INITIAL RESULTS OF THE CD-1 RELIABLE MULTICAST EXPERIMENT[1]

Deborah A. Agarwal[2], Richard Stead[3], Brian Coan, James E. Burns, Nishith Shah[4], Nicholas Kyriakopoulos[5]

Abstract

A new version of the CD-1 continuous data protocol has been developed to support a multicasting experiment. This new version was developed to study the use of reliable multicast for sending data to multiple receivers in a single operation. The purpose of the experiment is to evaluate the existing multicasting technology for possible use in the Global Communication Infrastructure. This paper describes the initial results of the experiment.

Background

IP Multicast

On the Internet, a class of applications has emerged that requires the ability to send messages to a group of receivers efficiently. This need has led to the development of the multicast capability in the Internet Protocol (IP). Basic IP multicast is a simple communication mechanism that allows a single message to be sent to a group of receivers as an integral part of the communications services offered at the internetworking level. IP multicast is an unreliable messaging service implemented in the hosts and routers of the network. The multicast addresses are a separate address range recognized by the routers as multicast groups. Multicast packets are sent addressed to a multicast address. Applications that wish to receive the multicast packets open a connection to the multicast address and their host automatically transmits a join message to the nearest router. The router then adds the host to the multicast dissemination tree for the group. The multicast tree is dynamic and provides an efficient means of transmitting a packet through the network to reach all the receivers without traveling any link more than once. The routers at branch points in the tree duplicate the packet and send it down all the tree branches. For more detail regarding the multicast routing protocols, see [5] and [10].

The IP multicast communication mechanisms are gradually becoming a standard part of the Internet protocol suite and they co-exist with the unicast Transmission Control Protocol (TCP)[4] and User Datagram Protocol (UDP) mechanisms. The IP multicast mechanisms do not replace the unicast mechanisms; they instead provide an additional service. Since IP multicast is still a relatively new technology, routers do not come with IP multicast enabled by default. The router administrator must enable it before it can be used in the network. Not many commercial applications make use of IP multicast so many Internet Service Providers (ISP) have not yet enabled the capability.

Reliable Multicast

Reliable multicast is effectively the multicast equivalent to the TCP protocol. Reliable multicast provides reliable delivery of messages to multiple receivers. It uses IP multicast to provide the message dissemination capability and adds reliable delivery mechanisms. A reliable multicast protocol provides several potential advantages over TCP when there are in fact multiple receivers. With reliable multicast the receivers in a group can be reached by sending a single message. Using TCP the messages would need to be sent to each receiver individually by the original sender

---

[1] The views expressed in this paper are those of the authors and not necessarily those of the United States Government.

[2] Ernest Orlando Lawrence Berkeley National Laboratory. Sponsored by U. S. Department of Energy, Office of Nonproliferation and National Security, Office of Research and Development, Contract No. DE-AC03-76SF00098

[3] SAIC

[4] Telcordia Technologies. Sponsored by U.S. Department of Defense, Defense Threat Reduction Agency, Contract No. DTRA01-99-C-0025.

[5] The George Washington University

or a site acting as a forwarder. Reliable multicast is not yet a standard communication protocol that is part of the operating systems of hosts. Reliable multicast is instead run as an application-level protocol. An instance of the reliable multicast software is run at each of the senders and receivers participating in a reliable multicast session. The software then uses IP multicast for its underlying communication mechanism. There are several commercial and freeware reliable multicast protocols available today. Some of the available reliable multicast protocols are described in [1], [5-8], and [11].

Reliable multicast protocols are each designed for use in a particular class of applications. The application classes differ in the message delivery reliability and ordering properties required and the number of participants in a multicast group. The reliable multicast protocols also differ in their methods of indicating message loss, determining membership, and retransmitting lost messages. Currently the network routers do not participate in the reliable multicast protocol; they only need to handle IP multicast messages.

