

A Survey of QoS Multicasting Issues

Aaron Striegel and G. Manimaran, Iowa State University

ABSTRACT

The recent proliferation of QoS-aware group applications over the Internet has accelerated the need for scalable and efficient multicast support. In this article we present a multicast “life cycle” model that identifies the various issues involved in a typical multicast session. During the life cycle of a multicast session, three important events can occur: group dynamics, network dynamics, and traffic dynamics. The first two aspects are concerned with maintaining a good-quality (e.g., cost) multicast tree taking into account member join/leave and changes in the network topology due to link/node failures/additions, respectively. The third aspect is concerned with flow, congestion, and error control. In this article we examine various issues and solutions for managing group dynamics and failure handling in QoS multicasting, and outline several future research directions.

INTRODUCTION TO MULTICASTING

The phenomenal growth of group communications and quality of service (QoS)-aware applications over the Internet has accelerated the need for scalable and efficient network support [1–5]. These group applications include videoconferencing, shared workspaces, distributed interactive simulations (DIS), software upgrading, and resource location. The traditional unicast model is extremely inefficient for such group-based applications since the same data is unnecessarily transmitted across the network to each receiver.

The difference between multicasting and separately unicasting data to several destinations is best captured by the *host group* model [3]: “a host group is a set of network entities sharing a common identifying multicast address, all receiving any data packets addressed to this multicast address by senders (sources) that may or may not be members of the same group and have no knowledge of the groups’ membership.” This definition implies that, from the sender’s point of view, this model reduces the multicast service interface to a unicast one. Thus, the multicast model was proposed to reduce the many unicast connections into a multicast tree for a group of receivers.

The multicast definition also allows the behavior of the group to be unrestricted in multiple dimensions: groups may have local (LAN) or global (WAN) membership, be transient or

persistent in time, and have constant or varying membership. Consequently, we have the following types of multicast (or host) groups:

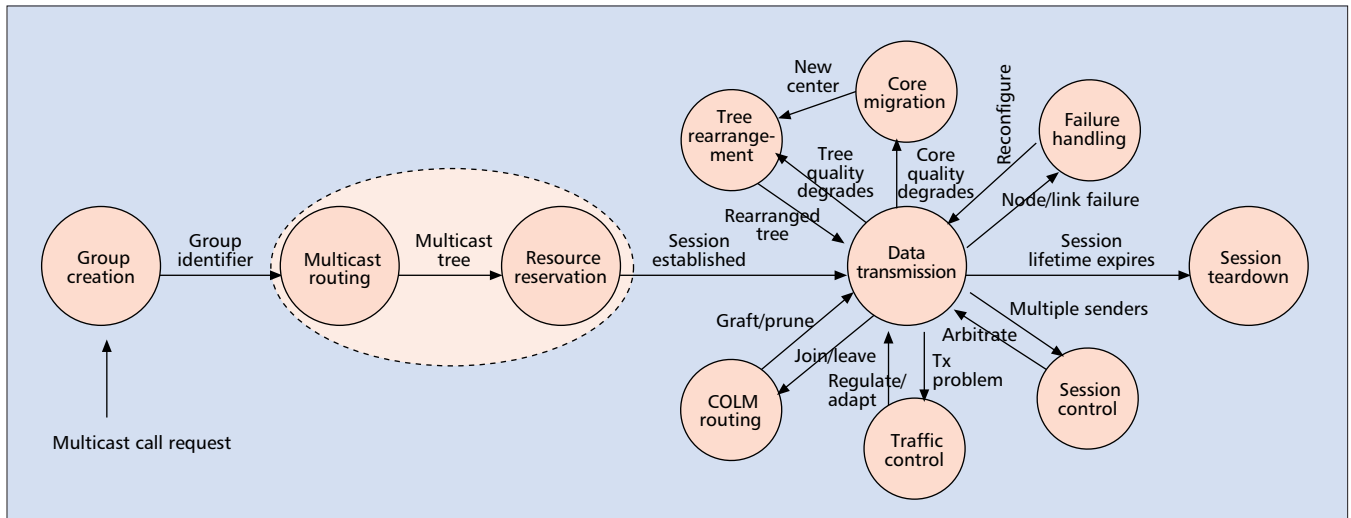
- *Dense groups* have members on most of the links or subnets in the network; *sparse groups* have members only on a small number of widely separated links.
- *Open groups* are those in which the sender/source need not be a member of the group; *closed groups* allow only members to send to the group.
- *Permanent groups* are those groups that exist forever or for a longer duration than do *transient groups*.
- *Static groups* are those groups whose membership remains constant in time; *dynamic groups* allow members to join/leave the group.

