The Spectro-Microscopy Collaboratory: Design Overview of a Prototype Virtual Laboratory*

Imaging and Distributed Computing Group
Information and Computing Sciences Division
Lawrence Berkeley Laboratory , University of California
Berkeley, CA 94720

Project Participants:
Lawrence Berkeley Laboratory
Deb Agarwal, Keith Beattie,
Sonia Sachs
University of Wisconsin
Brian Tonner, Eli Rotenberg,
Jonathan Denlinger, Jian Zhang

Introduction

The Advanced Light Source (ALS) is a unique resource that lends itself especially well to a remote collaborative environment or virtual laboratory. The first of the ultrabright "third generation" synchrotron radiation facilities, it can serve hundreds of users who have a broad range of applications. These users come from all over the world at considerable trouble and expense, so a successful remote collaborative environment would be immediately useful. We would also expect considerable interest from users of the two dozen or so synchrotron radiation sources that are running or being built or planned worldwide.

One experiment that takes particular advantage of the characteristics of the ALS is the Spectro-Microscopy Facility on Beamline 7.0, which we are using as a testbed for a virtual laboratory. The amount and combination of information available will add up to remote access that goes much beyond than mere remote control, thus creating a virtual laboratory environment at the remote site. This project is one of four diverse testbeds in the Distributed, Collaboratory Experimental Environments (DCEE) program supported by the Department of Energy. The overall goals are to provide easier access for users and to foster more and better opportunities for collaboration. To understand why BL7 lends itself so well to a virtual laboratory, and the basis for the design decisions, it is helpful to briefly review the functions of the beamline and its experimental end stations.

The philosophy of the SpectroMicroscopy Project is to take full advantage of the first undulator beamline at the ALS. A 5-m-long, 5-cm-period undulator creates a small, narrow, intense cone of soft x-rays. The desired wavelength is selected with a spherical grating monochromator whose optics

*. This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098. This document is LBL-37293.

set new standards for surface finish and accuracy. The output of the beamline (by which we mean the undulator and monochromator) is refocused by adaptive grazing-incidence mirrors into a very small spot, less than 40 microns in diameter. This spot can be steered between two branch lines called MicroFOCUS I and MicroFOCUS II. At the end of one of these branches, the beam becomes the object point for two scanning X-ray microscopes downstream of the microfocus location. Zone-plate lenses demagnify the beam further, achieving spot sizes from 1000 angstroms down to, eventually, below 500 angstroms. These two zone-plate focal points are called the NanoFOCUS I and II stations.

Many issues must be addressed in order to remotely control such a sophisticated facility, and in order to ensure success with remote collaborative environments in general. Security of access is an important one; mechanisms for preventing unauthorized access and guarding proprietary data must be provided. Safety must also be addressed: not all aspects of the experiment can be safely operated remotely, and adequate safeguards are required on the rest. Since the collaboratory is designed to allow multiple remote researchers to participate in an experiment, data transfer mechanisms and arbitration mechanisms that determine which site is currently in control of the experiment need to be designed and built. Interaction between researchers at the remote sites and at the ALS will provide additional feedback regarding the current condition of the apparatust, and the goals of the experiment. As we address these issues, we must produce a user-friendly and easy-to-maintain set of tools.

The ALS Collaboratory project is organized in two phases. The first phase provides remote monitoring of experiments. along with a video conference shared among the researchers conducting an experiment. In this phase, network performance issues are evaluated, and bottlenecks are detected and addressed. Security issues are not critical in this phase because the monitoring facility consists only of data streams leaving the ALS Collaboratory. The second phase expands the monitoring capabilities to include actual ontrol of experiments from a remote site. In this second phase, security and coordination issues associated with providing access to the facility have to be addressed.

This document describes the individual components of the Collaboratory design. In Section 1.0 we provide some background about the ALS and the Spectro-Microscopy Facility. In Section 2.0 we outline the approach taken in the design of the Collaboratory facility. The network is discussed in Section 3.0. The data dissemination mechanisms are discussed in Section 4.0. In Section 5.0 we discuss the safety and security mechanisms, and issues related to experiment arbitration. Experiment control, experiment recording, and video conferencing are addressed in Section 6.0. Section 7.0 outlines the project status and Section 8.0 lists the anticipated future accomplishments.

1.0 Background

The Advanced Light Source is the brightest source of soft X-ray beams in the world today and the bellwether of a “third generation” of synchrotron radiation sources whose hallmarks include very-low-emittance electron beams and provisions for a large number of magnetic insertion devices (“wigglers” and “undulators”) to enhance production of photons. The SpectroMicroscopy Facility is based on an undulator light source coupled to a high-resolution spherical grating monochromator. Both the undulator and the monochromator are state-of-the-art machines, representing an investment of nearly \$4.0M. This undulator beamline provides the soft-X-ray photons for a suite of experimental analytical equipment that makes up the SpectroMicroscopy Facility; these analytical instruments themselves are state-of-the-art and represent another investment of nearly \$1.0M. The instruments of the SpectroMicroscopy Facility which are relevant to the current proposal are the UltraESCA Project, the atmospheric pressure scanning transmission X-ray microscope (STXM) and the ultra-high vacuum scanning photoemission microscope (SPEM).

These instruments are designed to provide spatially resolved chemical information at length scales ranging from below 1 micron to, in the case of photoelectron diffraction structural imaging, the atomic scale. Their capabilities are beyond what can be achieved anywhere else in the world, with the possible exception of the one or two other sites with similar soft-X-ray undulator beamlines. Because of the unique capabilities of these instruments, the SpectroMicroscopy Project was conceived from the outset as a rather large collaboration.[**] Nevertheless, the use of these machines currently involves a very substantial investment in training, staffing, time, and travel costs, so plainly the success of the Collaboratory would have far-reaching implications for users of the Spectro-Microscopy Facility, the ALS, and synchrotron radiation sources in general.

The SpectroMicroscopy Facility is primarily an analytical tool, which implies that the “product” is not a physical object, but rather, information in the form of images, spectra and sophisticated chemical and structural data. It is therefore very well matched to remote electronic communication using high speed computer networks. Through this project, the SpectroMicroscopy Facility can be opened up to users from a much wider range of organizations. A particular target audience consists of industrial users, such as those in the semiconductor industry, who can greatly benefit from this proposed electronic economy of scale.

Current network and videoconferencing technology can be used to provide the required type and degree of access. Internet-based videoconferencing and whiteboard applications, shared electronic publishing, and distributed system services provide the basis of the geographically distributed environment that will permit scientists to participate in the BL7 experiments from their home institutions as effectively as they do now by travelling to the ALS.