Two of the existing reliable multicast protocols are the Multicast Dissemination Protocol (MDP)[8] and the Reliable Multicast Transport Protocol (RMTP)[7]. The MDP protocol provides bulk data (large messages with no delivery timing constraints) transfer capabilities. It sends the data in a first round and then sends retransmissions in subsequent rounds until all receivers have acknowledged receipt. The MDP protocol was originally developed as the underlying reliable multicast protocol for a satellite image dissemination service. The RMTP protocol provides reliable delivery to groups with a single sender. It provides synchronous delivery of messages; emphasis is placed on timely delivery of the messages to the members of the group. RMTP provides hierarchical mechanisms to scale to large groups; processes within a local area join together and a leader represents the local area. The RMTP protocol uses the leader to report packet loss and acknowledge receipt. RMTP currently uses a rate-based flow control mechanism. The current developers of RMTP are working with the IETF reliable multicast research group to identify congestion control algorithms that will be acceptable to the Internet community at large. The RMTP protocol was originally developed for file transfer applications but it is also an effective means of transferring real-time data. The current version of the RMTP protocol is RMTP II and it is now a commercial product available from Talarian Corporation.

Multicast in the Global Communication Infrastructure (GCI)

The GCI Integration Laboratory within the International Data Center (IDC) contains a test network composed of all the essential components of the GCI but isolated from the GCI so that it can be used for testing without impacting the GCI. The GCI Integration Laboratory is composed of three "remote" sites that are connected via VSAT to a satellite hub in Germany. The satellite hub is connected to the IDC in Vienna using a frame relay link. The workstations representing the "remote" sites and the IDC are all located in the GCI Integration Laboratory (See Figure 1). The routers and software in the GCI Integration Laboratory network are the same types and versions used in the rest of the GCI. The IP multicast capabilities were enabled on the GCI Integration Laboratory routers temporarily for feasibility tests to allow testing of IP multicast.

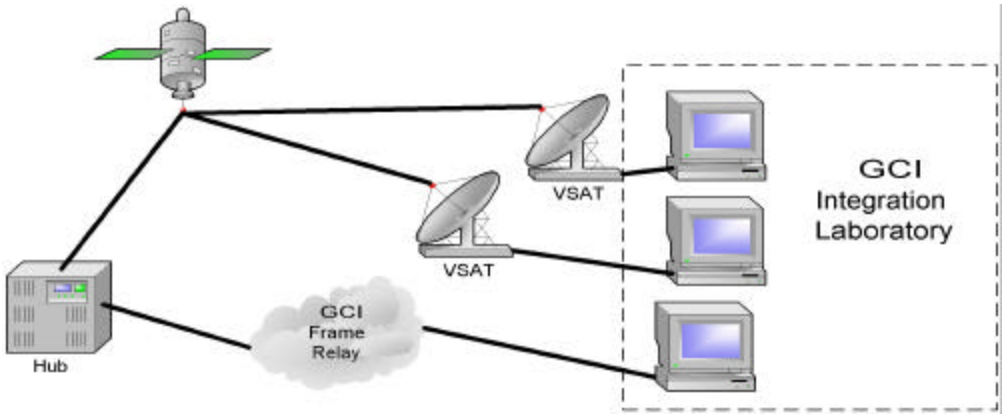

Figure 1: The GCI IP multicast test configuration

Extensive tests measuring throughput, loss, round trip time, and packet size were conducted in the GCI Integration Laboratory. The test software was designed to allow packets of a designated size to be generated and sent using a periodic rate. The software had the ability to send the data either one-way or with the receiver(s) bouncing the traffic back using another IP multicast address to allow round trip times to be measured. The round trip times were measured using a workstation at the end of the frame relay link and one "remote" workstation. One-way tests using one sender and two "remote" receivers were also conducted. The tests indicate that the GCI Integration Laboratory components are able to pass IP multicast traffic at the full capacity of the links without problem. It should be noted that these tests were conducted on the GCI Integration Laboratory components; however, there is no reason to think that similar performance could not be obtained in the rest of the GCI. The only problems encountered were with the firewall; the firewall blocked all IP multicast traffic. This problem was solved by running the tests on machines that were all on the same side of the firewall.

