

MQ: An Integrated Mechanism for Multimedia Multicasting

De-Nian Yang, Wanjiun Liao, and Yen-Ting Lin

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

Abstract This paper explores the problems associated with the integration of IP multicast, QoS routing and resource reservation to support multimedia group communications in the presence of user heterogeneity. A protocol called MQ, Multicast with QoS, is proposed to support multimedia multicasting with QoS guarantees. With MQ, while resource reservation is de-coupled from multicast QoS routing, they are integrated in such a way to avoid "sender-oriented" path determination, a problem which occurs when RSVP is used with QoS routing for heterogeneous reservations. Being a truly receiver-oriented and integrated scheme for multimedia multicasting, MQ enjoys lower blocking probability for users to join the groups of interest with the requested QoS, much reduced protocol overheads, and better resource usability to enhance system performance, compared to conventional approaches that employ loosely coupled integration of IP multicasting, resource reservation and QoS routing.

Keywords: MQ, multicast with QoS, Quality of Service, multimedia multicasting

1. Introduction

Multimedia communications over the Internet is the trend. Compared to traditional applications, multimedia applications pose more stringent demand on the quality of service (QoS) requirements to the system. IP multicasting achieves resource sharing by avoiding transmitting packets from a sender to each of the receivers separately. The success of IP multicasting has been proven by the extensive use of Mbone for multimedia conferences on the Internet.

The important issues of multimedia multicasting include QoS, resource reservation, heterogeneity, routing, hierarchical coding, and open loop control techniques [1]. While tremendous research efforts have been made on each individual topic [2-10], an integrated framework accounting for all the issues simultaneously has received little attention. This paper will study the integration of multimedia multicasting from the perspective of supporting user heterogeneity. User heterogeneity is an important and practical consideration for the provision of multimedia multicasting on the Internet, because different users may request for different QoS according to their individual needs. Without taking user heterogeneity into account, the utilization of the system resources may be reduced, thus increasing the blocking probability of admission.

Resource reservation in conjunction with QoS routing is considered to be a feasible way to ensure QoS for real-time traffic because resources can be reserved successfully only when the underlying routing protocol has found a feasible path that meets the users' needs. Receiver-oriented reservation like RSVP [2] is an accepted approach to accommodate user heterogeneity. When RSVP is used with QoS routing protocols, however, the resultant integration becomes "sender-oriented," without considering the heterogeneous needs of the users for path selection. It will be desirable to have a truly "receiver-oriented" integrated mechanism for multimedia multicasting.

In this paper, a protocol called MQ, Multicast with QoS, is proposed to be a feasible integrated mechanism for multimedia multicasting. MQ targets the potential problems of the integration arising from the presence of user heterogeneity. We will show the advantages of MQ over the conventional, loose integration of resource reservation with QoS multicasting, in terms of blocking probability, resource utilization, and control message overhead.

The rest of the paper is organized as follows. Section 2 identifies the potential problems of their integration. Section 3 presents the proposed MQ mechanism, including tree construction and tree maintenance. Section 4 shows some simulation results. Finally, we conclude in Section 5.

2. The problems of IP multicast with QoS

This section will discuss the issues related to multicast with QoS in the presence of user heterogeneity, from the perspectives of the integration of IP multicasting, resource reservation and QoS routing. In the following, RSVP in conjunction with two types of multicast routing will be examined for the integration problems: shortest path multicast and QoS multicast.

For ease of explanation, we will use Figure 1 to illustrate how they work. Figure 1 shows a network with eight nodes. Assume that S is the flow source, and R1, R2, R3, R4 are the flow recipients. The two entities (a,b) on a link label describe the link bandwidth and delay, respectively. The number beneath a recipient indicates the bandwidth requirement corresponding to the requested QoS of the corresponding receiver. The flow specification (spec) is assumed to be at a rate of 1.5Mbps.

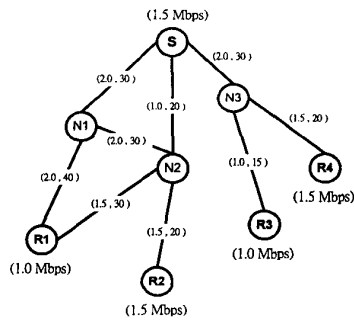
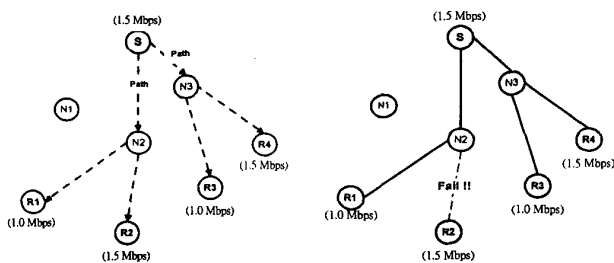


