Multisites Coordination in Shared Multicast Trees *

H.-P. Dommel and J.J. Garcia-Luna-Aceves
School of Engineering
Computer Engineering Department
University of California
Santa Cruz, CA 95064, USA
{peter,jj}@cse.ucsc.edu

Abstract *The majority of today's Internet applications relies on point-to-point transmission. In recent years, however, multicast transmission has become the foundation for such applications as multiparty video conferencing, distributed interactive simulations, and collaborative systems. We describe a novel protocol to coordinate multipoint groupwork in the IP-multicast framework. The protocol supports Internet-wide coordination for large and highly-interactive groupwork, relying on transmission of coordination directives between group members across a shared end-to-end multicast tree. We also describe how addressing extensions to IP multicast can be put to use for our multisite coordination mechanism.*

Keywords: group coordination, multicast, collaboration.

1   Introduction

Internet computing is gradually migrating from the standard unicast transmission model to multicasting. In the IP multicasting model [3], a source needs to send a packet only once to the network interface, and multicast routers replicate the packet on its transmission path to multiple receivers. The Internet Group Management Protocol (IGMP) allows a host to join a multicast group by informing its local router to forward multicast traffic for this group to the leaf subnetwork where the host resides. Protocols such as DVMRP, MOSPF, and PIM [7] perform the construction of multicast delivery trees and enable packet forwarding between routers.

With IP multicast, no guarantees are given for reliable or order-preserving delivery of packets, and a message is delivered on a best-effort basis to all members of a multicast group. These shortcomings have spurred much research on reliable multicast between end hosts, and on mechanisms to refine IP multicast, such as using addressing information to enable subcasting or anycasting [9]. Subcasting delivers or retrieves data between a source and select members of a multicast group, and anycasting transfers data to any one member of a group, for example the nearest proxy from a group of servers. While IGMP targets group membership, and multicasting routing protocols are concerned with delivery, no protocols exist to tackle an emerging problem of multisite communication, which is group coordination. This problem surfaces especially for tightly-coupled sessions featuring explicit conference membership control.

Group coordination denotes services to support distributed hosts in coordinating their joint activities, including synchronization of flows from different sources, ordered delivery of distributed event information, and the concurrent use of and access to shared resources, referred to as floor control [4].

Early paradigms of group coordination, mutual exclusion [13] or concurrency control [2], have been restricted to discrete data domains rather than multimedia contents, using locking to manifest control, and have not been deployed on an Internet scope. We discuss a novel group coordination scheme, called Aggregated Coordination Protocol (ACP), which operates on a shared multicast tree, benefiting from the underlying tree structure to store and

		Form Approved
	Report Documentation Page	OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**
4. TITLE AND SUBTITLE **Multisites Coordination in Shared Multicast Trees**		5a. CONTRACT NUMBER
		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) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		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 **The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **7**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

forward coordination primitives between hosts in different multicast groups on the tree. ACP coordinates distributed activities via message passing, and manifests control by ephemeral permissions rather than actual locks, allowing control over continuous media flows as well as discrete data.

Online group coordination in relation to face-to-face meetings has been studied by Isaacs *et al.* [8], showing that greediness for time and bandwidth on behalf of non-cooperative users, as well as lack of social cues such as eye contact contribute to coordination problems. While geared toward supporting humans in their interactions, the concept of group coordination also applies to agent-based interaction. Methods to mediate resource contention at the user interface level have been discussed for example by Ellis and Gibbs [5]. Many existing systems for online collaboration are proprietary, sparsely documented, and limited to local area networks, or sessions with few users. Abdel-Wahab discussed an early prototype of a token-based control mechanism for a shared workspace. An alternative approach was proposed by Aguilar *et al.* [1], in which distributed, task-activated floor control serves as a high-level analogy to collision-sensing in channel access.

Ziegler *et al.* [17] researched packet-switched voice conferencing in a broadcast and unicast setting. ITU standards 120 [14] and 320 for video conferencing are circuit-switched and designed for conferences with few users. Yavatkar [16] proposed the MCP coordination protocol between concurrent media flows based on token passing. A framework for hierarchical collaboration, looking at bandwidth and delay issues, was presented by Vin *et al.* [15]. Raymond [13] discussed a mutual exclusion algorithm operating on a static logical propagation tree, in which nodes maintain a pointer to the neighbor node that leads to the current token holder with access to the critical section. The algorithm by Neilsen and Mizuno [11] adds a dynamic link between a requester and the token holder to avoid backtracked routing of reply messages between a source and target node.

