GROUP COORDINATION SUPPORT for Synchronous Internet Collaboration

Group coordination mechanisms, such as floor control, support fair access to shared resources whose semantics do not allow for concurrent usage. One new approach integrates group coordination with extended multicast services.

HANS-PETER DOMMEL AND J.J. GARCIA-LUNA-ACEVES
University of California, Santa Cruz

Early implementations of collaboration technology facilitated applications such as joint calendar scheduling and groupware[1] products. E-mail and the World Wide Web provided the basic support for this type of asynchronous collaboration. IP multicast offers an infrastructure to support a shift from small-scale and single-media collaboration to wide-area, synchronous multimedia collaboration (see the sidebar, "Same Time, Different Place: Collaboration on the Internet"). In particular, support for multicast technologies enables applications such as distributed collaborative design or distributed interactive simulations, where many users share many resources.

We use the term *group coordination support* to address the services required when many users try to access and manipulate objects synchronously in a shared workspace. A collective of users connecting from various locations to work together on shared data or using conferencing tools to communicate ideas is called a session. Sessions can consist of individuals or multicast groups sharing specific interests. Group coordination services complement group membership and communication, and entail synchronization of content and activities in remote workspaces with regard to time and space, delivery of events and updates to end hosts in causal or total order, and mutual exclusion in resource access. This applies to shared tools, resources, and content which are sensitive to time, ordering, or concurrent usage.

In this article, we introduce fundamental concepts and trade-offs in group coordination support, focusing on the floor control aspect. Floor control is particularly important for "tightly coupled" sessions, which require explicit member registration and follow a more formal agenda. We will use collaborative visualization as an illustration of coordination issues

 1089-7801/99/$10.00 ©1999 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE **Group Coordination Support for Synchronous Internet Collaboration**	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	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

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

and propose a novel coordination mechanism for Internet collaboration using addressing extensions to current multicast technologies.

SCENARIO

Consider a session with many users linked together for the purpose of collaborative visualization on a critical weather condition. The information for this session is retrieved from a real-time database, which receives data from a global network of sensors measuring environmental conditions. Some researchers are interested in wind information, whereas others form a subgroup analyzing temperature data. Such subgroups may overlap and data are to be multicast only to its members. In addition, researchers may want to remotely control sensor devices to deliver specific data, but the devices may not be able to service various requests at the same time.

How would scalable, Internet-wide group coordination need to be designed to allow these researchers to collectively visualize their data and avoid resource contention?

FLOOR CONTROL

Floor control is a component of group coordination support that prevents or resolves resource contention.[2] Working in a network-centric way with continuous media and short-term permissions, it complements access control on static files in operating systems, concurrency control for transaction management in database systems, and mutual exclusion mechanisms for resource control in distributed systems. As we envision it, a general floor-control architecture is a distributed, reusable, and transparent service below the application layer, involving the following entities:

- human users or their system agents, who aggregate in multicast groups and assume various session roles such as moderator, panel member, or lecturer;
- a collection of multimodal resource objects in the shared workspace, which can be hardware devices at certain end-hosts or application constructs such as graphical widgets being replicated at each host. Resources can be shared at various levels of granularity, for example, in the case of a video stream, as an entire video sequence, a scene with several frames, a single frame, or parts of a frame.

System support for floor control may not be needed for a small shared whiteboard session or video conference, where social cues suffice to coordinate joint activities with a "free-for-all" floor policy. However, users may have difficulty achieving consensus in large sessions, when hosts are heterogeneous, network delay is high, shared tools or information are more complex, or simply, users experience cultural differences and social protocols are misunderstood. In these cases, floor control helps to establish a sharing etiquette, fostering organized turn-taking. From a system perspective, floor control allows more effective allocation of bandwidth, because data packets are sent only between hosts, which are authorized to send and to receive.

Floor control can be deployed with one or more human moderators in a session, or by the system using prediction, filtering, and reservation to respond to floor requests for a shared resource. Floor control uses coordination primitives called floors to place short-lived permissions on resources to mediate concurrent access. For example, floors for a video stream can be "open," "pause," "edit frame," or "replay." Floors need to be requested and granted in a session-wide contention scheme. An individual usage period for a floor is called a turn, and the switching of control is called turn-taking.