Figure 1. Example network configuration

2.1 RSVP with shortest path multicast

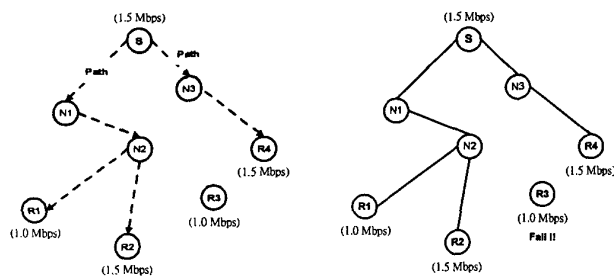


(a) Path setup by RSVP with MOSPF (b) Resultant QoS tree

Figure 2: RSVP with the shortest path multicast

When the shortest path multicast (SPM) routing protocols like MOSPF [3] or DVMRP [4] are used with RSVP, resource reservation for a recipient will succeed only when the route contains sufficient resources to satisfy the requested QoS level. Otherwise, a failure in reservation occurs even if other paths with sufficient resources in the network may exist for the receiver. Figure 2 is a simple illustration of RSVP with SPM applied to the network in Figure 1. Figure 2 shows that R2 fails in making reservation on the link between S and N2, even though there exists a path ($S \rightarrow N1 \rightarrow N2 \rightarrow R2$) that meets R2's service requirements.

2.2 RSVP with QoS multicast



(a) Path setup by RSVP with QOSPF (b) Resultant QoS tree

Figure 3: RSVP with QoS multicasting

Unlike SPM which characterizes trees by a single metric, QoS multicast routing like QOSPF[5]. Once a feasible QoS route is

determined, the path must contain sufficient resources for flow transmission, even though the route found may not be the shortest path as in SPM.

When RSVP is used with QoS multicasting, the Path message, using the flow spec as the routing criteria, focuses on the calculation of the feasible QoS routes without considering the heterogeneity of each receiver. Those receivers who have lower QoS may fail to take part in the QoS tree even if there are other paths in the network that are able to meet their needs with lower QoS demand. Figure 3 depicts such a case. Figure 3 (a) shows that R3 cannot participate in the group because there is no route that meets the need of the flow toward R3, even though R3 demands a lower service level than the path $R3 \rightarrow N3 \rightarrow S$ does satisfy the request. Figure 3 (b) shows the resultant delivery tree.

To sum up, the fundamental problem of RSVP with QoS routing is that the route determination is "sender-oriented," ignoring the heterogeneity of receivers' individual needs. Since these receiver-oriented approaches accommodates heterogeneous receivers better than the sender-oriented one, it should be desirable to have a truly "receiver-oriented" integration approach for multicast QoS in the presence of user heterogeneity.

3.MQ: an integrated mechanism for multicast with QoS

Based on the discussions above, we propose an integrated mechanism called MQ, Multicast with QoS, to fulfill the design goal of being a truly "receiver-oriented" mechanism to support heterogeneous users for IP multicast with QoS. MQ consists of two phases: tree construction and tree maintenance. MQ was designed to set up a QoS multicast tree with "explicit" join from the group members, so that no more network bandwidth than necessary is used for control overhead messages. Besides, MQ adopts soft state to maintain the constructed QoS multicast trees for the sake of robustness. Due to the space limit, we will just describe tree construction in the paper. Besides, [10] shows that MQ works very well in the context of shared trees. In the rest of the paper, we will focus only on source-based multicast protocols.

When transmission starts, a sender sends a Flow_Ad message as a vehicle to distribute the QoS spec of a data source to all the flow recipients, multicasting through the shortest path delivery tree to the destinations. Each receiver then initiates the QoS route determination according to its need. The Join_Request message is then sent along the QoS route newly determined toward the sender (i.e., upstream). When received by the intermediate routers along the path, the path states and temporary reservation states for the receiver are recorded.

The Join_Request message travels only as far as the closest point of the delivery tree where the requested reservation level is met, from where a Join_Ack message is returned following the

same path in the reverse direction traversed by the Join_Request message. If the reservation level already made in an on-tree router cannot meet the QoS requirement of an incoming Join_Request, the router first tries to expand the reservation level upstream to satisfy the new request. If it fails, the Join_Request passes along the original QoS route made by the receiver. Upon receiving a Join_Ack, the breakout router sends a ResvRev (reservation remove) message upstream to relinquish resources reserved by the previous requests and makes the resources reserved sufficient to meet the new request with a higher QoS level.

