submitted to IEEE MILCOM 96' Conference

RELIABLE MULTICAST DATA DELIVERY for MILITARY NETWORKING

Joseph P. Macker
Naval Research Laboratory
Washington, DC 20375
macker@itd.nrl.navy.mil

J. Eric Klinker
Naval Research Laboratory
Washington, DC 2037
klinker@itd.nrl.navy.mil

M. Scott Corson
University of Maryland
College Park, MD
corson@isr.umd.edu

Abstract

Multicast networking support is becoming an increasingly important technology area for both commercial and military distributed or group-based applications. The underlying delivery mechanism for IP multicast is presently the User Datagram Protocol (UDP) or raw IP packets. At present, these mechanisms provide a "best effort" delivery service. Best effort implies that IP packets are treated with essentially equal weight, and while IP makes an effort to deliver all packets to their destination, packets may be occasionally be delayed, lost, duplicated, or delivered out of order. In the past such delivery mechanisms have worked fine for supporting traffic insensitive to occasional lost or missing data (e.g.. voice, video). An increasing variety of distributed multimedia applications are being developed in which a consistent and/or reliable data delivery of all or a subset of data packets is a critical performance factor. In future military tactical internetworks, situational awareness data will play a major role as a critical multicast application. Reliable group file transfer (e.g.. image dissemination) and interactive mission planning applications are also likely applications for military mobile units.

This paper presents a taxonomy of presently available reliable multicasting solutions. The protocols are classified in terms of performance issues and scalability. Using this taxonomy, reliable multicast solutions are considered for various military applications such as mission planning, Distributed Interactive Simulation (DIS), and situational awareness dissemination in a shared WAN environment.

Introduction

The current model for IP multicast delivery is increasingly inadequate for the variety of multicast applications being developed. In particular, providing a degree of reliability is critical for many applications. Currently, several reliable multicast solutions are available and many approaches are being developed. To aid in understanding the reliable multicast solutions that are available, a general classification, or taxonomy, of reliable multicast techniques is presented in the following section. This taxonomy is then applied to several example military applications to give the reader some idea of the issues that must be considered when selecting or designing a reliable multicast approach for a particular application class and scalability goal.

Application Requirements

Multicast applications require varying degrees of reliability and ordering. Understanding these requirements aids in classifying reliable multicast design.

Reliability Requirements

Application reliability requirements can be loosely defined as follows [51].

- *Best effort* reliability is similar to that which is provided by the UDP-based IP multicast delivery schemes most commonly present today. No reliable delivery is guaranteed.
- *Absolute* reliability is the most familiar requirement. It states that all packets in a session be reliably delivered to the receivers. This is the form of reliability that is commonly supported by TCP at the transport layer for unicast sessions.
- *Bounded latency* requires that each packet adheres to a specified lifetime over which the data is useful to the receiver. This is defined as an upper bound on its delivery latency. Packets arriving outside this timeframe are discarded. The common application requiring bounded latency is a video stream. Each packet has a "playback" time and any packet not meeting this deadline is discarded.
- *Most recent* reliability is reliable transmission where only the most recent data of a particular parameter is of interest. A simple example would be a service that provides reliable stock

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1996		**00-00-1996 to 00-00-1996**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Reliable Multicast Data Delivery for Military Networking	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

updates. If a particular stock update is lost, and a new update is received before a retransmission can occur, the old data is rendered useless. Thus, it is possible that the data may take on a value that is never known to some or all of the receivers. Most recent reliability is a common requirement for many data types in advanced distributed simulations and in situational awareness dissemination.