We are interested in the question, how the routing and end-to-end geometry used for multicasting data can be used effectively for coordinating the activities among individuals and groups in sessions of Internet scope. The rest of this paper is organized as follows: Section 2 presents the system model and assumptions. Section 3 discusses ACP in its operation and correctness. Section 4 summarizes the benefits of tree-based group coordination.

2 Model and Definitions

We define a coordination session $C_S = (H, L)$ as a computer network with hosts H and links $L \subset H \times H$. Communication is by message passing only, and we assume that messages eventually arrive correctly. Each host in a session is client and server for coordination primitives (CP) to other hosts. CPs are issued between hosts to synchronize their joint tasks, to implement causal or total ordering in distributed events, and to mitigate access to shared, but exclusive resources. Coordination management must be aligned with membership operations such as joining, leaving or splitting groups.

The entities involved are users (processes), resources at the various hosts, and the CPs coordinating them. Users assume social roles (moderator, panelist, student), and both users and agents assume control roles (controlling, who may work with a resource, or holding the right to work with it). We call the host controlling access and operations of a resource the *floor controller* (FC) for that resource. The host being permitted to access the resource at a given moment is called *floor holder* (FH). Resources can be located at a specific end host (camera), in replicated form at every host of a multicast group handling the same resource (telepointer), or in the network (voice channel). We distinguish between generic CPs ("grant", "release", "open") and resource or media-specific CPs ("rotate left", "zoom in"). The temporary privilege to work with a multimedia resource is also called *floor*. CPs contain the sender id, the receiver ids, the time of creation, the allowed duration, and an optional priority value.

Control over shared resources can be centralized, distributed or a hybrid of both. It can be performed successively by individual session members, partitioned, where various session hosts contribute different control functions, or democratic, where consensus is achieved by

negotiation, yielding a new consistent control state. A host holds a floor in his turn for a time interval T, which can be preset or timed out. In our model, we assume that the FC role is associated with a specific host, but it may infrequently rove among hosts. The FH changes at every turn. We assume that each host in a session runs the same coordination protocol, serving requests from other hosts and transmitting requests for resources placed by users or their processes.

3 Aggregated Coordination Protocol

3.1 Description

The Aggregated Coordination Protocol (ACP) operates on a control tree, consisting of three main types of nodes: *holder nodes* host the FH, operating on a resource and being transmission sources; *control nodes* host the FC for a specific resource, and are addressed by other nodes asking for a floor; *target nodes* receive updates of the operations by a FH. Nodes on the path from a holder to its targets are referred to as hop nodes. Leaf nodes terminate downward forwarding of control information in the tree.

CPs are disseminated across a single shared tree connecting all hosts in a session. The tree corresponds to a single shared acknowledgment tree for concurrent multicasting, allowing multiple sources instead of building separate dissemination trees per source and multicast group. The tree can be a working copy of the reliable multicast tree prepared by a protocol such as Lorax [10]. We outline tree maintenance for the case that group coordination is deployed in a network complementing an existing underlying reliable multicast tree.

Various mechanisms for initiation, joining, and advertising of collaborative sessions exist. We assume that one host, representing itself or a multicast group, initiates a session and advertises the session description, multicast address and media in use in a session directory [6]. The directory serves as a rendezvous interface, which allows other hosts to join via a call-up mechanism. The control tree is grown from the initiating node as the root, and other hosts joining the session are considered first off-tree, unicasting request-to-join messages to the inviting root node based on addressing information provided by the session directory. A TTL (time-to-live) field in the join packet restricts the session scope.

A successful adoption of a child node to the control tree is confirmed with a bind message. Each newly joined host, as the root of its subtree, locally advertises invitation-to-join messages. It may also be the case that separate subtrees may fission together to create a joint control tree. Each node in the tree has a maximum degree D, which must be high enough to reflect the session structure, but not exceed the capacity of a host to efficiently serve its children. Furthermore, if the tree serves also as a media mixing hierarchy [12], the permissible height must satisfy the end-to-end delay constraints.