**. The SpectroMicroscopy Facility participating research team (PRT) consists of members from LBL (D. Attword, T. Warwick), Lawrence Livermore National Laboratory (J. Tobin), Stanford Synchrotron Radiation Laboratory (P. Pianetta), the University of Washington (M. Olmstead), North Carolina State University (H. Ade), the University of Oregon (S. Kevan), Pennsylvania State University (R. Willis, Jr.), the University of Michigan (J. Allen), the University of Wisconsin (B. Tonner) and the University of California, Berkeley (J. Bokor).

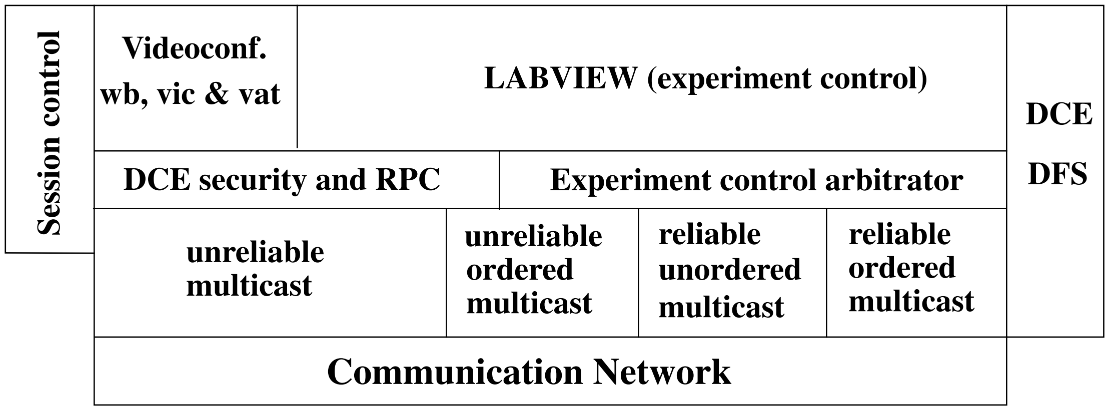

Figure 1 Diagram showing the components of the BL7 Collaboratory. The first phase will provide monitoring and conferencing; the second phase will progress to actual remote experimentation, which has implications for security and multi-client control protocols

2.0 Design Approach

As shown in Figure 2, the components of the ALS Collaboratory facility are organized into four layers of functionality. The bottom layer consists of the physical network, and associated data link, network controls supporting the ALS Collaboratory facility. This layer provides the mechanisms for the unicast and multicast transmission of messages between sites (Section 3.0). The next layer consists of data dissemination mechanisms (Section 4.0), which provides various levels of reliable and ordered transmission of multicast messages. It also provides mechanisms to track membership of the participants during a communication exchange. In order to resolve contention for the use of the ALS Collaboratory facility, and in order to offer safe and secure access to it, a layer containing the access control components is built on top of the data dissemination mechanisms (Section 5.0). This layer builds on the Distributed Computing Environment (DCE) developed by the Open Software Foundation (OSF). The top layer is composed of the video conferencing, remote experiment control, and data storage components (Section 6.0).

The first phase of the project will deliver the video conferencing, and a subcomponent of the remote experiment control, which allows viewing of the control actions and data taken at the ALS. This approach allows us to quickly introduce the equipment and software to the researchers, so that feedback and evaluation of the design will occur at an early stage of development. This approach also allows for the evaluation of network performance issues, and for the detection of bottlenecks. The second phase of the project will deliver the complete remote experiment control component, including the control from a remote site. This phase will also deliver the security and safety mechanisms required to provide access to the ALS over the network, as well as the experiment arbitration component.

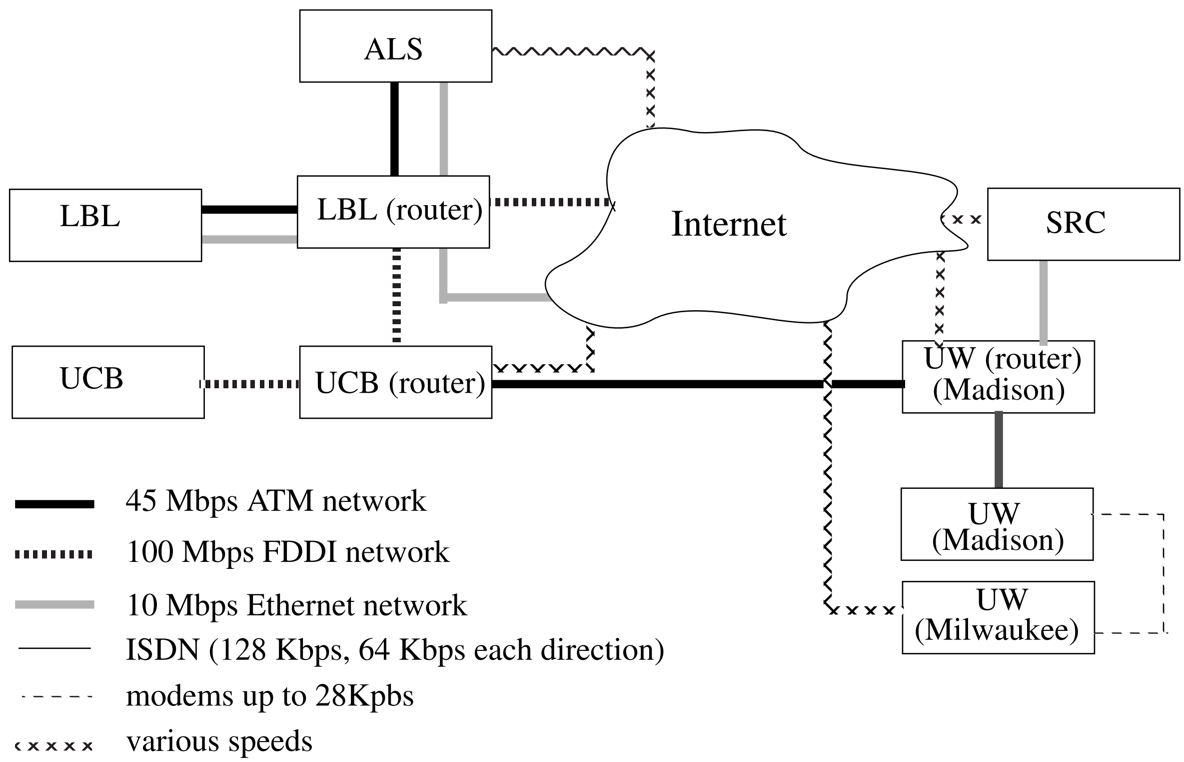

Figure 2 Current Network Connectivity between the ALS Collaboratory, LBL/UCB, the University of Wisconsin, and a Milwaukee site.