The MDP protocol which is a freely available was obtained via the World Wide Web and installed in the GCI Integration Laboratory to test the basic capability of the reliable multicast protocols to run in the GCI network. Reliable multicast transfers of JPEG images were performed using the MDP protocol. The images were transferred back and forth between remote VSAT located sites and an IDC located machine. The MDP protocol provided reliable multicast over these GCI links without requiring any modification or tuning to account for the satellite links. The RMTP II protocol is a commercial product so it has associated licensing fees and is not source code available. The RMTP II protocol was not tested in the GCI Integration Laboratory but it was later installed in a local-area environment and evaluated for ease of use and robustness. More information about the studies conducted can be found in [2].

The CD-1 Continuous Data Protocol

The CD-1 protocol is the protocol currently in use within the GCI for sending continuous data. The CD-1 protocol is designed to provide transmission of continuous data between two locations over a network. Specifications for new versions of the continuous data protocol have been submitted. These new versions are CD-1.1 and CD-2. The overall data rate in the GCI is expected to be on the order of 10 gigabytes per day. Approximately 8.5 gigabytes per day of this data is continuous data sent using the CD-x protocol.

The experiment described in this paper focussed on early experimental deployment of reliable multicast and so the CD-1 protocol provides the basis for the prototype. The unmodified CD-1 protocol used the TCP protocol to send data reliably. CD-1 delivers data only while there is a TCP connection between the sender and the receiver. When the connection is down the sender buffers data locally waiting until a connection can be re-established to the receiver. Some amount of data may be lost by the CD-1 protocol when a TCP connection is closed due to a failure.

The CD-1 protocol runs at the sender of the data and at the receiver and is responsible for transmission of the data to the receiver. The CD-1 protocol retrieves the data from a Last In First Out (LIFO) Heap at the sending side and stores it in a Disk Loop at the receiving site. The protocol uses a reliable unicast TCP connection to transmit the data. While the sender is connected to the remote site, it sends data from the LIFO Heap to the receiver. Whenever the connection is down, data accumulates in the LIFO Heap and the data is sent when the connection is re-established.

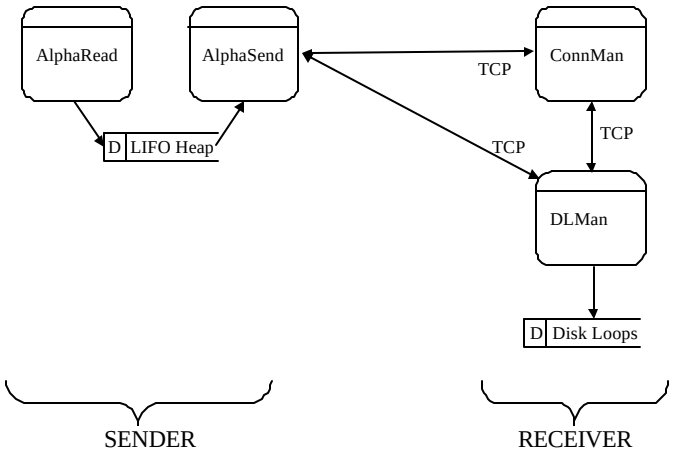

Figure 2: The CD-1 continuous data protocol.

The CD-1 protocol as implemented as a series of separate software processes (Figure 2). The Alpha Read module reads the sensor data and places it in the LIFO Heap. The Alpha Send module establishes a connection to the receiver by first contacting the ConnMan using TCP to get connection parameters. Alpha Read then uses the connection parameters to establish a TCP connection to the Disk Loop Manager (DLMan). Once the connection to DLMan has been established AlphaRead reads the data from the LIFO Heap and sends it to the DLMan process for storage in the Disk Loops.

The AlphaSend module generates and sends a Format Frame before sending any data to the DLMan. Receipt of a format frames is critical to the correct interpretation of the data stream at the receiver. The format frame describes the content and representations for the data to follow.