Open sessions with dynamic membership may incur frequent joining and leaving, or accidental withdrawal of hosts from the control tree. When the root leaves the control tree, the eldest child in the subtree is the designated new root. Age may be determined by location, joining time, or address labels. A hop node, which lost contact with its parent for a timeout larger than possible congestion delays, must rejoin the tree as described. CPs from a node identified as lost are held at the processing host for a timeout and discarded if the host does not reappear. A lost and rejoined host must resend its pending CPs.

Routing of CPs in the tree is performed as follows: if a target node is in the subtree of a node, the CP is routed downward the subtree branch where the target resides, otherwise it is sent upward to its parent node. This operation could be performed by using only directionality information on where a target is located, as proposed in the previously mentioned mutual exclusion scheme by Raymond [13]. However, in contrast to a logical geometry used to propagate critical section requests, our protocol must disseminate CPs involving many resources, media types, and their coordination properties across an infrastructure prescribed by the actual underlying multicast tree.

We need therefore a more expressive addressing and control mechanism for multisite coordination among multicast groups, because many nodes may assume FC or FH roles during

a dynamic collaboration session. ACP assigns recursively and top-down unique prefix labels to each node joining the control tree, such that a child node label contains as prefix the label of its parent. For example, using a binary alphabet, the root receives label **1**, its children are numbered **10** and **11**, etc. The idea of using binary labels has first been used for routing of instructions in multiprocessor systems and has recently been applied to multicast routing and reliable multicasting [9]. Labels allow nodes to be addressed individually, although being part of one or more possibly overlapping multicast groups. Also, hosts can be placed in the tree independent of their membership in different multicast groups, in contrast to other approaches, which demand that the tree organization reflects group membership.

Each CP contains the source's label, the target label(s), a sequence number, a local timestamp, a session wide unique resource id obtained from the session directory, and a floor id, which denotes the temporary access permission or an activity descriptor for a resource. The structure of a CP packet is shown in Figure 1. $CPid$ identifies the coordination primitive and the type of operation, characterizing various resource modalities. TTL indicates the scope of the CP. Opt is reserved for priority codings. The timer and sequence numbers tag the CP uniquely in the session and event space. $Checks$ is the checksum field. The $Sourceaddr$ and $Targetaddrs$ fields contain the labels for the sender of a CP and its target nodes.

0 2		12	20	31
V	CPid	Typ	TTL	Opt
Timest		Seq#		Checks
Source addr				
Target addrs ...				

Figure 1: Packet header fields for coordination primitives (CPs).

Although the initial source and root will change over the course of a session, the branching properties of the positional tree will not change, when the tree is virtually rehung with shifting of control roles. CPs such as requests for the floor on a resource are propagated across the tree using aggregation. This mechanism corresponds to the solution for the Ack implosion problem, used in reliable multicast to limit the feedback of receivers to a source on lost or corrupt packets. Aggregation limits the control process to local groups, rather than letting CPs that can be satisfied locally, flow back all the way to the holder or controller node. ACP packets are assumed to be transferred reliably, which is guaranteed by the underlying reliable multicast protocol. If ACP performs independently, it needs to supply its own reliability mechanisms.

Hosts need to maintain locally the following state: the resource ids shared from local to the remote hosts, and the remote resource ids accessed locally, together with their CPids; a state table indicating which resources are locally available or held remotely, together with the id of the remote FH; and a request queue $ReqQ$ which collects successive CPs from different hosts (the queue is limited by the number of nodes in the session). If a hop node receives the same CPs from different nodes, it aggregates them into one CP and checks, if a response to these request can be satisfied locally by polling its own state and the state of neighboring nodes. Otherwise the composite CP is self-routed up or down in the tree toward the target node(s). This implies that the number of CPs required to coordinate nodes decreases as the request activity increases, because requests are not sent further, if a hop node is reached that already processed the CP. The target address may also contain the name of a multicast group, which is then resolved into its members locally at the primary receiver node for this group. A joining node retrieves the current control state for resources of concern by polling its parent node.