THE LIFE CYCLE OF A MULTICAST GROUP

A network architecture that aims to provide complete support for multicast communication is burdened with the task of managing the multicast session in a manner transparent to users. This goal of transparent multicast service imposes specific requirements on the network implementation. To demonstrate the different functionalities that such a network must provide, Fig. 1 shows the various steps and events that can take place in the life cycle of a typical multicast session. The sequence of phases/steps relevant to the multicast session are:

- Multicast group (session) creation
- Multicast tree construction with resource reservation
- Data transmission
- Multicast session teardown

Multicast Group Creation — The first step in the initiation of a multicast session is assigning a unique address to the multicast group such that the data of one group does not clash with the other groups. Both groups (sessions) and addresses have associated lifetimes. Similar to groups, group addresses are classified as either *static* or *dynamic*, depending on whether they are assigned permanently to a given group or assigned to different groups at different instants of time. The most obvious matching between groups and addresses is to assign static addresses to permanent groups and dynamic addresses to transient groups. Note that assignment of static addresses to transient groups could result in insecure communication (wherein nonmembers receive messages



■ **Figure 1.** The life cycle of a multicast session.

meant for a certain group), whereas assignment of dynamic addresses to permanent groups merely causes unnecessary communication overhead.

Multicast Tree Construction with Resource Reservation — Once the group is created, the next phase in the multicast sessions is the construction of a multicast distribution tree, spanning the source(s) and all the receivers (QoS routing), and reserving resources on the tree. Multicast route determination is traditionally formulated as a problem related to tree construction. There are three reasons for adopting the tree structure:

- The source needs to only transmit a single packet down the multicast tree.
- The tree structure allows parallel transmission to the various receiver nodes.
- The tree structure minimizes data replication, since the packet is replicated by routers only at branch points in the tree.

It has been established that determining an optimal multicast tree for a static multicast group can be modeled as the *Steiner problem in networks*, which is shown to be NP-complete. An additional dimension to the multicast routing problem is the need to construct trees that will satisfy the QoS requirements of modern networked multimedia applications (delay, delay jitter, loss, etc.).

QoS routing and resource reservation are two important, closely related issues. Resource reservation is necessary for the network to provide QoS guarantees — in terms of throughput, end-to-end delay, and delay jitter — to multimedia applications. Hence, the data transmission of the connection will not be affected by the traffic dynamics of other connections sharing the common links. Before the reservation can be done, a tree that has the best chance of satisfying the resource requirements must be selected. Resource reservation and tree construction are discussed further in a later section.

Data Transmission — Once the above two phases have been completed successfully, data transmission can begin. During data transmission, the following four types of runtime events can occur.

Membership dynamics: Since group membership can be dynamic, the network must be able to track current membership during a session's lifetime. Tracking is needed both to start forwarding data to new group members and to stop the wasteful transmission of packets to members that have left the group, identified as constrained online multicast (COLM) routing in Fig. 1. Tracking of membership dynamics may be done in either a flooding, centralized, or distributed scheme [5].

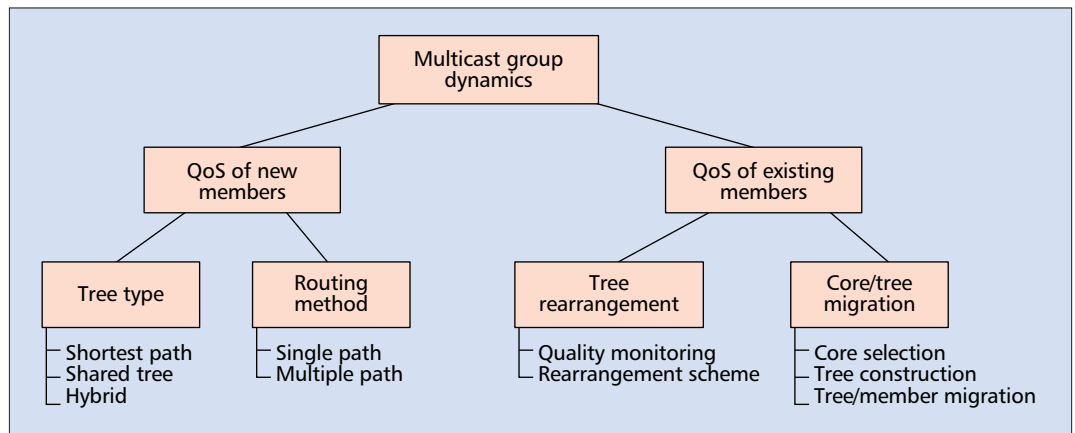
Network dynamics: During the lifetime of a multicast session, if any node or link supporting the multicast session fails, service will be disrupted. This requires mechanisms to detect node and link failures, and to reconfigure (restore) the multicast tree around the faulty links/nodes (identified as failure handling in Fig. 1). Note that multicast routing protocols based on underlying unicast routing protocols are as survivable as the routing protocol. If the multicast routing protocol is independent of the unicast routing protocol, it must implement its own restoration mechanism [4].