3.0 Communication Network Component

3.1 Physical Network

There are several separate streams of data within the ALS Collaboratory environment. The videoconferencing tools require network bandwidth to transmit voice and video from each of the collaborating researchers. In addition, there will be at least one video stream from a camera that is monitoring the sample chamber, so that the remote researchers are provided with feedback on experiments. The other data streams carry experiment parameters, experiment results, and control and coordination information.

The current network interconnections are shown in Figure 3 , The initial connectivity between the University of Wisconsin and the Advanced Light Source will be provided by the AT&T XUNet high-speed network. This network provides a 45Mb/s connection between LBL/UCB and University of Wisconsin (Madison), using an Asynchronous Transfer Mode (ATM) backbone and Fiber Distributed Data Interface (FDDI) rings at the endpoints. One of the XUNet FDDI rings runs through Soda Hall at UC Berkeley, and to a concentrator in building 50B at Lawrence Berkeley Laboratory (LBL). The connections from the concentrator are copper twisted pair. There are two options available for connecting the machines at the Advanced Light Source to the XUNet. The first option is to use ATM from the ALS to a router in building 50B which connects to the FDDI concentrator. The other option

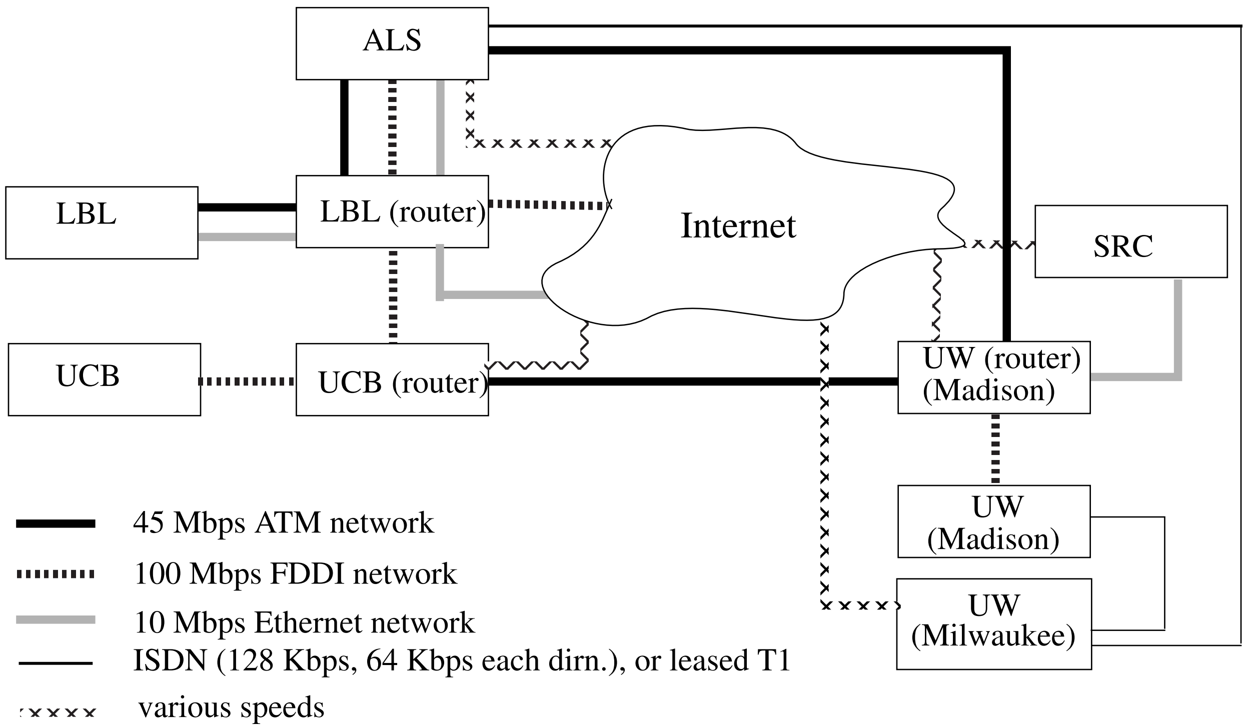

Figure 3 Current and Future Network Connectivity between the ALS Collaboratory, LBL/UCB, the University of Wisconsin, and a Milwaukee site.

is to run the FDDI fibers from the concentrator to the ALS, thus connecting the machines at the ALS directly to the XUNet FDDI ring, and consequently avoiding the additional router. The long-term plan for connecting to the XUNet is to provide an ATM connection from the ALS directly to the XUNet ATM switch. Other workstations at LBL may connect to the ALS via the ATM network, the XUNet FDDI network, or the Ethernet. Integrated Services Digital Network (ISDN) or leased T1 links will substitute the modem connection between Madison and Milwaukee. The current and future network interconnections are shown in Figure 4 .

The XUNet will provide a high speed data connection across the country, but the connections in Wisconsin pose additional problems. Although the XUNet provides a connection to the main campus at University of Wisconsin (Madison), the Spectro-Microscopy Facility users are located at a facility several miles from Campus (SRC), and at the University of Wisconsin (Milwaukee). The connection between Madison and these other facilities will be either Ethernet, Internet, leased T1 or ISDN.

One advantage of using the XUNet as a supporting network for the ALS Collaboratory is that it offers the opportunity to experiment with the Tenet real-time communication protocols developed at the University of California, Berkeley and implemented on the XUNet. These protocols can provide data throughput guarantees if needed. The disadvantage of relying on the XUNet is that it is an experimental network, which can be reserved by researchers testing new network routing and transmission algorithms. During reserved periods, the user reserving XUNet may reprogram the switches arbitrarily, thus preventing other users from accessing it.

The long term plan is that the connectivity will be provided by the 45Mbps Department of Energy - Energy Sciences Network (ESNet). The advantage of using the ESNet over XUNet is that it offers a higher degree of connectivity to various sites and greater flexibility of router and interface vendor. In addition, ESNet is expected to provide network users with control of quality of service. The use of ESNet depends on the deployment of a connection from the University of Wisconsin to ESNet.

The project has the additional long-term goal of providing access to the ALS Collaboratory over the Internet, since many of the users of the Collaboratory may only have Internet access. Although the Internet can provide connectivity between the researchers, it is not expected that its bandwidth will be sufficient to allow natural video conferencing interaction, and real-time data dissemination. The feasibility of using the Internet to provide connection to remote sites will be investigated. Experiment monitoring capabilities over the Internet is a project goal.