In addition, each node maintains a FIFO queue of pending CPids, identified by the senders' labels. A hop node receiving a floor compares the label of the elected node with the head of the queue, and self-elects if its own label matches the head, or forwards it in the routing procedure outlined above. A control node receiving a request responds immediately to the request by sending back a grant message to the requester, if its local queue is empty, or it appends it to its queue. When control shifts from a node to another one, the pending request queue is transferred to the new control node and its new address is multicast to all groups sharing the associated resource.

3.2 Example

Consider a scenario where three hosts from three different multicast groups MG1 - MG3 must coordinate access to a shared resource they are contending for. Figure 2 depicts a snapshot of the protocol operation in a ternary tree.
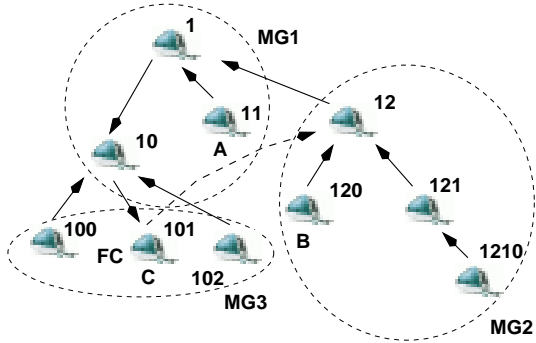

Figure 2: Snapshot of ACP operation among multicast groups MG1 - MG3.

Assume that hosts **12**, **100** and **11** all request the floor held by FC at location **101**. All request packets need to be routed along branches of the tree to **101**. The prefix property of the labels allows self-routing of these packets. For example, assume that all hosts are informed that host **101** is FC. Host **12** compares its label with the target label. Its prefix matches (1), but the second identifier indicates that the FC is on subtree 0. The request packet is hence sent upward to host **1**, which compares its label with the target, and, detecting that **101** is one of its children, sends the packet to host **10**, whose label matches the prefix of **101**.

This node performs the same comparison and the packet ultimately arrives at FC, which finally grants the floor to node **12**. The forwarding of control directives is aggregated, i.e., multiple requests for the same information from different nodes in the tree are assembled in a hop node in the tree, that lies on the path to the target, and are forwarded combined. This limits control traffic and unnecessary propagation of requests that for example cannot be satisfied at a FC node. FC is hence liberated from the need to communicate with every host in the session, and deals only with relevant requests reaching it from neighbor nodes by self-routing.

3.3 Correctness

We outline correctness and resilience properties of ACP. Assuming that CPs are sent reliably, a node can receive the privilege associated with a CP only if sends a request to the node controlling the associated resource. A message enacting the CP (e.g., grant-floor) can only be sent by a controlling node. The addressing labels in the tree are unique and a CP is only issued for a specific requesting node. Provided that there is a single holder node for a CP *a priori*, the protocol will hence continue to assign a CP to exactly one node.

If several nodes send a CP and another node holds the privilege associated with it for an infinite time, a deadlock exists. Reasons may be that the wrong node or no node holds the privilege, it is unreachable, or the propagation of a request fails prematurely. Assuming finite FIFO request queues at each node, a request reaching a control node will eventually be served. For a tree of height $h = 2$, we have a star-based scheme with $n = D + 1$ nodes. A floor will rove between the $D + 1$ nodes. Using induction over the height of the tree, we assume that the liveness argument holds for any height l with $2 \leq l < h$. A tree of height h has one additional level, from where CPs can be sent and must be replied to. Because of the acyclic tree geometry, it is impossible that a CP can be outrun by a moving floor privilege. Assume that a node in the additional level of the tree (the root, or a leaf level of a subtree) sends a CP. The addressing semantics of the protocol ensures that messages are self-routed to this level of the tree and will eventually reach the target node.

Finally, although privilege passing may progress during a session, a node may starve by being exempted indefinitely from the allocation process. However, the addressing semantics of ACP and the finite FIFO queue forbid that accepted CPs at a control node are ignored indefinitely. Based on these qualitative arguments we conjecture that ACP is correct.