Transmission problems: This could include events such as swamped receivers (needing flow control) or faulty packet transmissions (needing error control). The traffic control mechanism, working in conjunction with the schedulers at the receivers and routers, is responsible for performing the necessary control activities to overcome these transmission problems (identified as traffic control in Fig. 1).

Competition among senders: In many-to-many multicasting, when multiple senders share the same multicast tree (resources) for data transmission, resource contention occurs among the senders. This will result in data loss due to buffer overflow, thus triggering transmission problems. This requires a session control mechanism that arbitrates transmission among the senders (identified as session control in Fig. 1).

Group Teardown — At some point in time, when the session's lifetime has elapsed, the source will initiate the session teardown procedures. This involves releasing the resources reserved for the session along all of the links of the multicast tree and purging all session-specific routing table

With the introduction of applications demanding quality of service, the multicast problem becomes more challenging. In addition to requiring scalable and efficient network support, the group-based applications also demand stringent QoS requirements in terms of end-to-end delay, delay jitter, and loss.



■ **Figure 2.** Issues in multicast group dynamics.

entries. Finally, the multicast address is released and group teardown is complete.

QoS AND MULTICASTING

With the introduction of applications demanding QoS, the multicast problem becomes more challenging. In addition to requiring scalable and efficient network support, group-based applications also demand stringent QoS requirements in terms of end-to-end delay, delay jitter, and loss. Although resource reservation protocols such as RSVP [6] address the issue of reserving resources for a multicast tree along a given path, such protocols do not address how to determine that path. It is the responsibility of the multicast routing protocol to determine that path.

Multicast routing protocols can be classified into two main approaches: source-based protocols and center-based protocols. The source-based approach uses the notion of a shortest path tree (SPT) rooted at the sender/source. Each branch of the tree is the shortest path from the sender to each group member. Since the shortest path (in hops) is usually the shortest delay path, the receivers in the multicast tree typically receive excellent QoS. However, source-based trees introduce scalability problems for large networks since each individual receiver must have a shortest path from source to receiver. The shortest path provides additional performance (QoS) at the cost of network resources. Source-based routing is currently employed in Distance Vector Multicast Routing Protocol (DVMRP) [7], dense-mode protocol-independent multicasting (PIM-dense) [8], and Multicast Open Shortest Path First (MOSPF) [9].

The other type of protocols, center-based or shared-tree protocols, construct a multicast tree spanning the members whose root is the center or core node. These protocols are highly suitable for sparse groups and scalable for large networks. However, just as shortest path trees provide excellent QoS at the cost of network bandwidth, shared trees provide excellent bandwidth conservation at the cost of QoS to the receivers. The Core Based Tree (CBT) [10] is a well-known example of a shared tree routing protocol. When a node wishes to transmit a message to the multicast group in the CBT protocol, the node sends the message toward the core. The message is distributed to

group members along the path to the core, and to the remaining members once it reaches the core. Requests to join or leave the multicast group are processed by sending the request toward the group core. When a join request reaches a tree node, the tree node becomes the point of attachment for the new node. Conversely, when a node leaves a group, the part of the tree between the node and nearest tree node whose degree is greater than two is pruned.

Recently, several hybrid routing protocols have been proposed that allow receivers to switch from a shared-tree to a shortest path tree. Protocols such as sparse-mode PIM (PIM-sparse) [8] and Multicast Internet Protocol (MIP) [11] are examples of hybrid routing protocols.