Experiment Technical Plan

The purpose of this experiment is to provide a small-scale technical feasibility trial of the use of reliable multicast as a transport mechanism for CTBT continuous data. At the beginning of the experiment the CD-1 protocol was the only existing version of the continuous data protocol: the CD-1.1 and CD-2 protocols existed only as specifications. Unfortunately, since the CD-1 protocol contains no end-to-end reliability mechanisms [3], it is a less than ideal candidate for long-term use with reliable multicast. The CD-1.1 and CD-2 protocols will both contain end-to-end reliability mechanisms when they are implemented and replace CD-1. However, the use of CD-1 for a technical feasibility prototype allowed for rapid development. To reduce development cost and time, the prototype multicast-enabled implementation of CD-1 used as much of the existing code as possible.

Another goal of the experiment is to have a comparable frame loss rate to that exhibited by the existing CD-1 protocol implementation. The software is intended for limited duration deployment but it is important that the system be sufficiently robust to operate in the field with limited manual support. The system needs to be robust to network and end station failures and able to automatically start and restart processes on initialization and after failures. The experiment also provides an opportunity to test the robustness of the COTS solution (RMTP-II from Talarian Corporation) to achieve TCP-like reliability.

Reliable Multicast Protocol

The MDP version two and RMTP II protocols were obtained for evaluation. At the time of the evaluation, the MDP protocol was only available as a bulk-transfer program; an application-programming interface (API) was available as an early beta but was not yet ready for use. The RMTP II protocol included a released API with good documentation and support.

Each of these two protocols were evaluated for use in the CD-1 multicasting experiment. The principle deciding factors were that the application-programming interface (API) was readily available for RMTP and the commercial support. The release of the MDP API only happened after the evaluation period was over. Both protocols were tested and run in a local network to gain experience with their operation and evaluate ease of use, features, and efficiency in a real implementation.

Multicast-Enabled CD-1 System Design

The goal of minimal modification to the existing CD-1 software and reuse of code was achieved by reconfiguring the existing CD-1 software (Figure 3). In the new design, the LIFO Heap that was originally only located at each CD-1

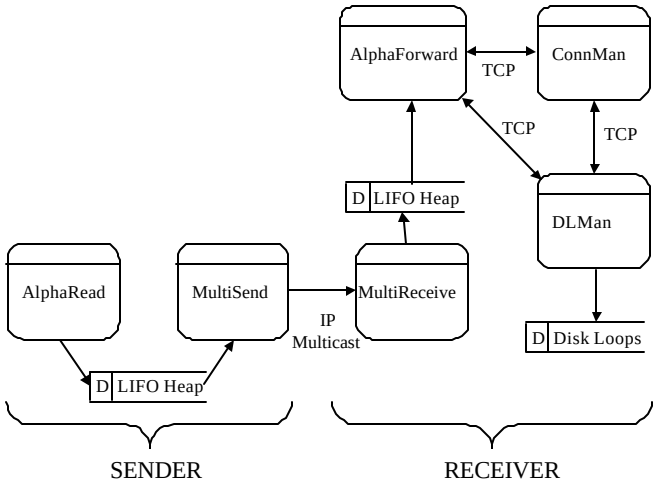

Figure 3: Design of multicast-enabled CD-1.

sender is replicated at each receiver. The MultiReceive module places received data into a LIFO Heap at the receiver. AlphaSend is replaced by AlphaForward and migrated to run on each node that receives the multicast. The migrated AlphaForward reads frames from its local LIFO Heap and uses TCP to connect to the ConnMan and DLMan (which are co-located with AlphaForward). In the reconfiguration, the multicast components are inserted between the two LIFO Heaps. The original CD-1 code that writes and reads the LIFO Heap remains unchanged. The MultiSend and MultiReceive modules are new but they reuse a significant amount of code from the CD-1 implementation. These new modules integrate the RMTP protocol as their mechanism for transporting data between sender and receiver. The multicast group used for sending the data, in this experiment, is set using a parameter at the sender to remove the need for modification of ConnMan. The RMTP protocol provides membership information and allows the sender to determine which receivers are currently connected to the group and receiving data.