Ordering

Multicast applications may also be classified by their ordering requirements. These may be broadly classified into the following categories [51]:

- *Ordered* delivery means that packets are delivered to the receiver's application layer in the sequence that they were transmitted. This is the ordering classification delivered by TCP for unicast transmission [14].
- *Unordered* delivery allows the reliability mechanism to deliver the packets in any order.
- *Causally ordered* delivery further requires the reliability mechanism to maintain ordering across distributed processes [17]. For example if application A transmits m_1 to applications B and C, then B on receiving m_1 transmits m_2 to C, the reliability mechanism requires that m_1 is received at C before m_2.
- *Totally ordered* delivery specifies that multiple multicast streams from multiple senders are delivered sequentially to each receiver and are received in the same relative order at each receiver.
- *Causal, totally ordered* delivery specifies totally ordered delivery that does not violate causality.

Reliable Multicast Taxonomy

Reliable multicast protocols can be grouped into two broad classes, sender-reliable and receiver-reliable multicast. Each classification is based on the sender's knowledge of the multicast group and which party has the responsibility for state maintenance and the initiation of error correction [30].

Sender-reliable

In this approach, reliable delivery is primarily the responsibility of the sender. The sender monitors the reception state of each receiver through positive acknowledgment (ACK) and issues repairs upon error detection. This is a basic selective repeat approach. However, IP multicast implies no direct relationship between senders and receivers of multicast data. This severely hampers the ability of a sender to track and maintain reception state for each receiver. Even if each sender is made aware of all receivers, a severe ACK implosion effect is created at each sender when the number of receivers grows large (e.g., > 10 participants) [30].

This approach is appropriate when absolute control must reside at the sender (e.g., security reasons), but for most applications the approach does not scale due to the ACK implosion effect and the requirement of the sender to track state of all receivers.

Receiver-reliable

For receiver reliable multicast, reliable delivery is the responsibility of the receivers. Each receiver maintains reception state and requests repairs via a negative acknowledgment (NACK) when an error is detected.

Error detection is based on the receiver perceiving gaps in the data. This requires that individual packets be identified either with application level framing or generic transport sequence numbers as in TCP. Low latency gap detection requires frequent data transmission, otherwise "heartbeat" or "keep alive" transmissions are necessary.

In receiver reliable systems, mixed levels of reliability can be achieved at a receiver. A receiver may choose to NACK any missing data it requires to be reliable. The data packet may be encoded to indicate a reliability requirement. Encoding reliability at the application layer affords the most flexibility as many levels of reliability are possible.

There are several classes of receiver-reliable approaches that are discussed below:

Sender-oriented

In sender-oriented approaches an error detection at a receiver results in a NACK sent to the sender. While intermediate receivers may have received the data for which the NACK is issued, only the sender is involved in issuing repairs. This approach is appropriate when receivers cannot communicate with each other (perhaps for security reasons). However, such an approach ultimately limits scalability due to a NACK implosion effect at the sender for large receiver sets Thus, such an

approach is best suited for transmission of very large packets where a low ratio of NACK overhead to data content can be realized. This reduces the overall NACK implosion effect.

To explore the effects of unsuppressed NACK implosion, we simulated a set of symmetric multicast trees with an independent probability of packet loss *p* due to congestion at tree nodes. Each multicast tree of depth *d* and fan-out *n* contains members at branch and leaf nodes. Upon a packet loss at a node, all children of this parent are declared to all produce a NACK message The total number of NACK messages is obtained and averaged over a large set of trials. This produces an expected value of NACKs for a given tree and failure probability [51].

Figure 1. shows the average results of 100,000 trials for nodes with a fan-out, n=4, and tree depths, d = 4,5,6. The NACK implosion effect is quite apparent as the size of the tree and the packet loss per node increases. Even in cases where raw link error rates are very low, the probability of packet loss can be quite high, due to router congestion and other effects.