MANAGING GROUP DYNAMICS

The QoS of the multicast tree (receiver-perceived QoS) is not solely affected by the multicast routing protocol. Rather, the QoS of the multicast tree is a function of group dynamics, which includes the following issues:

- QoS-aware routing
- Tree rearrangement
- Core/tree migration

Figure 2 summarizes the issues that will be discussed in the following sections.

QoS-AWARE ROUTING

A multicast tree is incrementally constructed as members join and leave a group. When an existing member leaves the group, it sends a control message up the tree to prune the branch that no longer has active members. When a new member joins the group, the tree must be extended to cover it. The dynamic QoS multicast routing problem can be informally stated as

Given a new member M_{new} , find a path from M_{new} to an on-tree node that satisfies the QoS requirements of M_{new} .

The QoS requirements can be classified into link constraints (e.g., bandwidth) or path constraints (e.g., end-to-end delay or path cost). The optimization problem with two or more path constraints is known to be NP-complete [2]. Many practical instances of QoS routing problems have at least two path constraints; hence, most QoS multicast routing protocols employ heuristic solutions.

In addition to satisfying the QoS requirements of the receiver, a good QoS-aware multicast routing protocol should aim to:

- Improve the probability of successful join
- Minimize the cost of the joining path
- Minimize the joining time
- Be scalable to large networks

Based on how the new member is connected to the tree, multicast routing protocols can be classified into two broad categories: single-path routing (SPR) and multiple-path routing (MPR). An SPR provides a single path connecting the new member to the tree, whereas an MPR provides multiple candidate paths.

Most SPR protocols were originally designed for best effort traffic. Well-known examples are CBT and PIM, wherein a new member i is connected to the multicast tree along the unicast shortest path from i to the core/source of the tree. The unicast path is typically the shortest path in terms of hop length, which is good for best effort traffic. However, such a shortest path may not have the required resources to support the QoS needs of the member. Several SPR protocols such as delay-constrained unicast routing (DCUR) and residual delay maximizing (RDM) [7] have been proposed for QoS unicast routing that can be used for multicast routing as well. These protocols typically use delay and cost tables in making routing decisions during QoS path setup.

In contrast to SPR protocols, MPR protocols provide or probe multiple candidate paths in order to increase the chances of finding a *feasible path* (i.e., a path that satisfies the QoS requirements of the member). Among these candidate paths, the best path is selected. Spanning join, QoS multicast Internet protocol (QoSMIC), QoS multicast routing protocol (QMRP), and parallel probing are among the recently proposed MPR protocols [8, 9, references therein].

Spanning Join — In the spanning join protocol, the new member broadcasts join-request messages in its neighborhood to find on-tree nodes. Upon receiving the join-request message, the on-tree nodes reply to this message, and the member chooses the best on-tree node to connect to from the set of on-tree nodes that have replied.

QoSMIC — In QoSMIC, the search for candidate paths consists of two parallel procedures: local search and tree search. The local search is equivalent to spanning join, except that only a small neighborhood is searched. The tree search handles the case when there is no on-tree node in the neighborhood checked by local search. In the tree search, the new member contacts a designated manager node that is responsible for ordering a subset of on-tree nodes to establish a path from them to the member. Each such path is a candidate path. The new member then selects the best path out of these candidate paths.

QRMP — The QRMP protocol consists of two sequential procedures: single-path mode and multiple-path mode. The protocol starts and continues with single-path mode until it reaches a node that has insufficient resources to satisfy the join request. When such a node is encountered, the protocol switches to multipath mode.

Parallel Probing — This protocol was originally proposed for QoS unicast routing and can be adapted to multicast routing. The objectives of the protocol are to minimize the path setup time and to minimize the resource reservation along the multiple candidate paths. To achieve this, the member sends multiple probes, using different heuristics for each probe, to an intermediate destination (ID). Upon receiving the first probe message, the ID initiates a parallel probe to the next ID. Upon receiving later probes, the ID releases the resources reserved between the ID and the previous ID (or member) by those later probes.

Although the above mentioned protocols do take into account some of the above requirements, none of them are designed to take into account all the requirements of a good multicast routing protocol. For example, spanning join and QoSMIC are not scalable to the Internet because of high message overhead due to their flooding nature. While QMRP has the same problem in multipath mode, QMRP also incurs a high joining time due to its sequential invocation of multipath mode from single-path mode. Although parallel probing takes into account many of the stated goals, its effectiveness primarily depends on the selection of intermediate destinations.

