*, Mexico City, October 1981, 211-217.

6.2 Saltzer, J. "Communication Ring Initialization without Central Control," MIT/LCS/TM-202, MIT Laboratory for Computer Science, Cambridge, MA, December 1981.

6.3 Bux, W., et al. "A Local-Area Communication Network Based on a Reliable Token-Ring System," *Local Computer Networks -- Proceedings of the IFIP TC 6 International In-Depth Symposium on Local Computer Networks*, P. Ravasio, G. Hopkins and N. Naffah (Eds.), Florence, Italy, April 1982, 69-82.

6.4 Muller, H., Keller, H. and Meyr, H. "Transmission in a Synchronous Token Ring," *Local Computer Networks -- Proceedings of the IFIP TC 6 International In-Depth Symposium on Local Computer Networks*, P. Ravasio, G. Hopkins and N. Naffah (Eds.), Florence, Italy, April 1982, 125-148.

6.5 Rudin, H. "Validation of a Token-Ring Protocol," *Local Computer Networks -- Proceedings of the IFIP TC 6 International In-Depth Symposium on Local Computer Networks*, P. Ravasio, G. Hopkins and N. Naffah (Eds.), Florence, Italy, April 1982, 373-388.

6.6 IBM Corporation, "Contribution of Working Papers to IEEE Project 802 on Local Area Area Networks, March 1982.

6.7 Corbató, F. "An MIT Campus Computer Network," Campus Computer Network Group Memo No. 1, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, 1981.

6.8 Saltzer, J., Reed, D. and Clark, D. "Source Routing for Campus-Wide Internet Transport," *Local Networks for Computer Communications*, A. West and P. Janson (Eds.), New York, North-Holland, 1981, 1-23

6.9 Singh, V. "The Design of a Routing Service for Campus-Wide Internet Transport," MIT/LCS/TR-270, MIT Laboratory for Computer Science, Cambridge, MA, August 1981.

6.10 Saltzer, J., et al. "Computer Systems and Communications," *MIT Laboratory for Computer Science Progress Report 18*, MIT Laboratory for Computer Science, Cambridge, MA, to appear.

6.11 Saltzer, J. "On Naming and Binding of Network Destinations," *Local Computer Networks -- Proceedings of the IFIP TC 6 International In-Depth Symposium on Local Computer Networks*, P. Ravasio, G. Hopkins and N. Naffah (Eds.), Florence, Italy, April 1982, 311-317.

7.1 Lampson, B. and Sproull, R. "An Open Operating System for a Single-User Machine," *Proceedings of the 7th Symposium on Operating System Principles*, Pacific Grove, CA, December 1979, 98-106

7.2 Lampson, B. and Redell, D. ""Experience with Processes and Monitors in Mesa," *Communications of the ACM*, (February 1980), 105-118.

7.3 Clark, D. "The TRIX 1.0 Operating System," *IEEE Distributed Processing Quarterly*, 1,2 (December 1981).

7.4 Reed, D. "Naming and Synchronization in a Decentralized Computer System," MIT/LCS/TR-205, MIT Laboratory for Computer Science, Cambridge, MA, September 1978.

7.5 Lieberman, H. and Hewitt, C. "A Real Time Garbage Collector that can Recover Temporary Storage Quickly," AI Memo 569, MIT Artificial Intelligence Laboratory, Cambridge, MA, April 1980.

7.6 White, J. "Memory Management in a Gigantic LISP Environment or GC Considered Harmful," Presented at the 1980 LISP Conference, Stanford, CA, August 1980.

7.7 Xerox NS Protocol Specification, Xerox Palo Alto Research Center, Palo Alto, CA.

7.8 Gurwitz, "VAX-UNIX Networking Support Project Implementation Description," IEN 168, Defense Advanced Research Projects Agency, Arlington, VA, January 1981.

7.9 Saltzer, J., Reed, D. and Clark, D. "End-to-End Arguments in System Design," *Proceedings of 2nd International Conference on Distributed Computing Systems*, Paris, France, April 1981.

7.10 Svobodova, L. "A Reliable Object-Oriented Repository for a Distributed Computer System," *Proceedings of the 8th ACM Symposium on Operating Systems Principles*, Pacific Grove, CA, December 1982, 47-58.

7.11 Gray, J. "Notes on Database Operating System," *Lecture Notes in Computer Science*, 60, New York, Springer-Verlag (1978), 393-481.

7.12 Metcalfe, R. and Boggs, D "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, 19,7 (July 1976), 395-403.

7.13 Liskov, et al. "ARGUS Reference Manual," MIT Laboratory for Computer Science, Cambridge, MA, to appear.

11.1 Chen, P. "The Entity-Relationship Approach to Logical Database Design," *ACM Transactions on Database Systems*, 1,1 (March 1976), 9-36.

11.2 Bush, V. "Memex Revisited," in *Science is Not Enough*, New York, William Morrow, 1967, 75-101.

11.3 Anderson, H. and Crane, R. "A Technique for Digital Information Broadcasting Using SCA Channels," *IEEE Transactions on Broadcastings*, BC-27,4 (December 1981), 65-70.

11.4 Landcaster, F. *Toward Paperless Information Systems*, New York, Academic Press, 1978, Chapter 3.

11.5 Gifford, D, "Cryptographic Sealing for Information Secrecy and Authentication," *Communications of the ACM*, 24,4 (April 1982), 274-286.