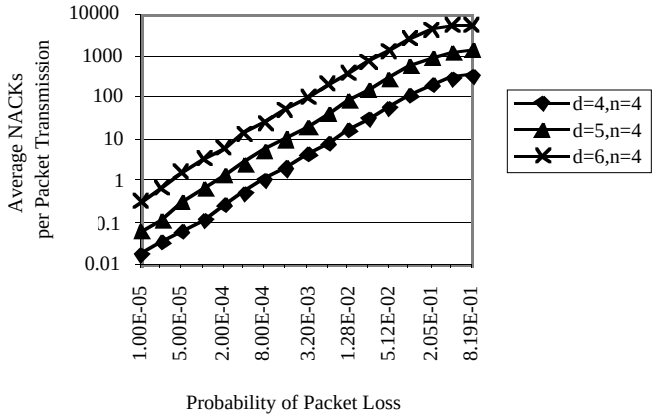

Figure 1: NACK implosion simulation

Flat Receiver-Oriented

In a flat receiver-oriented approach to reliable multicast, receivers can communicate with each other to assist in error recovery. Each receiver caches data for some time or for the entire session. When an error is detected at a receiver, a NACK is issued which other receivers can hear since it is multicast to the group. When a receiver that has correctly received and cached the missing data receives the NACK a repair can be issued. This in itself would not reduce the NACK implosion effect since the NACK is sent to the whole group and any receiver detecting an error issues a NACK. To make this scheme work well a distributed repair scheme is required which suppresses the number of control messages required per repair.

When a receiver detects an error, it is likely that other downstream and equidistant receivers will also experience the error. In addition, equidistant (in terms of delay) receivers will detect the error at roughly the same time. To reduce the chance of all such receivers issuing redundant NACKs at once, each receiver sets a random timer upon error detection. When the timer expires, if a NACK for the missing data has not already been heard, the receiver issues a NACK. When a receiver that has correctly received and cached the missing data hears a NACK, another random timer is set. When this timer expires, if a repair has not been heard, one is issued. This reduces the number of redundant repairs that might be sent by equidistant receivers simultaneously receiving NACKs [9,10,38,49].

Deterministic suppression is also possible along a linear topology [38,51]. This is useful when downstream receivers also detect the same errors as upstream receivers. By accurately estimating the delay between receivers, the uniform distribution of the downstream random timers can be adjusted to produce longer delays. Thus, it is likely that a downstream receiver will observe the NACK of an upstream receiver before issuing his own NACK.

Since most networks exhibit both linear and star (equidistant) characteristics, a combination of randomized and deterministic NACK/repair suppression should be used for a flat, receiver-oriented reliability scheme [38].

A drawback to a flat receiver-oriented approach is that NACKs and repairs are global in scope. Thus, they consume bandwidth for the whole group even for isolated packet losses. Enhanced localized scoping of repair messages is possible and can alleviate this effect [38,50,51]. In general a flat, receiver-based approach is highly fault tolerant and scaleable. Its biggest disadvantage is in the requirement to cache state at all receivers. This is not really a weakness for some applications, such as an interactive whiteboard [9,10], in which state may be maintained regardless. However, for long-lived applications requiring significant buffering this could quickly become a nuisance.

Hierarchical, receiver-oriented

To improve scalability performance and allow for distributed state garbage collection and more

organized repairing schemes, it is possible to introduce a hierarchy into a reliable multicasting. Several hierarchical, reliable multicasting approaches have been proposed. One such approach [36] forms a hierarchy of caching loggers, to which a receiver NACKs for a repair. Another [51] forms a self organizing shared repair tree. Localized scoping in this approach limits repair dissemination and the hierarchy aids in buffer management. A similar approach is the Tree-based Multicast Transport Protocol, TMTP [50] which forms source based trees for a repair hierarchy. The scoped repairs are used for state management and error recovery. There are limited performance results to understand the various performance tradeoffs of each approach. In summary further study of this topic is warranted.

This is not an issue for any application requiring bounded latency reliability since the sender is free to discard data that cannot meet the reception deadline. However, the validity of absolutely reliable data *never* expires and *must* be delivered. Thus, the sender cannot arbitrarily discard the data.

Supporting absolute reliability requires some form of ACKing mechanism from the receivers which allows the sender to periodically flush its buffers. This ACKing mechanism can be used in conjunction with a more general NACK error recovery approach and should be as infrequent as possible to reduce ACK implosion and general overhead. This mechanism also requires the senders to have knowledge of the set of receivers at any given time.