Various methods to implement floor control have been proposed within the past 10 years. The differences in these implementations can be narrowed down to three criteria:

- token passing between hosts, or permissions as local markers at each host are used to indicate the control state for a remote resource;
- control is centralized and static, centralized but roving among hosts, or fully distributed;
- the infrastructure used to disseminate control information among hosts, which is commonly a bus, star, ring, or tree geometry.

GROUP COORDINATION AND IP MULTICAST

In group coordination, control messages must be routed among hosts in the control tree built for managing session interactions, as in IP Multicast, and failed control directives must be retransmitted, similar to how packet losses must be recovered in reliable multicast.[3]

Recent collaborative applications use the IP Multicast model for dissemination of streams. This model alone seems not sufficiently powerful for the spectrum of distributed multimedia applications. To see why not, let's look first at how IP Multicast works.

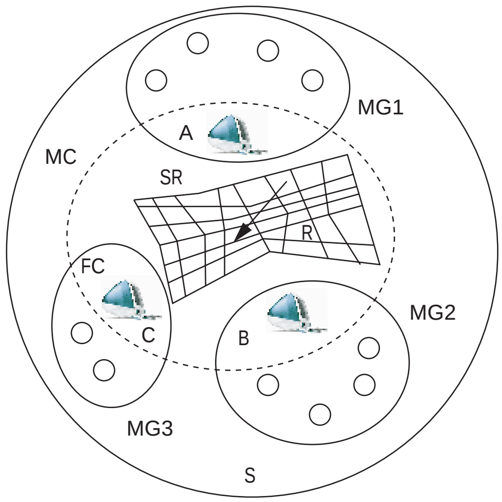

Figure 1. Snapshot of a virtual workspace shared by users, *A*, *B*, and *C*, from three distinct multicast groups, MG1, MG2, and MG3, in one session. The three users form a multicast coterie, MC.

A multicast tree is either a shortest path or a shared tree. A shortest path tree is a directed tree, where one source reaches all members of the multicast group; a shared tree is constructed for a group and shared by all sources. The multicast delivery tree is constantly pruned or extended by a multicast routing protocol such as DVMRP, PIM, or CBT (for a general reference on routing protocols, see Huitema[4]). Thus, the tree reflects the current state of subscriptions to multicast groups and presence of adjunct network resources. Source trees are suited for a scenario where one source incites a long-lived transmission to other session members, and no further individual source trees in the session must be built.

We consider again the collaborative visualization session S illustrated in Figure 1. The resource *R* under contention is a data grid with a shared telepointer, and the grid can be rendered differently depending on model assumptions and specific parameters. For example, it might display wind velocities or a three-dimensional temperature field.

Three multicast groups, MG1, MG2, and MG3, represent researchers in a shared workspace with different interests in the data. Among them, users *A*, *B*, and *C* form a multicast subgroup MC, with a special interest, for example, in wind data. With the standard IP Multicast model, any wind data renditions made by user C would be visible not only to subgroup members *A* and *B*, but also to all the members of groups MG1, MG2, and MG3. The resulting transfer sends content that not all session members are interested in and wastes network and host resources.

The researchers in MC could form an extra multicast group, but if many such intermediate results need to be created, it is more elegant and transparent to allow subgroups of multicast groups to "subcast" data on a per-packet basis. For such highly interactive group work, the per-source tree model would require hosts to join a new tree per turn and subsequently tear down the temporary multicast tree, which is impractical.

In the shared-tree model, one single tree is constructed in the beginning of a session, and hosts join the session by being added into the tree. When a host becomes floor holder, it transmits its data either to its children, if the target hosts are located in its subtree, or to its parent host if the target is located elsewhere in the tree. Hence, each transmission involves only as many hosts as the branching factor of the tree indicates. Stale links or failed hosts can be handled by using one of the many heuristics available for reconstructing and optimizing shared trees.[6]

This subcasting model motivates a refined intragroup addressability service to selectively multicast control information and data to subgroups on a per-turn or per-packet basis. IP Multicast lacks such addressing information that would allow elements of multicast groups to confer with each other without affecting the session as a whole. Thus, a floor-holding host can only address an entire group.