In the first phase of the ALS Collaboratory, the nodes at the ALS facility are connected to an ATM switch (not backbone) in Building 50B of LBL. The ALS nodes have an ATM interface, which operates at 100 Mbps, and uses the Transparent Asynchronous Transmitter/Receiver Interface (TAXI) protocol. The adaptation layer segments higher layer protocol frames into cells, and reassembles received cells into frames for delivery to the higher layers. The ATM connection between the ALS and the XUNet FDDI ring carries ATM cells.

3.2 Network Performance Measurements

The throughput requirements of the ALS Collaboratory vary. For data transmission, the highest throughput required is 280 Kilobits per second (Kbps), when an array of 512×512 elements of 32 bits is sent in 30 seconds. Other data transmissions may require 80 Kbps (10 Kilobytes per second), and a typical requirement is 8 Kbps. The throughput required for the transmission of black and white camera images with 512×492 pixels, assuming 8 bits per pixel, and updated as slow as 1 Hz is about 2 Mbps without compression. For real-time video conferencing, one needs to transmit 30 frames per second, although the size of each frame may vary, if only the changes between consecutive frames are transmitted. Without compression, the transmission of 15 frames per second, requires a throughput of about 120 Mbps. Assuming that the quality of 128 Kbps compressed video streams are acceptable, we may be able to reduce this requirement to128Kbps.

Real-time voice and video need isochronous service from the communication network, whereas data uses asynchronous service. For isochronous service, a network must provide guaranteed bandwidth and controlled jitter. Although the XUNet supports isochronous service, the Ethernet, and Internet networks support only asynchronous traffic. When asynchronous data transfer is used for video and voice traffic, no guarantees about regular delivery intervals, or network bandwidth are provided. The sending processor timestamps the voice data. The receiver processor buffers the packets, and recomputes their playout time. The video packets sent as asynchronous data are displayed as they arrive, so some jerkiness may be observed. Since voice and video packets sent using asynchronous mechanisms may be lost (usually due to congestion), so that clipped voice or video may be observed.

The physical network interconnections using shared transmission facilities of 10 Mbps or higher may be adequate for the purposes of the ALS Collaboratory. However, the achievable throughput over the various communication networks largely depends on the used congestion control mechanisms, the length of the connections, and the size of the messages. For example, the Transmission Control Protocol (TCP) does not allow full utilization of the network bandwidth for the first transmitted and acknowledged packets, because it uses a slow-start technique to avoid network congestion by gradually increasing the transmitter's window size until steady state is reached. The TCP flow control window size is a significant parameter of network performance. Also, it has been noted that throughput varies quite significantly with the connection length, and with message size [3]. The same study has shown that for cases in which the TCP congestion window becomes larger than the router's queue packet limit, reduced throughput is observed. This is one indication of the effect of different congestion control mechanisms operating at different levels, namely, at the transport layer (TCP), and at the network layer (router congestion control).

The delays introduced by the communication network also need to be quantified. The end-to-end delay of transmissions over the XUNet-based network depends on delays at the switches, and at the routers (Ethernet-FDDI, FDDI-XUNET). The XUNet backbone has been designed to have less than 1ms delay at each switch for short messages, and less than 100 msec for call setup. If TCP is used, one round-trip delay is required for establishing each TCP connection. One more round-trip delay is required for closing each TCP connection. Many other factors impact end-to-end delay, such as congestion control mechanisms, which may cause packets to be lost or buffered by the network. The end-to end delays associated with the Internet network may suffer large variances, because different routes may be used at different times. We are interested in determining average end-to-end delay, and its variance for both XUNet-based and Internet-based interconnections. If the User Datagram Protocol (UDP) is used, the protocol does not provide flow control or reliable transfer of packets.

In order to quantify delay-throughput characteristics of the different communication networks supporting the ALS Collaboratory, performance measurements must be conducted. In order to identify bottlenecks, we also need to perform measurements on each segment of the path between the ALS facility and the University of Wisconsin, Madison-Milwaukee.

We are currently carrying out the following network measurements:

1. XUNet measurements: For varying message sizes (from 64 bytes to 65K bytes), and for varying connection lengths, we are collecting delay-throughput data with the tool “ttcp” (both reliable TCP and UDP) for the following segments of the path between the ALS and UW, Madison:

1.1 ALS node to LBL Building 50B node (cell relay ATM connection).

1.2 LBL Building 50B node to UCB node (FDDI connection)

1.3. UCB node to UW, Madison (XUNet ATM frame relay connection)

1.4. ALS node to UW, Madison node.

2. Internet measurements: We are collecting (also with “ttcp”) delay-throughput data for paths via the Internet between a LBL Building 50 node, and a UW-Madison node, for varying message sizes (from 64 to 65K bytes), and also for varying connection lengths.

The next step in network measurement is to perform the tests 1 and 2, above, with a tool developed at U.C. Berkeley. This tool has a debug mode, which allows the monitoring of connection establishment, and transmission/reception of messages. Also, it is available in source code form, so that it may be useful in detecting network problems. Network measurements will also be conducted for the ISDN connections, which are planned between UW-Madison and UW-Milwaukee, and between the ALS facility and UW-Milwaukee. The results of these measurements will allow us to determine:

1)If there is a problem using any of the particular communication networks, or segments of them;

2)If the lack of guaranteed bandwidth, controlled average delay, and delay variance of networks which carry asynchronous traffic, significantly affect the isochronous voice and video traffic;

3)Which message sizes and flow control window size should be preferred.

4)What is the expected network performance for varying connection lengths.

An important element of the network measurement subcomponent is the study of the impact on delay-throughput of the different congestion control at the XUNet ATM switches. There are various proposals for new congestion control algorithms which improve the achievable throughput over ATM networks. Therefore, the above mentioned experiments will be repeated with different congestion control algorithms.

A long-term goal of this component is the study of real-time end-to-end delays achievable with the Tenet protocol suite, which may prove to be necessary for the support of all video conferencing needs of the ALS Collaboratory.

4.0 Data dissemination layer

The communication mechanisms currently available to connect with the experiment sites include TCP/IP (Transmission Control Protocol/Internet Protocol), and the new Internet multicast capabilities. TCP/IP, although useful to demonstrate early project capabilities, has disadvantages which will prevent its use as the final communication mechanism for the ALS Collaboratory. TCP/IP connections are point-to-point, thus the overhead of the communication scales exponentially with the number of participants and the overhead on the sending processor scales linearly. Communication overhead of a multicast mechanism scales sub-linearly with the number of participating sites and the additional overhead on the sending processor is negligible. Since this project is intended to allow multiple remote sites to participate in each experiment, multicast capabilities are required.