Figure 4 illustrates how MQ works on the network configuration in Figure 1. Figure 4 (a) shows how the pair of Join_Request and Join_Ack works after R1 (R4) receives the Flow_Ad. Fig. 4 (b) shows how R3's Join_Request message merges to the MQ tree at router N3, the closest point of the delivery tree where the requested QoS level is met. Fig. 4 (c) depicts how ResvRev works. R2 receives the Flow_Ad and sends its Join_Request message. R2's Join_Request message travels to router N2 and finds the reservation level already made cannot meet its request. Since the upstream interface to the tree root does not contain enough resources for reservation expansion, N2 (i.e., breakout router) forwards R2's Join_Request message to N1, and to S. Upon receipt of the Join_Ack from N1, N2 then issues a ResvRev message to the source (N2 → S) to relinquish the resources reserved previously. Figure 4 (d) shows the resultant MQ delivery tree.

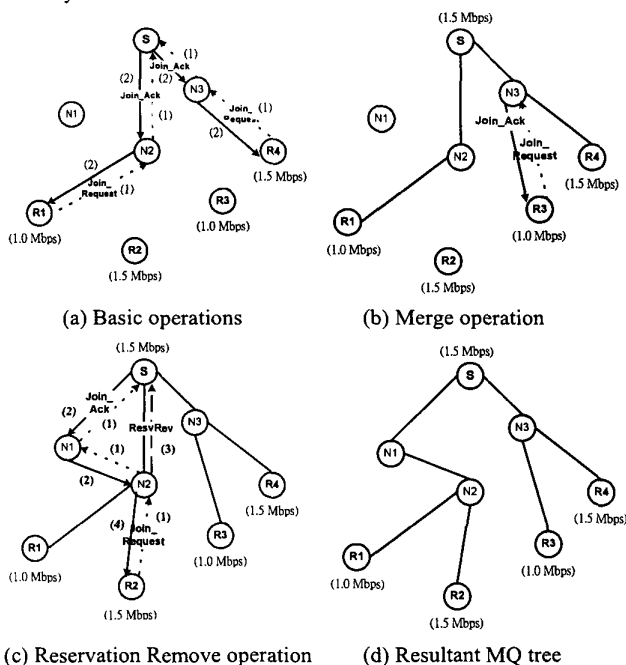


Figure 4: MQ tree construction

To leave the MQ tree, a receiver sends a leave message toward the root to clear up the states and release the resources reserved in the intermediate routers. If a router detects that the departing

interface has the highest QoS, the router will shrink the reservation upstream to the maximum level among the rest of the downstream interfaces, avoiding over-reserving resources. To better shape the tree, a tree-reshaping operation may be initiated. The tree reshaping process prunes those skewed branches and makes the resultant tree more balanced to further share resources.

4. Performance Evaluation

In this section, we present the simulation results to validate the arguments we have made in the previous sections. The simulation setup will be described first, followed by the simulation results.

4.1 Simulation setup

The network topologies for the simulation are generated by the random graph model designed by Waxman to reflect the structure of real internetworks. The network has 100 nodes. The QoS metrics are bandwidth and hop count. The bandwidth is uniformly distributed between 128kbps (B1) and 3.088Mbps (B2), so that the aspect ratio (i.e., the ratio of the downstream bandwidth to the upstream bandwidth) ϕ between a pair of complementary links is no more than 3 using the formula:

$$b(u,v) = \text{rand}(B1, B2),$$

$$b(v,u) = \text{rand}(\max(B1, b(u,v)/\phi), \min(B2, \phi b(u,v)))$$

where $\text{rand}(X,Y)$ denotes a random number uniformly distributed between X and Y.

Each node may have one of the following requested bandwidth requirements from users: 256 kbps, 128kbps, 64kbps, and 32 kbps. The inter-arrival time is exponentially distributed with mean 6 minutes, and the time a receiver spends in a group is exponentially distributed with mean 60 minutes. The measurement lasts over a 3-hour-time span.

4.2 Performance metrics

The simulations were performed to compare the three approaches, namely RSVP with MOSPF, RSVP with QOSPF and our proposed MQ. For a fair comparison, MQ used the same QoS routing mechanism as in QOSPF. For MOSPF, the hop count is the single metric; for QOSPF and MQ, the QoS routes are determined using the metrics of hop count, bandwidth on a link and the QoS levels of the (modified) flows.

The performance comparisons were measured by the following parameters:

- (1) Blocking probability: the probability that a receiver is denied to join the QoS tree.
- (2) Protocol overhead: the total number of control messages generated. Routing overhead is not considered in the simulation.
- (3) Resource usability: resource usability is defined as the

reserved bandwidth over the total link bandwidth.

For a fair comparison, we will normalize the protocol overhead and the resource usability by the number of admitted recipients because both of them increase as the number of admitted receivers increases.