3.4 Fairness

Fairness refers to the frequency and duration, by which nodes acquire a privilege on the average for a given period, which is for example the session lifetime, or the lifetime of a

shared resource in a session. Network latency, geographic distance and location of nodes, or varying host capabilities can all be factors in causing uneven dissemination patterns of CPs and unfair allocation of floors. Leaf nodes take more time to propagate their requests across the root to a node on the other side of a tree, than nodes just below the root. Shared trees also do not provide shortest paths between a source and its receiver set [7], which may cause increased latency in CP transfer. It is hence important to establish service policies, which counteract these factors. One simple solution, a "least-recently-served" policy, can be enacted by letting each node maintain a local record of the most recent CPs and their originating nodes. Those nodes are serviced first, which do not appear on the list, or appear last in time or frequency of service.

3.5 Resilience

Previously, we assumed that transfer of CPs and accounting of control information among nodes is failure free. Even we if assume reliable multicasting to ensure that CPs are eventually transferred, the control apparatus may need additional recovery mechanisms to ensure consistency. This applies to regular node failure, control node failure, link failure, or token loss or duplication. Such exceptions can be preempted by redundant dissemination of status information, or by detection of loss and recovery. Regular node or controller failures are typically detected via timeout and recovered with an election protocol, with neighbor nodes providing state updates. Continuation of a split session is possible if the members in each partition agree to continue, e.g., if a quorum exists.

One method to deal with the case that a CP is lost completely or reaches only a subset of nodes is to multicast a `CP_probe` message from a node i to the session remainder. The response time t_r to receive a response is bound by the maximum time for the probe to traverse the longest link, t_{max}, plus the time t_{ack} for the receiver nodes to send a positive or negative acknowledgment, $t_r = 2t_{max} + t_{ack}$. If the CP is diagnosed as lost, the controller node for the respective floor must regenerate the token and send an update to the session.

3.6 Performance

Attaching positional labels to nodes in a D-ary tree implies an additional storage cost of $log_2 D$ bits per level in a positional tree of N receivers and height $log_D N$, i.e., lgN bits are needed. Using 32-bit labels for designating sources and targets in message headers, up to 2^{32} hosts can hence be accommodated. Prefix comparison is cheaper for nodes close to the root due to shorter labels. Serving a CP costs $C_{CP} = c_{req} + c_{resp} + c_{upd}$, comprising the cost to send a request to a control node, receive a response, and multicast an update on the new state. We compare the delay in a unicast, multicast, and aggregated multicast communication model under full load (each node sends a CP), assuming that the host processing cost for request, response and update packets is equal and normalized. The average path length between nodes is assumed to be the same for all models. λ represents the individual processing, packetization and transmission delay for each type of packet.

In unicast, the coordination delay incurs $(N-1)$ requests, replies from control nodes, and updates, where N is the current session size, i.e., $CD_{uc} = 3(N-1)\lambda$. In multicast, $(N-1)$ nodes send requests, and the control node multicasts one reply and one update back to the session, i.e., $CD_{mc} = (N+1)\lambda$. In aggregated multicast, CPs are handled within multicast groups and only the root of a group forwards a composite request to its parent, or responds to group-local requests, if it holds the information locally. With K groups we have on the average $G = \lceil \frac{N}{K} \rceil$ members per group, and per group there are G requests inside a group, K aggregated requests sent to a control node from all groups, and one multicast response and update, i.e., $CD_{amc} = ((G-1) + (K-1) + 2)\lambda = (G+K)\lambda$.

Figure 3 shows the average cost to coordinate hosts in sessions up to size $N = 1000$, clustered into $K = N/10$ groups, and with normalized transmission delay λ. It elicits the benefits of aggregated multicast coordination.