Various multicast primitives will be provided by the data dissemination mechanisms within the ALS Collaboratory: unreliable multicast, unreliable ordered multicast, reliable unordered multicast, and reliable ordered multicast. Unreliable refers to the fact that messages may be lost in transit between source and destinations. Unordered refers to the fact that messages may be received out of order at the destinations. The multicast algorithms that have recently been implemented on the Internet provide unreliable unordered multicast. We are building a generic data dissemination tool that will provide an interface to the application, allowing the application to choose the reliability and ordering characteristics that a particular data stream should have.

There are several properties provided by the data dissemination mechanisms on a connection regardless of the level of ordering and reliability requested by the application. These include: intact message delivery, membership information, and message buffering. Messages received from an application are not fragmented or corrupted when they are delivered at the destination site. Although the physical network may break the message into smaller packets, these packets will be reassembled at the destination before delivery to the application, and corrupted packets will be discarded upon receipt and treated as lost. If unreliable multicast is requested, and part of the message is not received, the entire message is discarded rather than deliver a partial message. If reliable multicast is requested, the message will be delivered as a single complete unit. The membership information facility provides an interface which allows an application to track the membership currently participating in a multicast connection. Messages are buffered by the data dissemination tools until the requested reliability and ordering have been achieved. If a large number of messages are pending within the data dissemination tools, and the buffers are full, then new messages from an application are refused until buffers become available.

An application program can open as many connections as it requires; the reliability and ordering characteristics are specified by the application when it establishes each connection. The ordering of multicast messages is only guaranteed for messages sent on a given connection. Messages received from different connections are not ordered with respect to each other. Although system wide message ordering could be provided, it is not currently planned.

The application programming interface consists of a series of library functions. The functions for handling a connection will mimic the TCP/IP functions, which include open, write, read, poll and close. The open connection will have as arguments the level of reliability and ordering requested for the connection, along with the normal port and address specifications. Additional functions will be added so that parameters to the membership facility may also be specified.

The method for using a multicast capability is chosen by the application programmer, since it is application dependent. One possible organization of the data dissemination mechanisms is as follows:

a) Experiment parameters use unreliable ordered multicast mechanism, since the most recent versions of the parameters are the most important and supercede the old settings;

b) Data to update 'real-time' display of the experiment output use reliable unordered or ordered multicast, depending on the requirements defined by the experiment and data display program;

c) Data that does not require 'real-time' distribution will be stored in a distributed file system, using the DCE Distributed File Service. This will allow experimental results to be shared among experimenters without duplicating the file system at each researcher's site.

The primary reason for choosing unreliable or unordered over reliable or ordered is performance. In the case of unreliable unordered multicast messages are passed to the application as soon as they are received. Unreliable ordered multicast performs a table lookup and update (minimal overhead) before passing the message to the application. With reliable unordered multicast, messages are passed to the application immediately on receipt but there is additional processing and network overhead in requesting, buffering and calculating the retransmissions. Reliable ordered multicast has the highest overheads since it has all of the same overhead as reliable unordered multicast and messages are potentially delayed before delivery to the application while a preceding message is retransmitted.A chart showing the effect of choosing one data dissemination protocol over another is given in Table 1 below. The four levels of reliability and ordering provided by the data dissemination tools are described in greater detail below.

4.1 Unreliable multicast connection

The unreliable multicast provides a best effort data stream that can be used to send data. Under this scheme, older messages are not rebroadcast, and the loss of a message by a single receiver does not

Protocol	Reliable ordered	Reliable unordered	Unreliable ordered
Unreliable unordered	Proc, msg, buf	Proc, buf	Proc
Unreliable ordered	Proc, msg, buf	Proc, buf	X
Reliable unordered	msg	X	X

TABLE 1. Comparison of the overheads associated with each data dissemination protocols. Each row and column lists a data dissemination protocol. Each square lists what overhead would increase if using the protocol in the column instead of the protocol in the row (proc = processor, msg = message delivery latency, buf = number of buffers needed to store messages).

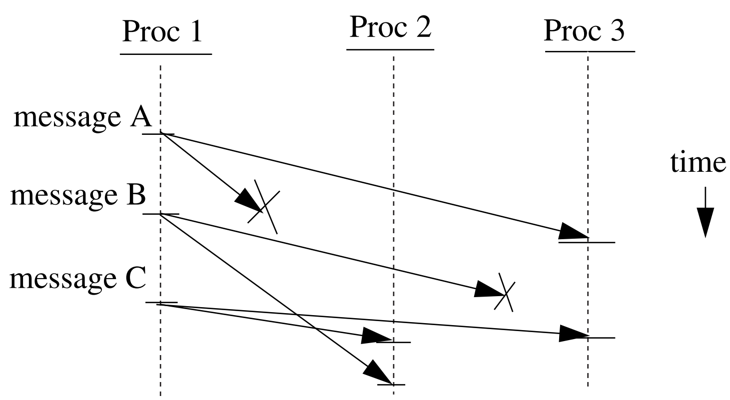

Figure 4 Unreliable unordered multicast. Each vertical line represents the progression of time within a processor. Arrows correspond to multicasts of messages and an X indicates that the message was lost or discarded. In this example, message A is lost in transit to processor 2 but is received by processor 3. Message B arrives at processor 2 after message C and it is lost in transit to processor 3. The lost messages are not retransmit.

mean that all receivers have lost the message. Also, each message is routed through the network independently, so that a later message may take a shorter path than an earlier message, and in fact arrive before the preceding message. The example with three processors shown in illustrates a message delivery pattern that could occur using unreliable unordered data dissemination., Some applications may benefit from using this data dissemination mechanism, because they do not require the delivery of old messages. In fact, it is detrimental to the goal of some applications for lost messages to be retransmit and delivered. Unreliable multicast can be used by applications that provide status displays, and parameter updates. The unreliable multicast mechanism of the ALS Collaboratory will rely heavily on the existing IP multicast mechanisms used on the Internet.