INTEGRATION WITH RELIABLE MULTICAST

We have developed a way to address this limitation by integrating results from recent work on extended multicast services[5] into group coordination support. In contrast to earlier systems supporting group coordination in a unicast or broadcast communication style, we assume that hosts in our system use multicast routing and reliable multicast to disseminate control primitives.

We propose to supply floor control as a modular service above the transport layer and to provide addressing extensions to reliable multicast that allow for self-routing of control messages in a single shared control tree. By mirroring the end-to-end multicast tree, a floor-control protocol need not maintain its own logical control infrastructure. This approach also allows for an unlimited number of subgroups within a multicast group, overlapping of groups without delivery conflicts, and floor-con-

SAME TIME, DIFFERENT PLACE: COLLABORATION ON THE INTERNET

Most collaborative work conducted today via the Internet is asynchronous in nature, as with e-mail, bulletin boards, and Web pages. Tools for synchronous communication, such as the Internet relay chat (IRC), Multi-User Dungeons, and object-oriented MUDs (MOOs), are largely text-oriented. Commercial groupware tools such as Lotus Notes, Novell Groupwise, Microsoft NetMeeting, O'Reilly Webboard, or ICQ are based on replication and transactive store-and-forward of shared information across a centralized server.

The ITU videoconferencing standard H.320 is based on centralized conference mediation in a circuit-switched environment with little provision for group coordination. Many groupware tools include floor control, but the implementations are generally monolithic, proprietary, and unscalable.

There is no general framework for group coordination at this time, but there are several individual approaches. You can check the following examples out on the Web:

- The REINAS project at UC Santa Cruz is a distributed measurement-gathering system for environmental data, which supports both collaborative and single-user work for analysis and visualization with the Cspray application.[1]
- The MINT architecture implements distributed floor control in a lightweight multicast model,[2] and control messages are communicated multiply among floor agents to achieve reliability.
- The Web-based collaboration system, JETS, implements floor control through Java applets that lock resources at session servers.[3]
- The Berkeley MASH project revamps Mbone tools based on an active service model using floor control to adapt bandwidth share to receiver interest.[4]
- Groupkit is a toolkit from the University of Calgary, based on Tcl/Tk and geared toward building small-scale applications for synchronous group work.

REFERENCES

1. A. Pang, C. Wittenbrink, and T. Goodman, "CSpray: A Collaborative Scientific Visualization Application," *Proc. Multimedia Computing and Networking*, SPIE, Bellingham, Wash., 1995, pp. 317-326.
2. D. Sisalem and H. Schulzrinne, "A Floor Control Application for Light-Weight Multicast Conferences," *Proc. Int'l Conf. Telecomm.*, Thessaloniki, Greece, June 1998, pp. 130-134.
3. S. Shirmohammadi, J.C. De Oliveira, and N.D. Georgana, "Applet-Based Telecollaboration: A Network-Centric Approach," *IEEE MultiMedia*, Vol. 5, No. 2, Apr.-June 1998, pp. 64-73.
4. E. Amir, S. McCanne, and R. Katz, "Receiver-Driven Bandwidth Adaptation for Light-Weight Sessions," *Proc. ACM Multimedia*, ACM Press, New York, 1997, pp. 415-426.

COLLABORATIVE WORK LINKS

on the Web

ACM SIGGROUP • www.acm.org/siggroup/
European Telework Online • www.eto.org.uk/
Groupkit • www.cpsc.ucalgary.ca/grouplab/groupkit/
Groupware Links • www.usabilityfirst.com/cscw.html
Human-Computer Interaction (HCI) Sites • www.acm.org/sigchi/hci-sites/
International Multimedia Teleconferencing Consortium • www.imtc.org/
International Telecommunication Union • www.itu.int/
IP Multicast Initiative • www.ipmulticast.com/
JETS • www.mcrlab.uottawa.ca/jets/
MASH • www-mash.cs.berkeley.edu/mash/
MINT • www.fokus.gmd.de/research/cc/glone/products/mint/
Multimedia Communications Forum • www.mmcf.org/
REINAS • www.cse.ucsc.edu/research/reinas/
Telecooperation Links • www.telekooperation.de/cscw/cscw-links.html

trolled anycasting. Furthermore, using reliable multicasting to disseminate control directives ensures proper delivery and consistency of floor states during a session.