TREE REARRANGEMENT

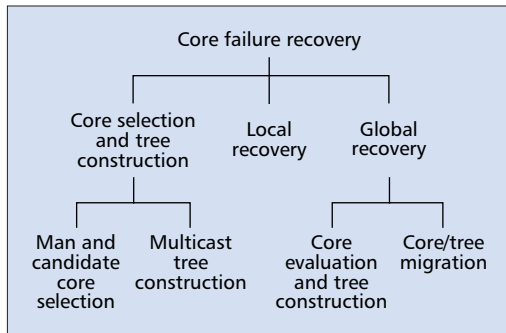
In a dynamic multicast session, it is important to ensure that member join/leave will not disrupt the ongoing multicast session, and the multicast tree after member join/leave will still remain near optimal and satisfy the QoS requirements of all on-tree receivers [2]. One way to handle dynamic member join/leave is by reconstructing the tree every time a member joins or leaves the session. This involves migration of on-tree nodes to the new tree, which may result in a large service disruption that may not be tolerable, especially by QoS multicast sessions. Another way to handle dynamic member join/leave is by incrementally changing the multicast tree through the graft/prune mechanism. This incremental change approach suffers because the quality (e.g., tree cost) of the tree maintained may deteriorate over time. Therefore, an online multicast routing algorithm must take into account two important and possibly contradicting goals [10]: cost reduction and minimization of service disruption. Thus, a balance needs to be struck between these goals by employing a technique that monitors the quality of the tree or portion of the tree and triggers tree rearrangement when the quality degrades below a threshold. The tree rearrangement mechanism is a means to achieve this balance [10].

CORE AND TREE MIGRATION

Another significance of tree maintenance is in core-based multicasting, where core selection is an important problem because the location of the core influences the tree cost and delay. The quality (e.g., cost) of the tree based on the current core may deteriorate over time due to dynamic join and leave of members (i.e., the core *degenerates* [11] with time). The maintenance of a good-quality multicast tree requires online selection of a new core, online construction of a multicast tree based

In a dynamic multicast session, it is important to ensure that member join/leave will not disrupt the ongoing multicast session, and the multicast tree after member join/leave will still remain near optimal and satisfy the QoS requirements of all on-tree receivers.

The maintenance of a good-quality multicast tree requires online selection of a new core, online construction of a multicast tree based on the new core, and migration of the members from the old multicast tree to the new one.



■ **Figure 3.** Issues in Core fail recovery.

on the new core, and migration of the members from the old multicast tree to the new one.

Core migration can also be invoked as part of core failure recovery when the current core fails. A number of heuristics were proposed in [12] for core selection. Also, several algorithms have been proposed for multicast tree construction of core-based trees [2, references therein]. The issues in core migration frequently involve trade-offs, which are discussed below.

Trade-off — Invocation of core migration: Maintaining a good-quality tree requires frequent core migration, but incurs service disruption and overhead. Thus, there exists a trade-off between 1) minimizing service disruption and overhead, and 2) maintaining a good-quality tree (i.e., cost). The triggering mechanism for core migration is critical for capturing the trade-off between service disruption and quality of the tree.

Trade-off — Multicast tree construction: During tree construction, again there exists a trade-off between cost of the tree and service disruption.

Trade-off — Tree migration: During tree migration, there exists a trade-off between service disruption and resource wastage (i.e., the amount of resources overallocated momentarily during core migration). In other words, more resource wastage may result in less service disruption and vice versa.

FAILURE RECOVERY IN QOS MULTICASTING

A communication network failure can have an adverse effect on today's society. In the future, as more applications employ multicast routing, a strong need will emerge for algorithms that can be employed by survivable multicast routing protocols [5]. Although faults may seem uncommon, the chance of faults is higher than one might expect. For example, it was recently observed that the Internet occasionally experiences periods of routing instability, also known as routing flaps, when the network can lose connectivity as floods of routing updates are processed. This network instability could lead to timeouts that result in transient faults. Moreover, fault handling is even more important in mobile and multihop ad hoc networks in which, as the mobile host moves, the connectivity of the network is likely to change and the structure of the multicast tree may break. Therefore, multicast protocols must be equipped with mechanisms to survive or detect and recover from link/node failures.

In order for them to be widely deployable, these mechanisms need to take into account several design considerations, such as scalability, fast recovery, and minimizing protocol overhead.