4.2 Unreliable ordered multicast connection

Because unreliable multicast may deliver messages out of order, an unreliable ordered multicast mechanism will also be provided by the ALS Collaboratory. The unreliable ordered multicast mechanism of the ALS Collaboratory will have the same properties as unreliable multicast, except that it will discard messages that are received out-of-order. If an application sends message A, then message

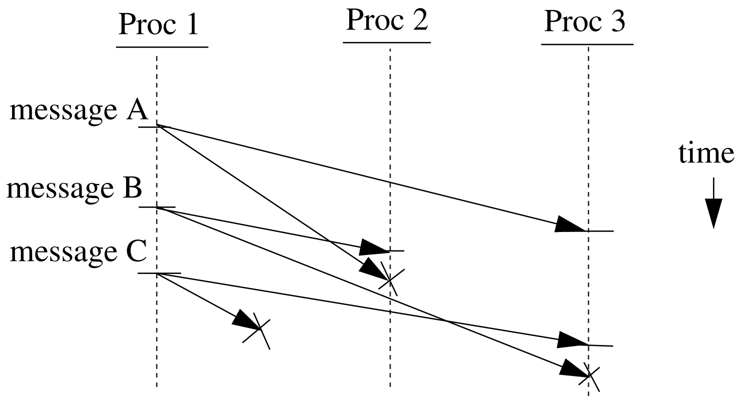

Figure 5 Unreliable ordered multicast. Messages A and B arrive out of order at processor 2 and so message A is discarded on receipt. Message C is lost in transit to processor 2. Processor 3 receives message A and then message C; message B is discarded when it arrives after message C at processor 3.

B, and message B is received at the destination before message A, the latter message will be discarded if it subsequently arrives. The unreliable ordered multicast will be implemented as a filter above unreliable unordered multicast. It will place a timestamp in the header of each message when it is sent, and the highest timestamp received from each source will be maintained. A message is discarded on receipt if its timestamp is less than or equal to the current timestamp for the given source (Figure 6). This mechanism does not require synchronized clocks since a separate timestamp is kept for each source.

4.3 Reliable unordered multicast connection

For data streams that require reliable dissemination of data, two reliable multicast mechanisms are needed. The first mechanism provides reliable unordered multicast, which passes the messages to the application as they arrive; messages that have not been received are retransmitted, and eventually received and delivered to the application (unless there is a failure at the receiver, or at all processors that have received the message, or some processors become disconnected from the network).This type of transfer will be useful to applications that need all the messages delivered at the remote site,

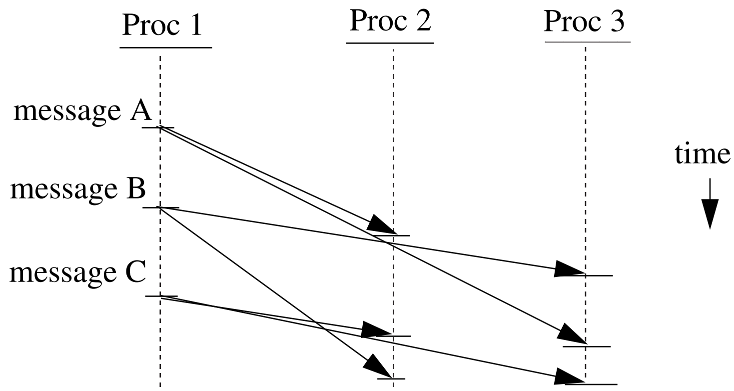

Figure 6 Reliable unordered multicast. Messages A, B and C are all received but message B is received after message C at processor 2.

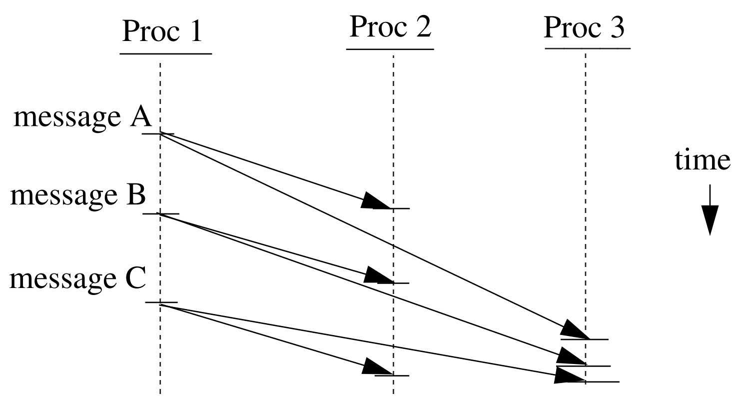

Figure 7 Reliable ordered multicast. Messages A, B and C are all received in order by all processors.

but order of that delivery is unimportant. Figure 7 shows a possible application level receipt pattern for messages A, B and C which are multicast by processor 1. In this case, the large delay between the time processor 1 multicasts message A, and the time it is received at processor 3 may be either due to its having taken a long route through the network, or to the fact that the original multicast of message A was lost, and the subsequent retransmission was received by processor 3.

4.4 Reliable ordered multicast connection

The second mechanism provides reliable ordered multicast, where messages from any given source are delivered to the application at the destination in the order sent by the originating application. Loss of a message will delay delivery of all subsequent messages until the message can be retransmitted. Recovery of a lost message takes at least a roundtrip time to the nearest processor participating in the multicast. On average, the recovery will take longer than that, since the retransmission may be lost, the retransmission request may be lost, or the nearest processor may also have lost the message. Figure 8 illustrates a possible scenario for a reliable ordered multicast.

The design of the reliable ordered and unordered multicast data dissemination mechanisms will be based on the Totem protocol [1].

5.0 Security and Arbitration layer

5.1 Safety and Security component

Security and safety issues need to be adequately addressed before placing a multi-million dollar facility like the Advanced Light Source on the Internet. The current researchers at the Advanced Light Source are intimately familiar with the hardware and software used to run the experiment, because they developed most of the hardware and software themselves, and they operate the facility on a full-time basis. Regardless of whether the researcher is remote or local, application control systems must have safety features. The remote researchers are unlikely to be so well versed in the equipment and procedures, thus additional safety features must be considered. Safety features are built by the application developers, because the data dissemination layer does not have the application specific information required to build safety mechanisms.

Connections made over a wide-area network like the Internet are less reliable than local connections, and may be unavailable or too slow to run the experiment. In these cases the security and arbitration layer will provide mechanisms to inform the experiment control layer of the connection status. The data dissemination facilities will provide functions for estimating the current round-trip time between the local site and any other site. It is also possible to build facilities that will allow an alarm to be set so that applications are advised when the round-trip time is larger than a set value. Such facilities are not currently being planned.

The security of the system will be controlled from the end points of the ALS Collaboratory, namely the ALS facility, and the workstations in Wisconsin. The security is provided by the Distributed Computing Environment (DCE). This choice was made because DCE offers a security server and clients using essentially Kerberos. The security server will allow the researchers to join a session without having to send their password over the network, and it will also allow experimental data to be encrypted, if necessary. Each user of the system will have a DCE login and will run the DCE client software at their site. The data dissemination and experiment arbitration and control facilities will use the security software to authenticate the remote user before allowing access to the system.