Fault-Tolerance

During failures, there is more needed to provide behavior equivalent to the original behavior of the CD-1 system. If the sender crashes and recovers, the sending of data is discontinued during the crash and resumes after the recovery. This behavior is the same as in the original CD-1 implementation. If a receiver crashes, the frame sending continues and the operational receivers continue to get frames. When the receiver recovers, it rejoins the multicast group and immediately resumes receiving frames. The recovered receiver missed frames sent while it was down. In the original CD-1 protocol, when a receiver is unavailable the sender quits sending and instead buffers new data until a connection to the receiver is re-established. If the multicast enabled version of the CD-1 software were to stop transmitting when any one receiver is down then the perceived reliability of the CD-1 software from the other receiver(s) would be less than the original unicast CD-1 software. So, a mechanism for a recovered receiver to "catch-up" and get the frames that it missed while it was down is required.

As shown in Figure 4, the UniSend and UniReceive modules provide this "catch-up" capability. The RMTP II protocol tracks the status of senders and receivers and the sender is notified of down or recovered receivers. The frames destined for a down receiver are stored in the "catch-up" LIFO Heap and the frames are sent to the recovered receiver using a unicast TCP connection in parallel with the ongoing multicast connection. "Catch-up" frames and new frames are merged at the receiver using existing AlphaForward functionality. With the addition of the catch-up mechanisms, the multicast-based CD-1 system provides a behavior comparable to the unicast-based CD-1. It should be noted that, the number of unicast connections operating in parallel with the multicast connection is proportional to the number of receivers participating in the "catch-up" process. As the number of the receivers participating in "catch-up" increases, so does the number of the additional TCP connections resulting in a system that would begin to resemble data forwarding by the transmitter. The end-to-end reliability mechanisms designed into the CD-1.1 and CD-2 protocols directly integrate the "catch-up" mechanism.

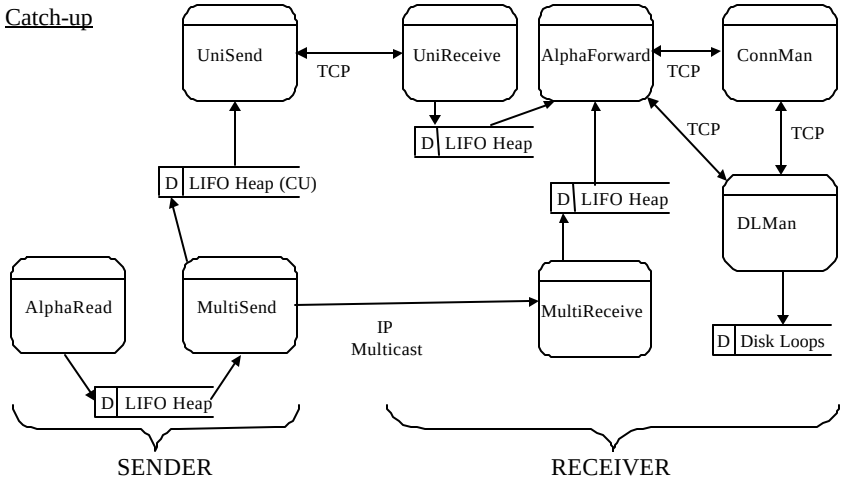

Figure 4: Multicast-enabled CD-1 with "catch-up" capability.

Test Results

The prototype multicast-based CD-1 implementation has been tested in several configurations to evaluate its performance. The initial tests of the system were performed internally at Telcordia using two receivers. These tests

were only able to check the receiver process through to the LIFO Heap. The next series of tests were between Telcordia and the PIDC. In these tests, the sender was at Telcordia and the receivers were at Telcordia and the PIDC. This test allowed testing with a moderate latency link over the Internet. They also allowed the software all the way through to the DLMan to be tested. The final tests used a version of the RMTP protocol that allows the network characteristics related to loss and latency to be emulated. In these tests characteristics representative of the GCI satellite connected IMS stations were emulated. All the test configurations used one sender and two receivers.