The principal idea is to map the properties of floor-control services onto the reliable multicast tree. While maintaining a state table tracking current floor properties at various hosts in the session, a floor-control protocol taps into the reliable multicast host table to infer about host connectivity and forwards information to members of a multicast group or session. Similar to the aggregated forwarding of packets in tree-based reliable multicast, control directives are sent out only to the hosts that are immediate children or parents in the control tree. This hierarchical floor control can be integrated with any tree-based reliable multicast protocol, such as RMTP, TMTP, or the Lorax protocol.[6]

The logical infrastructure used to route control messages is coherent with the underlying reliable multicast infrastructure and to an approximate degree with the multicast routing tree. This means

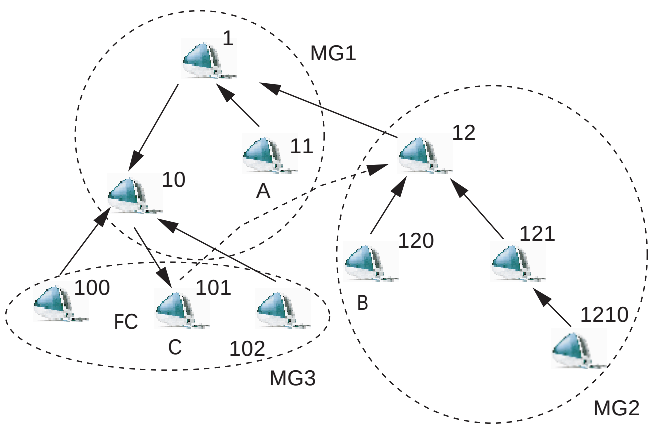

Figure 2. Aggregated dissemination of floor information across the control tree based on host labels.

that there is no need to set up and maintain a separate logical control geometry for group coordination purposes. Furthermore, as a unifying delivery medium, a single shared tree simplifies the mixing and orchestrated usage of many media by multiple parties in a session. Finally, control messages are delivered in an aggregated manner. That is, if several children of a host in the tree submit the same control directives, the parent node only forwards one such directive, and likewise, if nearby nodes are able to satisfy a specific control directive—say retrieval of updates to resource states—then the closest node to the requesting node will satisfy this request without affecting further nodes on the path to the target node or that node itself.

Hierarchical fulfillment of requests and state updates thus greatly reduces control traffic in a session and allows a host to interact precisely only with other hosts of interest, without having to alter the multicast tree.

Addressing

On-tree hosts are labeled recursively with prefix labels top-down from the tree root with a simple alphabet consisting of as many characters as the branching factor of the tree. Labels are assigned at tree creation and need only be reassigned during a session lifetime, if the tree incurs grave alterations or damage.

The labels mark a host's position relative to the addressing root of the control tree, assuming that the host initiating a session becomes the root. Source and target labels define a unique path through the tree, enabling self-routing of control packets based on prefix comparisons. If the target host label is a prefix of the source, the target host is a child of the source, and the message will be routed to this child host and possibly forwarded further. Otherwise, the packet will be sent up to the source's parent, terminating the forwarding process when labels match.

One drawback of such labeling is that the concatenation of labels will result in long tags for deep trees with a high branching factor. A solution to this problem is to stack labels hierarchically corresponding to the various subgrouping levels, however, it is unlikely that a collaborative session will reach such scope.

Aggregated Control

The core operational parameters of a floor-control protocol are

- a session identifier,
- a multicast group address,
- an identifier for the floor-controlled resource, and
- identifiers for three hosts, the floor controller (FC), the floor holder (FH), and the host sending a floor request.

FC allocates the floor to a host as an arbiter, and FH has the exclusive right to use the resource. Both roles can coincide in one host. The FC may either be static or roam among hosts, while FH shifts on a per-turn basis. The floor-holding time may be unlimited or timed out.

To balance the control load across the system, different hosts can become FC for the various floors. The address of FC and FH must be known to all other hosts, either by broadcasting an update on the new location after a change or by having the nodes broadcast requests to the session or multicast group handling the specific floor.

Operation

Consider again the collaborative visualization example. In Figure 2, host label information has been added to the on-tree nodes to allow for host and group-specific aggregation and forwarding of floor-control messages. Again, node *C* is FC for the floor to render the data grid in a certain way. Labels are used to address only those hosts actively partaking in the current turn-taking.