A significant piece of DCE is it's security service. This service is an implementation of the Kerberos Authentication Service developed at MIT [4]. Kerberos is an authentication system designed to allow authentication of users, clients and servers in a network. It uses a central database of passwords, (for both users and servers), and tickets encrypted using passwords. Clients include these tickets with each request to a server to verify their identity to the server.

When a user first logs on to a workstation, they are prompted for a username and a password. The username, but not the password, is sent off to the, (possibly remote), Kerberos Server. This special server knows all user passwords and creates a special ticket encrypted using the user's password as a key. This ticket is returned to the user's workstation where it is decrypted using the password entered by the user. If the decryption succeeds, the user is authenticated. The proof of this authentication is in the possession of this special ticket, known as a ticket-granting-ticket, created by the Kerberos server.

In order for a client program to use a server, a ticket must be presented to the server to authenticate the client. This ticket is custom made for that particular client/server pair. This custom built ticket is not created by the client but by another special server called the Ticket Granting Server.

A client must first contact the Ticket Granting Server before contacting any other server. Being a server itself, the Ticket Granting Server needs the Ticket-Granting-Ticket to verify the client requesting the server ticket. The Ticket Granting Server verifies the client by decrypting the Ticket-Granting-Ticket and then builds a server ticket, encrypted with the server's password, for the requesting client. The client uses the server ticket obtained from the Ticket Granting Server to make authenticated requests to a particular server.

The initial remote monitoring system does not require security and safety features. Since the security will be end-to-end, the intermediate nodes between Wisconsin and the ALS do not need to run a security system.

5.2 Experiment arbitration component

There is an inherent question when several researchers collaborate to run a single instrument; who is in control? This question is normally answered by looking at who has their hand on the control knob, but in the ALS Collaboratory there are potentially several instances of the control knob. There will inevitably be situations where several people attempt to turn the knob at the same time. We are in the process of evaluating existing distributed lock managers and conference control utilities, in order to evaluate their suitability to ALS Collaboratory. If needed, a customized distributed lock manager will be built. In any case, the collaborating researchers must be able to request control of the experiment, and the arbitration component must ensure that there is always only one researcher in control at any point in time. The design of this component is the subject of future work.

The resource arbitrator will be composed of two components. The first component is a directory service. An object that wants to allow remote control registers with the directory service. When an object registers with the directory service, it specifies several parameters. These parameters are

a) Object name

b) Initial site to place in control of the object (can be left unspecified)

c) Method of determining which requestor will get control of the object

    1) Never changes from initial site (all other sites are monitoring),

    2) First-come first-serve (requestors are granted control in the order of their requests and the next requestor is given control when the current requestor relinquishes control),

    3) Always determined by a specific site (control always reverts to a main site when the current requestor relinquishes control and the main site can choose which requestor is next granted control),

d) Network performance requirements (maximum allowable round-trip time to the controlling site)

e) Action to take when connection to site currently controlling the object is lost.

All the parameters will have default values except the object name which must be specified uniquely for each object. If none of the sites is specified to be initially in control, then the first site to request control is granted it automatically. When the object registers with the directory service, two actions occur. The directory service spawns a process to take care of arbitrating control of the object and the directory service advertises the object. The spawned process establishes a connection with the object and exchanges control status information with the object. The exact interface definition has not yet been developed but it will contain well defined message types such as control released (including reason for release) and control passed to 'hostname'.

Individual user sites are also running the directory service and can choose the object from their local directory listing. When the user chooses the object from the directory an arbitrator process is spawned. The arbitrator process provides a user interface containing the object's name, the current list of participants, the current controlling participant and requestors, and a button for requesting control of the object.

The user interface to the experiment arbitration component will have a list of participants and will display a green dot next to the researcher that is currently in control. When a researcher requests control of the experiment, a yellow dot is shown next to the requesting researcher's name. The arbitration mechanism will use the reliable ordered data dissemination mechanisms for delivery of its messages and thus control requests will be totally ordered with respect to each other.

6.0 Experiment control, recording, and video conferencing layer

6.1 Experiment control component

The existing experiment control system has been built in Labview on a 486 based PC located at the Advanced Light Source. Rather than reinvent the control system we have investigated the option of using Labview in the ALS Collaboratory environment. The chosen design has a PC that controls the hardware and acts as the Labview server. The researchers using the Collaboratory, which include the ones that are physically present at the ALS, run Labview as clients of this server. This design is implemented with relatively minor modifications to the existing Labview programs: the server and clients run the same Labview program, which is equipped with a switch that indicates whether the site is a server or a client. This approach has the advantage of requiring a single version of the program to be maintained. Of central importance to this scheme is the maintenance of a single repository of the current version of the code that is accessible from all researcher's sites.

The development of the experiment control software and its interface at the server and remote sites will be developed by the experiment programmers at the ALS. Example methods of using TCP and UDP communication have been developed and implemented. The interface to the data dissemination tools will be well defined and example methods of using these tools will also be developed but the actual use of the tools and integration of the tools into the experiment control are the task of the experiment control designer. They are thus not described in this design document.

6.2 Experiment recording component

The recording of experiments conducted at the Advanced Light Source Beamline-7 currently uses a paper notebook system. This notebook is the record of the experiments which have been performed, and the parameter settings used in each experiment. The names of the result and analysis files are also recorded in the notebook. This notebook thus provides an index to these files. In addition, the researchers include graphs of obtained results, generated by various graphics tools, photographs, sketches related to the experiment, and hand drawings of observed results. A large percentage of the annotations on the notebook require the use of special symbols. Although this recording paradigm has worked well in the non-collaborative environment, it is not an adequate recording mechanism for the ALS Collaboratory. The conversion to an on-line notebook system is critical to this project. Because different workstations may be connected to the ALS Collaboratory, the on-line notebook system must be a cross-platform system, and it must be able to emulate fairly closely the original notebook system. In addition, changes made to a notebook page must be visible to the collaborators, as they occur during the course of the experiment.

Several on-line notebook systems were investigated. The only cross-platform, commercial product, that could potentially fit the Collaboratory's needs is the Virtual Notebook System (VNS) from Forefront. However, tests of the VNS package revealed several problems. The first problem mentioned by the researchers who experimented with VNS is its counter-intuitive user interface. The other major problems are that special symbols are not easily used, and customized forms cannot be created, as required by researchers making annotations about experiments. Lotus Notes was also considered but despite the name, this product is not a notebook. It is instead a document database system which provides storage and cataloging of papers, notes, and e-mail for easier search and retrieval.