The tests of the multicast-based CD-1 software had several objectives:

- accurate end-to-end delivery of data;
- correct merging of frames from the multicast and catch-up channels;
- reliable operation over an extended duration;
- activation and operation of the catch-up channels;
- validation of the multicast parameters; and
- functionality during start-up and failure.

The tests between Telcordia and the PIDC were run continuously for eight days. Six times during the eight days the receiver at the PIDC became unreachable from the sender at Telcordia. In each of these cases, the catch-up channel was activated when the receiver rejoined the multicast channel. The correct transfer of the merged catch-up and multicast data to the DLMan at the receiver was also tested.

The simulation mode of RMTP was used to emulate the GCI connections. In these emulations, the network delay between the sender and the receivers used an exponential distribution with a mean of 1.2 seconds, a standard deviation of 0.2 seconds and cut-offs at 1.0 and 1.7 seconds. The loss probability was set to 0.5%. These parameters are consistent with data obtained from multicast tests performed in the GCI Integration Laboratory. The CD-1 prototype performed well in this test. Successful tests of all combinations of system start-up and recovery/catch-up were also performed. The RMTP II protocol contains many tuning parameters and it was a significant effort to reach a set of parameters that performed well in all of the networking environments. The tests between Telcordia and the PIDC have been left running and as of four weeks later the data transfer was still running without a problem.

The principle problem encountered during the testing was caused by the Network File System. The Network File System at Telcordia is heavily loaded and some times took excessive time to return from read and write operations. This delay often led to drop of the member from the multicast group and then subsequent recovery once NFS returned. In addition, the RMTP II protocol implementation contained a bug in its method of tracking file descriptors. The problem was identified by Telcordia personnel working in conjunction with the vendor and it was repaired by the vendor.

Data Forwarding as an Alternative to Multicasting

In the absence of multicasting, the conventional approach for sending data to multiple receivers is to send the data to each receiver in a distinct unicast communications operation. There are several places in the network where the forwarding operation could be implemented. AlphaForward is the software currently available to transmit/relay data to multiple recipients. Use of AlphaForward would minimize the need to develop a new data-forwarding module. In this section, we will discuss briefly the relative merits of using multicast and AlphaForward to reach multiple recipients.

In the case of unicast CD-1, a new data-forwarding site can be inserted in the system with minimal change to the data sender and receivers. The data-forwarding site simply appears as a new receiver to the sender. The original receivers treat the forwarding site as the sender. The complexity of adding the new forwarding site is largely borne by the forwarding site. The forwarding site must run and maintain the various software processes that make up the AlphaForward operation of the continuous data protocol or software that provides similar functionality. The current AlphaForward software requires an Oracle database and 2-3 high-availability computers with 100's of Gigabytes of disk space. If the data forwarding operation is moved to a site within the GCI then the forwarding operation will also need to meet the GCI end-to-end requirements on data delivery reliability and timeliness. Although this solution limits the changes to the site performing the data forwarding operation, running a data-forwarding site is a complex operation. The data forwarding software is responsible for data buffering when a receiver is down, generation of

Format Frames on establishment of connection, negotiation of connections, and maintenance of LIFO Heaps containing the data. In addition, the forwarding site introduces a single point of failure into the system.

If instead a multicast-enabled version of the CD protocol is used to send data to multiple receivers, the changes required are more widespread. Any data sender that needs to multicast would need to be upgraded to a multicast-enabled version of the CD protocol. The receivers of the multicast data would also need to be running a multicast capable version of the CD software. In addition, IP multicast would need to be enabled in each of the routers between the sender and the receivers. Any firewalls between the sender and the receivers would need to be configured to pass the multicast traffic. Although these changes are more widespread, they can be made incrementally. Under the GCI contract all routers should be capable of accommodating multicasting, so enabling multicast should be a simple operation. The multicast and unicast capabilities of the CD protocol co-exist, so the software upgrades at the data senders and receivers can be accomplished on an as-needed basis. The firewall configuration could also be accomplished on an as-needed basis.