Reliability Mode	Error Detection	Repair State	Scalability
Sender	Sender	Sender	Low: due to ACK implosion
Receiver	Receiver	see Table 2	see Table 2

Table 1: Reliable Multicast Approaches

Repair Orientation	NACK Scheme	Repair Scheme	Scalability
Sender-Oriented	Receivers NACK to Sender	Sender issues repair to group	Moderate: Least scalable of receiver-reliable approaches.
Flat, Receiver-oriented	Receivers NACK errors to group	Receivers cache data and can issue repairs	High: Limits Nack implosion effect, but distributed buffering required.
Hierarchical	Receivers NACK to some hierarchical node or group	Hierarchical nodes successively responsible for buffering and issuing repairs	Very high: Excellent scalability and limited network overhead. Allows for more controlled buffer management.

Table 2: Receiver Reliable Approaches

Supporting Absolute Receiver-Reliable Service

Support of absolute reliability in a receiver-reliable approach imposes constraints on senders. Since senders are not tracking receiver reception state, at any point in the future a receiver may require a retransmission. Strictly speaking this requires the sender to buffer data indefinitely. This represents a problem if the sender is participating in a long-lived session given finite storage capacity.

Thus, any scheme supporting absolute reliability represents a mixed requirement of both sender-reliable and receiver-reliable multicast.

Summary

Tables 1 and 2 summarize the taxonomy of reliable multicast protocols presented in this section.

Military Applications

In this section, we consider some military applications that will require or might benefit from reliable multicasting capabilities. In each case, we examine some of the requirements the application

demands from a reliable multicast protocol, and discuss which protocol class (as per the taxonomy) is a "best fit" for that application. The goal is not to mandate an approach for each of the applications, but rather highlight some of the issues that must be considered when selecting or designed a reliable multicast approach for a particular application.

Situational Awareness Applications

Situational awareness applications will likely require reliable dissemination of information to multiple parties. To the extent that this dissemination is desired will compound the impact scalability issues will have. If the situational awareness data is sent to the brigade or battalion level, scalability is not as important as if the data is transmitted to the individual soldier. Since survivability will be affected, the data must be reliably sent. Low to moderate available bandwidth will mandate that multicast be used. However, there are aspects of this data lead to a mixture of reliability requirements. Clearly, there will be situational data that is "most recent" reliable. This may be data with an object-oriented characteristic such as tracking information of an oncoming enemy. If a previous position report is lost, but a newer report is received, the old report is often rendered useless. In addition to the object-oriented form of reliability, a component of data exists that must be absolutely delivered reliably. This is message-oriented information such as a "one-time" report of a hazard (say a minefield) which should absolutely get to all interested parties.

Since the number of receivers is likely to be large (if taken to the individual level), receiver reliable schemes seem most appropriate to improve scalable performance and with appropriate NACK suppression will reduce the likelihood of control message implosion effects. Since the forming of hierarchies will be difficult, as most of the parties will be highly mobile, a flat receiver reliable approach is simplest and most robust. To generate the granularity of reliability required (most recent vs. absolute) different multicast groups can be used, the multicast group that has an absolute requirement can be perceived as a priority channel.

Imagery Dissemination:

Imagery dissemination applications can be characterized by large infrequent packet transmissions (with some exceptions). For the most part, the data should be considered absolutely reliable. If the number of recipients is small, a sender reliable approach is acceptable. If there are too many receivers such that ACK implosion is a severe problem, then a receiver-reliable approach should be adopted. Since there will likely be only one sender of images, a sender oriented, receiver reliable approach might be considered, but only if a handy mechanism for group awareness at the sender is possible. This requires only the sender to buffer potentially large amounts of data. Since there is no latency bound on the data being transmitted, the sender might have to buffer data indefinitely. Thus, a mixed sender/receiver reliable scheme might be considered to alleviate this problem. Again the periodic ACKs from receivers require the sender to track a certain amount of receiver state, and thus be group aware. Clearly, this application is afforded great flexibility when considering a reliable multicast approach, as many different approaches will work per given scenario.