We are investigating the remaining options available to provide an on-line notebook system that will meet the needs of the remote environment and the experiment. One option is to build our own shared notebook system. Netscape-based tools are being considered. The user interface to the shared notebook must have a very easy to use special symbols menu, a simple user interface to text input, capability to configure and easily utilize forms, and a drawing tablet interface. Also, the system must be able to maintain pointers to data files, and to import figures or photographic files, thus emulating the original notebook system. A word processing/document formatting package could be used as a notebook system, with the disadvantage that the forms-based entry and collaborative aspects would be unavailable with this option. Yet another option is to buy a source code license for VNS, and modify the source code to suit the needs of the project.

6.3 Video Conferencing component

The videoconferencing component is comprised of a conference control tool, videoconferencing equipment, and three tools developed for use on the Multicast Backbone of the Internet: *vic*, *vat*, and *wb*. These tools provide teleprescence of the researchers. The *vic* tool provides the video portion of the videoconference software. It allows the user to select a data format and compression parameters to be used in sending the video across the network. *Vic* also allows the user to choose a maximum data rate for the video. The *vic* tool can be used in viewing the output of any cameras in the environment including cameras focused on experiment equipment.

The network bandwidth requirements for video, particularly when sending more than one video stream, can be quite significant. The tool *vic* offers the user a wide range of choices between high and low quality video. Therefore, the video bandwidth requirements can be scaled according to the actual bandwidth available. *Vic* is an excellent tool for the ALS Collaboratory because, although it is designed for use over a high-speed network, there is a significant possibility that the actual connection may be provided over the Internet.

The *vat* tool provides the audio portion of the videoconference software and provides functions to change the microphone gain, mute the microphone, shut-off sound from specified sources, and set speaker volume. The *wb* tool provides a shared whiteboard that is displayed at all of the participating sites and allows the sites to share a drawing space. This space can be used to draw on, display postscript files, display screen images cut from elsewhere on the screen, or type directly onto the whiteboard. The *vic*, *vat*, and *wb* tools all use unreliable multicast to provide communication between the sites and *wb* implements its own version of reliable unordered multicast. Other videoconferencing tools such as nevot, ivs, nv, CU-SeeMe, insoft, proshare, and others are also being considered for use on the project.

Conference control tools that manage the videoconference still need to be developed or obtained from an outside source. The conference control tools will provide a uniform and intelligent interface to the tools and will handle coordination aspects of a videoconference. One desirable feature, which can be provided by the conference control tool, is the automatic start and stop of the videoconferencing tools. Bandwidth requirements of the tools can also be regulated by the control tool by sending low video rates from remote researcher sites, and increasing the video rate when the researcher is talking.

The experimental equipment at the beamline covers a large area, and the researchers at the beamline need to be able to move freely around the beamline. Two remote control cameras placed at the beamline will provide the remote user with a means of following the activity. The cameras have pan, tilt and zoom and are controlled via an RS-232 port from the Sun workstation at the beamline. The remote operator has a panel which allows individual cameras to be controlled, moved, and monitored. The software for providing this remote control via the RS-232 port is a part of the planned development.

7.0 Project Status

The first phase of the ALS Collaboratory, i.e., the experiment monitoring capability is nearly completed. We are currently installing a prototype remote monitoring system at the University of Wisconsin. Several issues are being addressed in the implementation of the first phase.

- The existing Labview programs for control of the experiment were modified to allow operation as server and client. The standard operating procedure at the Advanced Light Source will be converted to operate the experiments from a client located physically at the Beamline. This ensures that the client software is maintained as upgrades to the experiment control software are performed. The decision to integrate the client/server software into the Labview programs, and offer it as the experiment control component, has allowed us to give the researchers time to become familiar with the new experiment paradigm and to test it.

- The current system is only capable of handling a unicast connection to send experiment data and parameters, using UDP and TCP. The data throughput requirements for effective participation in the experiment will be evaluated using actual experiment data from the monitoring system.
- The Virtual Notebook System from Forefront Technologies has been evaluated, and deemed to be inadequate in its present form.
- The DCE software has been installed, and we have started work on the distributed file system and security server in preparation for phase two where these facilities will be critical.
- Some preliminary throughput measurements have been carried out. The results are the following:
  - From LBL (router) to UCB (router) via FDDI ring: approximately 32 Mbps.
  - From LBL workstation (on the Ethernet) to UCB workstations on the FDDI ring: approximately 8 Mpbs.
  - From LBL (router) to UW (routers) via XUNet (with XUNet reserved to other users): approximately 2 Mbps, with a packet length of 9,8304 bits. With a packet length of 160 bits, the throughput dropped to 1 Mbps.

8.0 Future Accomplishments

8.1 Remote experiment monitoring

As the first phase of the project completes, we will be conducting reviews of the hardware and software design using the prototype system. The collaborating researchers will use the prototype system to begin experimentation of the ALS Collaboratorium, and to evaluate the hardware and software designs. Feedback from these reviews will be incorporated into the design of the second phase (experiment control), as needed.

Study of throughput and bottlenecks are currently being carried out. A document on the measured throughput of the various network connectivity is forthcoming.

The primary task of the project first phase that remains to be completed is the reliable (unordered and ordered) multicast mechanism components.

8.2 Remote experiment control

In the second phase of the project we will be developing several additional tools to support control of the experiment from a remote site.

- DCE distributed security and file servers.
- Mechanisms for transfer of experiment control.
- Safety interlocks on the server control system to prevent equipment damage and unsafe operation.
- Reliable ordered multicast.

- Videoconferencing in a heterogeneous environment including PCs and Sun workstations.

BIBLIOGRAPHY

[1] D. A. Agarwal, “Totem: A Reliable Ordered Delivery Protocol for Interconnected Local-Area Networks (Ph.D. Dissertation),” ECE Technical Report #94-29, University of California, Santa Barbara.

[2] A. G. Fraser, C. R. Kalmanek, A. E. Kaplan, W. T. Marshall, R. C. Restrick, “XUNET2: A Nationwide Testbed in High-Speed Networking,” AT&T Bell Laboratories, Murray Hill.

[3] V. N. Padmanabhan, J. C. Mogul, “Improving HTTP Latency”.

[4] J. G. Steiner, C. Neuman, J. I. Schiller, “Kerberos: An Authentication Service for Open Network Systems,” Project Athena, M.I.T., Cambridge, MA. (March 30, 1988)