Note that the label information used for floor control can be independent from the labels used for reliable multicast, since multicast groups for controlled resource access may differ from groups involved in streaming and other data transmission.

We assume, however, that this control tree mirrors the reliable multicast tree. As a host in a collaborative session, each node in the tree needs to know the labels of those nodes with which it shares resources. Labels can also be seen as identification substitutes for hosts, allowing for sharing of information without the need to reveal actual IP addresses.

Assume that hosts 12, 100, and 11 contend for the floor held by FC at location 101, knowing that all request messages need to be routed along branches of the tree to 101. The prefix property of the labels allows self-routing of these packets. 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, it sends the packet to host 10 whose label matches the prefix of 101. This host performs the same comparison, and the packet ultimately arrives at FC, which finally grants the floor to host 12.

The forwarding of control directives is aggregated. Multiple requests for the same information from different nodes in the tree are assembled in the tree in hop nodes on the path to the target, and are forwarded combined or rejected early. This limits control traffic and unnecessary processing of requests that, for example, cannot be satisfied at FC. 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.

In a different model without a moderator or FC, nodes could address the current FH to ask for the floor following the same principle. In this case, an FH change would have to be multicast to all hosts interested in the floor administered by FH to update them on the new positional label information.

This selective addressing scheme allows nodes *A*, *B*, and *C* to communicate and subcast their floor control information and data without affecting the other users in their multicast groups, MG1, MG2, and MG3. In this sense, the subgrouping mechanism establishes lightweight multicasting in a session and allows for more reactive, fair, scalable, efficient turn-taking on multimedia resources.

This schema is also resilient because failure of a node only affects partitions of the multicast tree, and a session can still be continued in separate branches of the tree, or tree halves can be merged at a different anchor node.

PERFORMANCE

A comparative study by Pendergast[7] on the effectiveness of groupware systems to handle multiple sessions and data replication suggested that different implementation methodologies needed to be employed for effectively putting groupware applications to work of varying purposes. The study distinguished between three models: central sequencing, distributed operation, and independent objects. However, the research assumed generic state machines for modeling the three application types and did not take the network or user into account.

We recently compared social and machine-driven floor control subsumed under a turn-taking model and evaluated for fully connected sessions, rings, and trees. Results showed that tree-based aggregated management of floor information in a multicast context achieved better scalability and efficacy than solutions relying on social mediation or those operating in directly connected or ring-based networks.[2]

Attaching positional labels to N nodes implies a storage cost of $\log_2 N$ bits. If 16 bits were used for labels, this tree could accommodate up to 2^{16} hosts in a session. Each level in the control tree adds $\log_2 D$ bits in a tree of degree D. The cost for serving a floor request f is $C_f = c_{req} + c_{resp} + c_{upd}$, comprising the costs to send a request to a control node, receive a response, and multicast an update on the new state. We make a simple comparison for the delay in a unicast, multicast, and aggregated multicast communication model under full load (each node sends a control primitive), 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 overhead for each type of message.

In unicast, the coordination delay incurs $N - 1$ requests, replies from control nodes, and updates, where N is the current session size; that is, $C_{uc} = 3(N - 1)\lambda$. In multicast, $N - 1$ nodes send requests, and the control node multicasts one reply and one update back to the session; that is, $C_{mc} = (N + 1)\lambda$.

In aggregated multicast, floors 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 $G = N/K$ members per group, and per group there are G

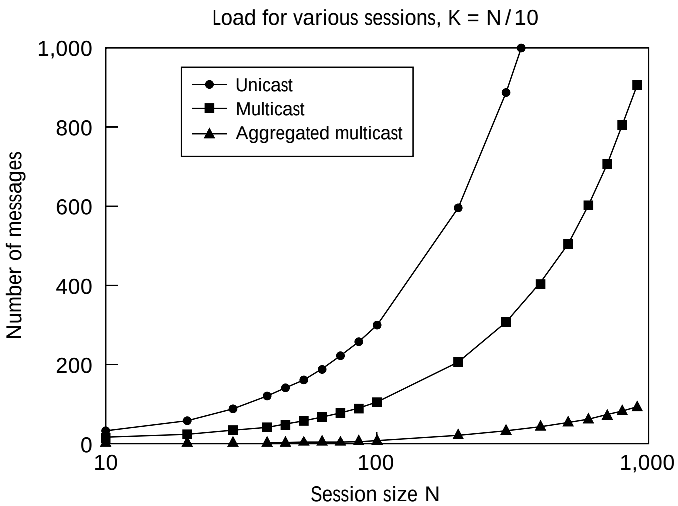

Figure 3.Coordination cost for various communication models.