The most important aspect of failure recovery is to minimize service disruption. For unicasting, one type of failure handling approach is the *protection-based* approach, wherein dedicated protection mechanisms, such as backup paths, are employed to cope with failures. This approach is more suitable for hard real-time communication, wherein every packet is critical. The other type is the *restoration-based* approach in which, on detection of a failure, an attempt is made to reroute (restore) the path around the faulty nodes/links with minimal service disruption.

With multimedia multicasting, the problem is much more complicated than with unicasting, since resource reservations are shared and group dynamics interact with network reconfigurations. Very little is known about how to deal with such problems [1]. The use of the protection-based approach for multicasting is prohibitively expensive in terms of resource usage since it requires one or more backup trees for each primary tree, and hence is not scalable. Moreover, for dynamic groups this approach is not well suited because the structure of the tree itself changes with time. Therefore, it is more appropriate to use the restoration-based approach.

FAILURE RECOVERY IN CORE-BASED MULTICASTING

Regarding core-based multicasting, the main problem is that it has a single point of failure at the core. If the core fails, the whole multicast session would be disrupted. To provide reliable multicast services, the multicast routing protocols need to be equipped with mechanisms for handling core failures as well as node/link failures.

Figure 3 details several of the issues associated with core failure recovery. Core recovery may be further subdivided into the following areas:

- **Core selection and tree construction:** A new core must be selected from a list of candidate cores with a multicast tree that minimizes service disruption and tree cost.
- **Local recovery:** Once a core failure or link/node failure has occurred, recovery may take place on a local scale whereby nodes contact other nearby nodes and perform local rerouting.
- **Global recovery:** In the event of a severe failure, it may be necessary to globally recover the multicast group. For these instances, the new core must be evaluated and the members of the multicast tree must be migrated to the new tree.

For core-based trees, there has been some work on link/node failure recovery. The original specification [6] for core-based trees included a mechanism for recovering from link or node failure. In order to resolve the problem of loop formation, the protocol specification of core-based trees was modified to eliminate the possibility of generating loops during failure recovery via flushing. Although the flushing modification eliminated loop formation, it had several drawbacks:

- Substantial time to rebuild the tree
- Substantial overhead if the number of members in the subtree is large

- Overhead at on-tree nodes possibly high when processing many simultaneous join requests

Another protocol was proposed recently in [13] with the aim of applying the flush operation less frequently. Although this protocol has a reasonably high success rate in restoring the tree without using the flush operation, it incurs a high message overhead and recovery time due to more message exchanges.

TARGETING TOWARD NEXT-GENERATION ARCHITECTURES

Most of the QoS multicasting solutions discussed in this article are well suited to per-flow QoS architectures such as integrated services. However, the adaptation of these solutions to aggregation-based QoS architectures such as differentiated services (DiffServ) is nontrivial due to several architectural conflicts [14]:

- DiffServ requires that the core routers be stateless, whereas multicasting relies on per-group (per-flow) state information at all nodes in the multicast tree.
- DiffServ employs sender-driven QoS, whereas multicasting typically employs receiver-driven QoS to accommodate heterogeneous members.

The issues of managing group dynamics and failures are further compounded by the distributed nature of decision making by the edge routers in a DiffServ domain. Moreover, additional complexity arises when providing end-to-end QoS across multiple DiffServ and/or non-DiffServ domains. In summary, the integration of DiffServ and multicasting is an important and relatively unexplored area and requires significant research attention.

CONCLUSIONS

In this article we first outline the various issues in multicast communication through tracing the life cycle of a multicast session. Then we focus on two key issues: managing group dynamics and failure recovery. These issues have a profound impact on QoS multicast routing and the QoS experienced by the end user. For these issues, we identify the following important research problems:

- *Join/leave QoS routing*: Although significant work has been done on QoS routing, the currently proposed schemes do not meet all of the goals of a good multicast routing protocol. Thus, further research is needed to develop schemes that provide better performance on both intra- and interdomain routing scales.
- *Tree maintenance*: Tree rearrangement has received significant attention [10] in the recent past and needs further research [2]. The management of group dynamics in an integrated manner addressing all of the subproblems (QoS routing, tree rearrangement, and tree migration) is an important problem for further research.
- *Core and tree migration*: Although there has been some work [11, 12] on online core evaluation, to the best of our knowledge there is no work on tree migration taking the service disruption aspect into account.

(Currently, the effect of changing the core on data loss is not well understood [12].)