CONCLUSIONS AND RECOMMENDATIONS

During the past year, an experiment has been underway to test use of reliable multicast capabilities for transmission of continuous data in the Global Communication Infrastructure. For the experiment a version of the CD-1 protocol was multicast enabled. The experiment has demonstrated the feasibility of transmitting data in a multicast mode over the GCI. In the case of the Comprehensive Nuclear Test-Ban Treaty the sender could be the station and the receivers the International Data Center and one or more National Data Centers. The potential advantages of multicasting include a) the timely receipt of the data by the IDC and the host NDC and b) the simultaneous availability of the raw station data at, at least, two locations. The latter, by introducing redundant data paths, decreases the probability of loss of station data due to a potential failure of a single data receiver. This experiment is only one element of a needed more thorough assessment of the reliability and cost-effectiveness of introducing redundancies in the data transmission paths and the data sinks of the IMS. The next stage of the multicast experiment planned is installation of the multicast-enabled CD-1 software at the GERES IMS station, at the German NDC and at the IDC for further experiments with actual IMS station data. This stage of the experiment is waiting on installation of a GCI link to the German NDC. Negotiations regarding price for this installation have been on going between the GCI contractor and the German NDC with no resolution.

Current development of the CD-x protocol is proceeding in two complementary directions. Along with the work on a multicast enabled version of CD-1 there is also work to develop CD-1.1, which will add end-to-end reliability to the CD-1 protocol among other things. A possible future activity would be to combine the reliable multicast and the end-to-end reliability mechanisms into one CD-x protocol version.

REFERENCES

[1] D. Agarwal, P. Melliar-Smith, L. Moser, and R. Budhia, "Reliable Ordered Delivery Across Interconnected Local-Area Networks," Transactions on Computer Systems, vol. 16, no. 2 (May 1998).

[2] D. Agarwal, "Using Multicast in the Global Communications Infrastructure for Group Communication," in the Proceedings of the 22nd Annual Seismic Symposium, Las Vegas, Nevada, September 1999.

[3] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design", *ACM Transactions on Computer Systems* 2,4 (November 1984) pp. 277-288.

[4] D. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, 2nd ed, Volume I, New Jersey: Prentice-Hall, 1991.

[5] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, P. Sharma, L. Wei, “Protocol Independent Multicast,” IETF RFC 2362, available from ftp://ftp.isi.edu/in-notes/rfc2362.txt

[6] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, “A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing,” IEEE/ACM Transactions on Networking, December 1997, Volume 5, Number 6, pp. 784-803.

[7] J. Lin and S. Paul, “RMTP: A Reliable Multicast Transport Protocol,” IEEE INFOCOM '96, March 1996, pp. 1414-1424.

[8] J. Macker and W. Dang, “The Multicast Dissemination Protocol version 1 (mdpv1) Framework,” Technical white paper, US Naval Research Laboratory, available from http://tonnant.itd.nrl.navy.mil/docs/mdpv1.ps.

[9] K. Miller, K. Robertson, A. Tweedly, M. White, “StarBurst Multicast File Transfer Protocol (MFTP) Specification,” IETF Draft Specification, draft-miller-mftp-spec-03.txt, dated April 1998.

[10] D. Waitzman, C. Partridge, S. Deering, “Distance Vector Multicast Routing Protocol,” IETF RFC 1075, available from ftp://ftp.isi.edu/in-notes/rfc1075.txt

[11] B. Whetten, M. Basavaiah, S. Paul, T. Montgomery, N. Rastogi, J. Conlan, T. Yeh, “The RMTP-II Protocol,” Internet Draft, draft-whetten-rmtp-ii-00.txt and draft-whetten-rmtp-ii-app-00.txt, dated April 1998.

[12] “Multicasting in the GCI - A Report of Study Results,” available on the CTBT Expert's Communication System as CTBT/WGB/TL-3/9/Rev.1/Amend.1, 2 September 1999.