4 Conclusions

The IP multicast model lacks refined support for intergroup coordination. We have outlined
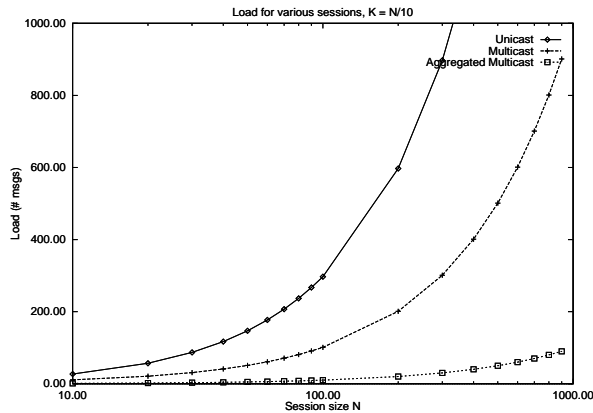
Figure 3: Coordination cost for various communication models.

the main issues in establishing a coordination mechanism based on IP multicast, as it can be of use for collaborative Internet or Intranet applications. The proposed protocol, ACP, is based on a logical control tree, and scales to large groups. Addressing extensions introduced to end-to-end multicasting have been put to use for our multisite coordination mechanism to facilitate efficient routing of CPs, and subcasting to subsets of multicast groups.

ACP is a standalone mechanism, but can rely on an underlying end-to-end reliable multicast tree. This approach allows to eliminate the need to build a separate control structure for tracking, routing, withholding, or forwarding control directives.

References

[1] L. Aguilar, J. J. Garcia-Luna-Aceves, D. Moran, E.J. Craighill, and R. Brungardt. Architecture for a multimedia teleconferencing system. In *Computer Communication Review, Proc. ACM SIGCOMM*, 16(3):126–136, Stowe, VT, Aug. 1986.

[2] N. S. Barghouti and G. E. Kaiser. Concurrency control in advanced database applications. *ACM Computing Surveys*, 23(3):269–317, Sep. 1991.

[3] S. Deering. Host extensions for IP multicasting. RFC-1112, August 1989.

[4] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Floor Control for Multimedia Conferencing and Collaboration. Multimedia Systems J. (ACM/Springer), 5(1): 23–38, Jan. 1997.

[5] C. A. Ellis and S. J. Gibbs. Concurrency control in groupware systems. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 399–407, Portland, OR, 1989. ACM Press, New York.

[6] M. Handley and V. Jacobson. SDP: Session description protocol. RFC 2327, Network Working Group, April 1998.

[7] C. Huitema. *Routing in the Internet*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[8] E. A. Isaacs, T. Morris, T. K. Rodriguez, and J. C. Tang. A comparison of face-to-face and distributed presentations. In *Proc. ACM CHI*, 354–361, Denver, CO, May 1995.

[9] B. N. Levine and J. J. Garcia-Luna-Aceves. Improving Internet multicast with routing labels. In *Proc. IEEE Int. Conf. on Network Protocols*, 241–250, Atlanta, GA, Oct. 1997.

[10] B. N. Levine, D. Lavo, and J. J. Garcia-Luna-Aceves. The case for reliable concurrent multicasting using shared ack trees. In *Proc. ACM Multimedia*, 365–376, Boston, MA, Nov. 1996.

[11] M.L. Neilsen and M. Mizuno. A DAG-based algorithm for distributed mutual exclusion. In *11th Int. Conf. on Distributed Computing Systems*, 354–60, Arlington, TX, May 1991. IEEE Comput. Soc.

[12] P. V. Rangan, H. M. Vin, and S. Ramanathan. Communication Architectures and Algorithms for Media Mixing in Multimedia Conferences. *IEEE/ACM Trans. on Networking*, 1(1):20–30, Feb. 1993.

[13] K. Raymond. A tree-based algorithm for distributed mutual exclusion. *ACM Trans. on Comp. Sys.*, 7(1):61–77, Feb. 1989.

[14] Int. Telecommunication Union (ITU). Recommendation T.120 on data protocols for multimedia conferencing. *http://www.itu.int*, July 1996.

[15] H. M. Vin, P. V. Rangan, and S. Ramanathan. Hierarchical conferencing architectures for inter-group multimedia collaboration. In *ACM SIGOIS Bull., Proc. Org. Comp. Sys.*, 43–54, Atlanta, GA, Nov 1991.

[16] R. Yavatkar and K. Lakshman. Communication support for distributed collaborative applications. *Multimedia Systems J.*, 2(2):74–88, Aug. 1994.

[17] C. Ziegler, G. Weiss, and E. Friedman. Implementation mechanisms for packet switched voice conferencing. *IEEE J. on Sel. Areas in Comm.*, 7(5):698–706, June 1989.