Command and Control:

Many command and control applications are distributed over several mobile platforms. The users of these distributed applications require a consistent environment in which to make correct decisions. Since these applications share information among many participants, they can benefit from multicast communication. The requirement to maintain a consistent environment leads the application to reliable multicast.

As in situational awareness applications some of the data that is transmitted is "most recent" reliable whereas some is absolutely reliable. If a track update is lost, but a new track update is received before the old is retransmitted, the old update is rendered useless. However, a one-time event such as a submarine contact might require absolute reliable transmission to all participants.

Given the large number of participants sharing the data, sender reliable schemes will likely not scale. Many command and control applications are considering the use of voice and video services to augment the traditional C2 functions (becoming C3 and C4I systems). Since this data has bounded latency constraints, indefinite buffering at the senders is not required. The "one-time" reports that require absolute reliability may be infrequent enough or small enough such that no positive ACKs from receivers are necessary to clear buffers. As a result, ACK implosion can be altogether avoided by not adopting any sender-reliable approach.

Distributed Interactive Simulation

Distributed interactive simulation can (and does) benefit from a reliable multicast solution. The majority of DIS traffic is characterized by frequent position updates to some or all of the participants in the simulation. Other events such a terrain or environmental events may be "one-time" occurrences that are never again transmitted. Lastly, there are some simulations that are incorporating command and control data over the same networks to enhance the realism of the simulation. This type of data is "message-oriented" and may be characterized as multicast file transfers over the simulation network. However, we again have a situation where multiple levels of reliability must be delivered. The "most recent" reliable data corresponds to entity state packets (which are changing frequently) and the absolutely reliable data may include things as Fire messages, detonation messages, terrain updates, and environmental updates. A consistency protocol is used in these cases to maintain a reliable, coherent simulation [49]. Message-oriented reliability features are being considered as extensions to the consistency protocol approach.

Given the severe scalability issues inherent in DIS applications, only receiver reliable schemes should be considered. Even then, NACK implosion will be a serious problem when multicast group membership grows large. The problem is compounded by the fact that most receivers are also senders of data.

Multicast Key Management

This may be an application where sender-reliable is the best fit. The inherent nature of the problem requires that the sender know of all receivers up front, although one can envision a hierarchical approach for distribution management to improve scalability. If infrequent transmission is a characteristic of the approach, then allowing explicit positive acknowledgment from each receiver can enhance robustness and the security properties of the scheme. The requirements for multicast key management are not entirely clear at present and this in excellent area for future study.

Mission Planning Systems

Mission planning systems can benefit from reliable multicast through the dissemination of Air Tasking Orders (ATOs) via reliable multicast. Such traffic can be characterized as large infrequent packets. This traffic is similar to that of imagery dissemination discussed above. This application is afforded the same flexibility when selecting a reliable multicast approach for this data.

If the mission planning systems are further augmented with some form of "whiteboard" application the requirements change significantly. Whiteboard applications can benefit from infinite buffering during the session. When a new member joins the whiteboard session, any of the receivers within the session can send the current whiteboard pages since all data has been buffered. The data is stored until it is erased by a participant.

Conclusions

This paper serves as a general guideline for classifying reliable multicast protocols. To be sure, some protocols can be envisioned that exist beyond the scope of the general taxonomy put forth, but the major issues such as scalability will still apply. The military examples presented in the previous section should aid in applying reliable multicast approaches to other applications.

References

[1] G. Parulkar, D. Schmidt, and J. Turner. "aitpm: a Strategy for Integrating IP with ATM". Technical report, Department of Computer Science, Washington University, 1995.

[2] S. Deering. "Host Extensions for IP Multicasting". Internet RFC 1112, August 1989.

[3] D. Waitzman, S. Deering, and C. Partridge. "Distance Vector Multicast Routing Protocol". Internet RFC 1075, November 1988.