4.3 Simulation results

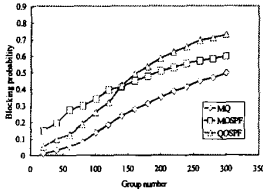


Figure 5: Blocking probability comparisons

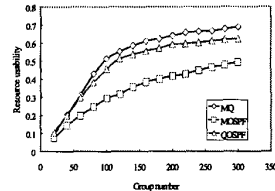


Figure 6: Resource usability comparisons

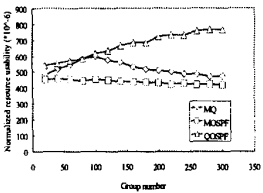


Figure 7: Normalized resource comparisons

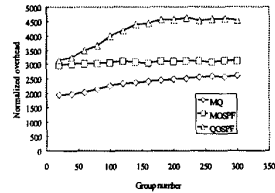


Figure 8: Overhead comparisons

(a) Blocking probability comparisons

Figure 5 shows the changes of the blocking probabilities for the three approaches as a function of group number. It depicts that MQ has the lowest blocking probability of the three approaches. MQ outperforms RSVP with MOSPF because MQ considers QoS in routing. Besides, MQ has lower blocking probability than that of RSVP with QOSPF, by a factor of more than 25%. The reason is that RSVP with QOSPF is sender-oriented so that its path finding procedure may fail even if there are paths that meet those receivers with lower QoS requirements. On the contrary, MQ is receiver-oriented and hence it may pose less demand on the determination of a feasible path for those with less stringent QoS.

(b) Resource usability comparisons

Figures 6 and 7 exhibit the resource usability comparisons: without and with being normalized by the number of admitted users, respectively. Fig. 6 shows that MOSPF makes the worst use of system resources because it does not take the available bandwidth of each link into account. QOSPF in Fig. 6 seems to make good use of system resources. However, as shown in Figure 7, QOSPF demonstrate the worse in resource usability. This is because QOSPF constructs QoS delivery tree in an inefficient source-oriented way. Figure 7 shows MOSPF consumes the lowest resources per admitted user because the MOSPF tree is constructed to be the shortest path tree. MQ stays rather close to MOSPF, and shows a decreasing curve beyond a certain point because the increase of the group number is slower than the decrease in the

number of users admitted to the groups beyond the threshold.

(c) Overhead comparisons

The simulation results were taken with a flow of two hours in length and with the states being refreshed every 30 seconds. QOSPF generates more control messages than MOSPF, because the normalized QOSPF tree size is larger than that of MOSPF (as MOSPF tree is the shortest path tree), and the control overhead is proportional to the link numbers per tree. MQ outperforms the other two approaches, thanks to the smaller number of the refresh messages sent periodically to refresh the unified states for both path and reservation, rather than sending Path and Resv separately, as in RSVP, and hence the refreshing overhead can be much reduced.

5. Conclusions

In this paper, we have proposed MQ to support multimedia group communications with QoS guarantees for heterogeneous users. With MQ, while resource reservation is still de-coupled from routing, they are integrated in a way to avoid the undesirable effects for RSVP with multicast QoS routing. Being a truly receiver-initiated, soft state, and integrated scheme for multicast QoS services, MQ shows lower blocking probability for users to join the groups of interest with the requested QoS, much reduced protocol overhead, and better resource usability, compared to traditional approaches that employ loosely coupled integration of IP multicasting, resource reservation and QoS routing.

References

- [1] J. C. Pasquale, G. G. Polyzos, and G. Xylmoenos, "The Multimedia Multicasting Problem," *ACM/Springer-Verlag Multimedia Systems*, Vol. 6, No. 1, pp. 43-59, 1998.
- [2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, pp 8-18, Sept. 1993.
- [3] J. Moy, "Multicast Routing Extensions for OSPF," *Comm. of the ACM*, vol. 37, no. 8, pp. 61-66, Aug. 1994.
- [4] S. Deering, C. Partridge, and D. Waitzman, "Distance Vector Multicast Routing Protocol," IETF RFC 1075, Nov. 1988.
- [5] Z. Zhang, C. Sanchez, B. Salkewicz, and E. Crawley, "Quality of Services Extensions to OSPF," IETF Internet draft, Sept. 1997. Work in progress.
- [6] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A Source-Based Algorithm for Delay-Constrained Minimum-Cost Multicasting," in *Proc. IEEE INFOCOM '95*, Boston, MA, April 1995.
- [7] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, pp. 1128-1234, Sept. 1996.
- [8] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, T. Przygienda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," IETF Draft, December 1998.
- [9] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet," IETF RFC 2386, Aug. 1998. Work in progress.
- [10] De-Nian Yang, Wanjiun Liao and Yen-Ting Lin, "An Integrated Mechanism for Multimedia Multicasting," submitted for publication in *IEEE Journal on Selected Areas in Communications*.