- *Failure recovery*: As of today, very little work [13] for link/node failure in core-based trees has been done on failure recovery in multicast communication; hence, this topic needs further research [5].
- *Interaction with QoS architectures*: As discussed earlier, the interactions of multicasting with QoS architectures need further research.

Besides group dynamics and failure handling, other issues such as traffic control and session control have many interesting problems that bear future research.

REFERENCES

- [1] J. C. Pasquale, G. C. Polyzos, and G. Xylomenos, "The Multimedia Multicasting Problem," *Multimedia Sys.*, vol. 6, no. 1, 1998, pp. 43–59.
- [2] J. Hou and B. Wang, "Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols," *IEEE Network*, Jan./Feb. 2000.
- [3] D. R. Cheriton and S. Deering, "Host Groups: A Multicast Extension for Datagram Internetworks," *Proc. Data Commun. Symp.*, 1985, pp. 172–79.
- [4] M. Ramalho, "Intra- and Interdomain Multicast Routing Protocols: A Survey and Taxonomy," *IEEE Commun. Surveys and Tutorials*, vol. 3, no. 1, Jan.–Mar. 2000, pp. 2–25.
- [5] L. Sahasrabudde and B. Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," *IEEE Network*, Jan./Feb. 2000.
- [6] T. Ballardie, P. Francis, and J. Crowcroft, "Core-Based Trees (CBT): An Architecture for Scalable Interdomain Multicast Routing," *Proc. ACM SIGCOMM*, 1993, pp. 85–95.
- [7] R. Sriram, G. Manimaran, and C. Siva Ram Murthy, "Preferred Link-Based Delay-Constrained Least Cost Routing in Wide Area Networks," *Comp. Commun.*, vol. 21, no. 18, 1998, pp. 1655–69.
- [8] S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-Aware Multicast Routing Protocol," *IEEE INFOCOM*, 2000, pp. 1594–1603.
- [9] G. Manimaran, H. Shankar Rahul, and C. Siva Ram Murthy, "A New Distributed Route Selection Approach for Channel Establishment in Real-Time Networks," *IEEE/ACM Trans. Net.*, vol. 7, no. 5, Oct. 1999, pp. 698–709.
- [10] R. Sriram, G. Manimaran, C. Siva Ram Murthy, "A Rearrangeable Algorithm for the Construction of Delay-constrained Dynamic Multicast Trees," *IEEE/ACM Trans. Net.*, vol. 7, no. 4, Aug. 1999, pp. 514–29.
- [11] C. Donahoo and Zegura, "Core Migration for Dynamic Multicast Routing," *Proc. ICCCN*, 1995.
- [12] E. Fleury, Y. Huang, and P. K. McKinley, "On the Performance and Feasibility of Multicast Core Selection Heuristics," *Proc. ICCCN*, 1998, pp. 296–303.
- [13] L. Schwiebert and R. Chintalapati, "Improved Fault Recovery for Core Based Trees," *Comp. Commun.*, vol. 23, no. 9, Apr. 2000.
- [14] A. Striegel and G. Manimaran, "A Scalable Protocol for Member Join/Leave in DiffServe Multicast," *Proc. IEEE LCN 2001*, Tampa, FL, Nov. 2001.

BIOGRAPHIES

AARON STRIEGEL [M '99] (adstrieg@iastate.edu) received his B.S. in computer engineering from Iowa State University in 1998. He is currently a Ph.D. student in computer engineering at Iowa State University and will graduate in fall 2002. His research interests include Internet QoS, multicasting, real-time systems, and ad hoc networking. His dissertation is focusing on multicasting in a DiffServ environment.

G. MANIMARAN [M '99] (gmani@iastate.edu) obtained his B.E. in computer science and engineering (CSE) from Bharathidasan University, Thiruchirapalli, India, in 1989, M.Tech. in computer technology from Indian Institute of Technology (IIT) Delhi in 1993, and Ph.D. in CSE from IIT Madras in 1998. Since January 1999 he has been with the Department of Electrical and Computer Engineering at Iowa State University as an assistant professor. His research interests include dependable routing and multicasting encompassing QoS, reliability, and security aspects, and resource management in real-time systems.

Most of the QoS multicasting solutions discussed in this article are well suited to per-flow QoS architectures such as integrated services. However, the adaptation of these solutions to aggregation-based QoS architectures such as differentiated services (DiffServ) is nontrivial due to several architectural conflicts.