[4] J. Moy. "Multicast Extension to OSPF". Internet Draft, September 1992.

[5] T. Ballardie. "Core Based Tree (CBT) Multicast - Protocol Specification". Internet Draft, June 1995.

[6] S. Deering, D. Estrin, et al. "Protocol Independent Multicast (PIM): Motivation and Architecture". Internet Draft, January 1995.

[7] M. Handley, J. Crowcroft, and I. Wakeman. "Hierarchical Protocol Independent Multicast (HPIM)". Internet Draft, November 1995.

[8] A. Thyagarajan and S. Deering. "Hierarchical Distance-Vector Multicast Routing for the Mbone". In Proc. ACM SIGCOMM, pages 60--65, 1995.

[9] V. Jacobson. "A Portable, Public Domain Network Whiteboard", April 1992. Xerox PARC, viewgraphs.

[10] V. Jacobson. "Multimedia Conferencing on the Internet", August 1994. Tutorial 4, ACM SIGCOMM 94.

[11] S. McCanne. "A Distributed Whiteboard for Network Conferencing", May 1992. UC Berkeley CS 268 Computer Networks term project.

[12] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. "RSVP: A New Resource ReSerVation Protocol". IEEE Network Magazine, pages 8--18, September 1993.

[13] D. Clark, S. Shenker, and L. Zhang. "Supporting Real-time Applications in an Integrated Services Packet Network: Architecture and Mechanism". In Proc. ACM SIGCOMM, September 1992.

[14] J. Postel. "Transmission Control Protocol - DARPA Internet Protocol Program Specification". Internet RFC 793, September 1981.

[15] H. Garcia-Molina and A. Spauster. "Ordered and Reliable Multicast Communication". ACM Transactions on Computer Systems, 9(3):242--271, August 1991.

[16] K. Birman, A. Schiper, and P. Stephenson. "Lightweight Causal and Atomic Group Multicast". ACM Transactions on Computer Systems, 9(3):272--314, August 1991.

[17] L. Lamport. "Time, Clocks and the Ordering of Events in a Distributed System". Communications of the ACM, 21(37):558--565, July 1978.

[18] K. Birman. "The Process Group Approach to Distributed Computing". Communications of the ACM, December 1993.

[19] A. Schiper, J. Eggli, and A. Sandoz. "A New Algorithm to Implement Causal Ordering". In Proc. of the 3rd Intern. Workshop on Distributed Algorithms, Lecture Notes on Computer Science 392, pages 219--232, Springer-Verlag, New York, 1989.

[20] C. Fidge. "Timestamps in Message Passing Systems that Preserve Partial Ordering". In Proc. 11th Australian Computer Science Conf., pages 56--66, 1988.

[21] F. Mattern. "Time and Global States in Distributed Systems". In Proc. of the Intern. Workshop on Parallel and Distributed Algorithms, North Holland, Amsterdam, 1989.

[22] K. Birman. "A Response to Cheriton and Skeen's Criticism of Causally and Totally Ordered Communications". Technical Report 1390, Department of Computer Science, Cornell University, October 1993.

[23] J. Chang and N. Maxemchuk. "Reliable Broadcast Protocols". ACM Transactions on Computer Systems, 2(3):251--275, August 1984.

[24] B. Whetton, T. Montogomery, and S. Kaplan. "A High Performance Totally Ordered Multicast Protocol". In Proc. INFOCOM, Boston, MA, March 1995.

[25] M. Kaashoek, A. Tannenbaum, Hummel, and Bal. "An Efficient Reliable Broadcast Protocol". Operating Systems Review, October 1989.

[26] S. Armstrong, A. Freier, and K. Marzullo. "Multicast Transport Protocol". Internet RFC 1301, February 1992.

[27] P. Melliar-Smith, L. Moser, and V. Agrawala. "Broadcast Protocols for Distributed Systems". IEEE Transactions on Parallel and Distributed Systems, pages 17--25, January 1990.