requests inside a group, K aggregated requests sent to a control node from all groups, and one multicast response and update. Therefore, $C_{amc} = ((G - 1) + (K - 1) + 2)\ \lambda = (G + K)\ \lambda$.

Figure 3 shows the average cost to coordinate hosts in sessions up to size $N = 1000$, clustered into $K = N/10$ groups, and a normalized transmission overhead. It elicits the benefits of aggregated multicast dissemination of floor control information.

CONCLUSION

Research on group coordination faces many open problems such as scalable session management, reliable and ordered multicast, composable and heterogeneous collaboration architectures, history management, and novel multimodal user interfaces. For instance, floor control information must be conveyed effectively through a graphical user interface to be acceptable to users.

These research topics, known from distributed systems and Web technology, will gain relevance as more Internet applications shift from asynchronous interaction to synchronous collaboration with new media and input modalities. Also of interest in this context are security and anonymity in collaboration, because receivers in IP Multicast need not announce their participation in a multicast group to other group members, and senders need not know the receiver set.

Label-based group coordination solution is well suited to support such interaction. We are currently developing a Java-based implementation of the concepts presented here. ■

ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency under grant F19628-96-C-0038.

REFERENCES

1. C.A. Ellis, S.J. Gibbs, and G.L. Rein, "Groupware: Some Issues and Experiences," *Comm. ACM*, Vol. 34, No. 1, Jan. 1991, pp. 38-58.
2. H.-P. Dommel and J.J. Garcia-Luna-Aceves, "Efficacy of Floor Control Protocols in Distributed Multimedia Collaboration," *Cluster Computing*, Vol. 2, No. 4, 1999, to appear.
3. C. Metz, "Reliable Multicast: When Many Must Absolutely Positively Receive It," *IEEE Internet Computing*, Vol. 2, No. 4, July-Aug. 1998, pp. 9-13.
4. C. Huitema, *Routing in the Internet*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
5. B.N. Levine and J.J. Garcia-Luna Aceves, "Improving Internet Multicast with Routing Labels," *Proc. IEEE Int'l Conf. Network Protocols*, Atlanta, Ga., 1997, pp. 241-250.
6. B.N. Levine, D. Lavo, and J.J. Garcia-Luna Aceves, "The Case for Reliable Concurrent Multicasting Using Shared Ack Trees," *Proc. ACM Multimedia*, Boston, Mass., Nov. 1996, pp. 365-376.
7. M.O. Pendergast, "Multicast Channels for Collaborative Applications: Design and Performance Evaluation," *Computer Comm. Review*, Vol. 23, No. 2, Apr. 1993, pp. 25-38.

Hans-Peter Dommel is a PhD candidate at the School of Engineering, University of California, Santa Cruz. His current research interests are networked multimedia systems, CSCW, and ubiquitous computing. He received an MS in computer science and theoretical linguistics from the Technical University of Munich, Germany, in 1990, and an MS in computer engineering from UC Santa Cruz in 1994. He is a member of the IEEE Computer Society, the ACM, and the German Computer Society GI.

J.J. Garcia-Luna-Aceves is a professor of computer engineering at the University of California, Santa Cruz and a visiting professor at Sun Microsystems Laboratories. He received a BS in electronic engineering from the Universidad Iberoamericana, Mexico City, in 1977, and an MS and a PhD in electrical engineering from the University of Hawaii. He is recipient of the SRI International Exceptional Achievement Award in 1985 for his work on multimedia communications and again in 1989 for his work on adaptive routing algorithms. He is a member of the IEEE and the ACM.

Readers can contact the authors at University of California, Computer Engineering Dept, Santa Cruz, CA 95064, USA; e-mail {peter,jj}@cse.ucsc.edu.