[28] XTP Forum, Santa Barbara, CA. "Xpress Transport Protocol Specification, XTP Version 4.0", March 1995.

[29] V. Jacobson. "Congestion Control and Avoidance". In Proc. ACM SIGCOMM, August 1988.

[30] S. Pingali, D. Towsley, and J. Kurose. "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols". In Proc. INFOCOM, San Francisco, CA, October 1993.

[31] A. Koifman and S. Zabele. "RAMP: A Reliable Adaptive Multicast Protocol". In Proc. INFOCOM '96 (to appear), San Francisco, CA, 1996.

[32] R. Braudes and S. Sabele. "Requirements for Multicast Protocols". Internet RFC 1458, May 1993.

[33] B. Dempsey, M. Lucas, and A. Weaver. "Design and Implementation of a High Quality Video Distribution System using XTP Reliable Multicast". Technical report, Computer Science Department, University of Virginia, 1994.

[34] D. Clark and D. Tennenhouse. "Architectural Considerations for a New Generation of Protocols". In Proc. ACM SIGCOMM, pages 201--208, September 1990.

[35] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing". In Proc. ACM SIGCOMM, August 1995.

[36] H. Holbrook, S. Singhal, and D. Cheriton. "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation". In Proc. ACM SIGCOMM, August 1995.

[37] D. Estrin, S. Shenker, and D. Zappala. "Routing Support for RSVP". Internet Draft, June 1995.

[38] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Extended Report". Technical report, Lawrence Berkeley National Laboratory, November 1995. URL ftp://ftp.ee.lbl.gov/papers/wb.tech.ps.Z.

[39] K. Birman and T. Clark. "Performance of the Isis Distributed Computing Toolkit". Technical Report 1432, Department of Computer Science, Cornell University, June 1994.

[40] IEEE Standard 1278-1993. "IEEE Standard for Information Technology-Protocols for Distributed Interactive Simulation Applications, Entity Information and Interaction", March 1993.

[41] D. Baker and A. Ephremides. "The Architectural Organization of a Mobile Packet Radio Network via a Distributed Algorithm". IEEE Transactions on Communications, COM-29:1694--1701, 1981.

[42] C. Li. "Clustering in Packet Radio Networks". In Proc. IEEE Int. Conf. on Com., Chicago, IL, section 10.5.1, 1985.

[43] M. Gerla and J. Tsai. "Multicluster, Mobile, Multimedia Radio Network". ACM/Baltzer Wireless Networks Journal, 1(3):255--266, October 1995.

[44] P. Krishna, M. Chatterjee, N. Vaidya, and D. Pradhan. "A Cluster-based Approach for Routing in Ad-hoc Networks". In Proc. 1995 Mobile and Location-Independent Comp. Symp., pages 1--10, 1995.

[45] M. Post, A. Kershenbaum, and P. Sarachik. "A Distributed Evolutionary Algorithm for Reorganizing Network Communications". In Proc. MILCOM, pages 133--139, 1985.

[46] A. Bhatnagar and T. Robertazzi. "Layer Net: A New Self-Organizing Network Protocol". In Proc. MILCOM, pages 845--850, 1990.

[47] M. S. Corson and A. Ephremides. "A Distributed Routing Algorithm for Mobile Wireless Networks". ACM/Baltzer Wireless Networks Journal, 1(1):61--82, February 1995.

[48] V. Sunderam. "PVM: A Framework for Parallel Distributed Computing". Journal of

[49] D. Van Hook, J. Calvin, J. Smith, " Data Consistency Mechanisms to Support Distributed Simulations," 95-12-059, Twelfth Workshop on Standards for the Interoperability of Distributed Simulations, March 13-17, 1995.

[50] R. Yavatkar, J. Griffioen, M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," Technical report, Department of Computer Science, University of Kentucky.

[51] M. S. Corson, J. Macker, "Reliable Scalable Multicast and a Self-Organizing Approach for Long-Lived Sessions", submitted to the IEEE Journal on Selected Areas in Communications, January 1996.