Scalable Support for Source Specific Multicast In Differentiated Services Networks

Ning Wang, George Pavlou, Konstantina Papantoni
Centre for Communication Systems Research
University of Surrey
Guildford, United Kingdom
{N.Wang, G.Pavlou}@eim.surrey.ac.uk
kpapa@hermes-technologies.com

Abstract. In this paper we propose a scalable framework for providing differentiated *QoS* multicast channels for end users with heterogeneous requirements based on the Source Specific Multicast (*SSM*) model, which we name *QoS* Source Specific Multicast (*QSSM*). By encoding each *QoS* service class into a multicast group address and by maintaining the corresponding mapping table at the edge of each DiffServ domain, the proposed *QSSM* architecture embraces per-group *QoS* statelessness at core routers, which is a vital requirement of the DiffServ paradigm. We construct independent trees for each class of service, i.e. *QoS*-specific trees, in order to reduce routing complexity and avoid *QoS* fairness issues among different classes.

1 Introduction

With the emergence of Quality of Service (*QoS*) support in the Internet through IP Differentiated Services (DiffServ [2]) and the requirement for group communications with heterogeneous *QoS* demands, research efforts have recently targeted DiffServ-enabled multicast solutions. The DiffServ architecture is seen as a promising technology for service differentiation in a large scale due to the fact that the core network is kept relatively simple, with most complexity confined at the network edge and the management plane (Bandwidth Broker). Admission control and traffic conditioning are performed at border routers, while core routers simply treat traffic aggregates on a Per Hop Behavior (*PHB*) basis according to the Differentiated Services Code Point (*DSCP*) in each packet. On the other hand, the basic mechanism of traditional *IP* multicast is to maintain group states where necessary *within* the network in order to route data to active receivers. It has been realised that the scalability of group state maintenance is one of the critical obstacles to the fast deployment of multicast services. When receivers demand the support for different *QoS* classes, existing schemes require core routers to record *QoS* service level information (e.g., *DSCP*) for downstream end users in addition to the original group state. This paradigm imposes even heavier memory overhead, which becomes a new issue concerning scalability.

In [1, 3, 9], for supporting receivers with heterogeneous *QoS* requirements, it is proposed that one single multicast tree containing multiple classes of services is constructed, with individual branches reflecting heterogeneous *QoS* requirements. The key idea of this type of tree is that branches with lower classes can be directly grafted from those with higher classes for the same group. We name this type of scheme Hybrid *QoS* Multicast (*HQM*). Currently all the schemes belonging to *HQM* need to append an additional field to the underlying routing protocols (e.g., *PIM-SM* [5]) as well as core router's forwarding infrastructure, for the inclusion of *DSCP* values in order to support multicast with heterogeneous *QoS* requirements in DiffServ networks. When the *QoS*-aware join request is received from a particular interface, not only the group address but also the desired *DSCP* value are recorded at that interface for further traffic treatment. This is necessary for service differentiation with distinguished *DSCP* values when the packet reaches the branching point of the multicast tree where heterogeneous *QoS* classes meet each other. On receiving group data from its incoming interface (*iif*), the core router should first look up the outgoing interface (*oif*) list with the group state, and then forward the data with the right class by checking the *QoS* state (i.e., *DSCP* value) on those *oifs*.

In this paper we propose the integration of the emerging Source Specific Multicast (*SSM* [6, 7]) service model and DiffServ infrastructure. By using the dedicated *SSM* group address to express and convey *QoS* requirement during group subscription from receivers, the fundamental conflict between sender-based DiffServ and receiver-oriented multicast can be gracefully handled. Since the proposed solution requires no extensions to existing router architecture and to the underlying multicast protocols such as *IGMP* [4] and *PIM-SM*, we believe that this framework can be directly deployed in a large scale. On the other hand, we are not trying to propose an all-in-one architecture with full functionality, but just to introduce a simple solution from the viewpoint of scalability and backwards compatibility.

2 *QSSM* Framework

2.1 *QoS* Mapping Overview

The *QoS*-Source Specific Multicast (*QSSM*) scheme can be regarded as an integration of the Source Specific Multicast and Differentiated Services models, which both address scalability issue in multicast and service differentiation respectively. In *SSM* each group is identified by an address tuple (*S*, *G*) where *S* is the unique *IP* address of the information source and *G* is the destination channel address (in the 232/8 address range). Since channels exist on a per-source basis, issues such as class *D* address allocation and inter-domain source discovery which are problems in the *IP* multicast service model are successfully eliminated.

In the *QSSM* architecture, an *ISP* provides external content providers/receivers with finite classes of unified Olympic services, each of which is uniquely encoded into a class *D* address in the *SSM* address range 232/8. In such a situation, the interpretation of the *SSM* address tuple (*S, G*) becomes straightforward: *S* identifies the address of the information source and *G* identifies the *QoS* service level (we name it *QoS* channel) available from *S*. The distinct advantage of this scheme is that since the *QoS* class is embedded into the group address, no *QoS*-related states need to be maintained inside Diffserv core routers, and hence the group forwarding state extension for *DSCP* is not necessary. In order to support compatibility with the conventional *DSCP*-based forwarding in DiffServ environment, a logical mapping table is constructed with the responsibility of translating group address into a *DCSP* value that is associated with a specific *PHB* (Figure 1). In section 2.2 we indicate that this type of mapping only needs to be maintained at edge routers.

Maximum length: 64

SSM group address	*DSCP*
G1	*EF*
G2	*AF11*
...	...
Gn	*BE*

Fig. 1. *QSSM* mapping table

By effectively encoding *QoS* states into multicast addresses and maintaining these states within the network, no additional states need to be added to the existing multicast forwarding entries. On the other hand, the maximum number of *QoS* classes in DiffServ is restricted by 6 bits of the *DSCP* field, and the allocation of 64 dedicated class *D* addresses will not cause any problem in the usage of the *SSM* address range that contains 2^{24} addresses. However, there is one restriction regarding the implementation of this approach. Since the *QoS* channel is source specific, it is impossible for a single source with a unique *IP* address *S* to send multiple data streams with different content. In the classic *SSM* model, an information source can be simultaneously in multiple groups because (*S, G1*) and (*S, G2*) are completely independent. One short-term solution is to allow the content provider to use multiple unicast source addresses, each for a particular group/application.

2.2 *QSSM* Tree Management

Recent research works have shown that *HQM* might result in fairness problems for receivers with different service levels. For example, the "*Good Neighbor Effect*" takes place when a group member subscribing to lower class is physically located near another receiver with a higher *QoS* class [8]. Relevant simulation studies indicate that the two subscribers might receive group data with almost the same *QoS* performance, although they have subscribed/charged at different service/price levels.
In our proposed *QSSM* architecture, we build source specific trees on a per *QoS* class basis, i.e., different *QoS* channels for a specific source *S* are independently maintained even if some of them might have overlapping tree links within the DiffServ domain. The basic characteristic of *QSSM* is that one source specific tree only serves a particular *QoS* level and data packets delivered on this tree exhibit the same class of service. Moreover, the *QSSM* multicast session should be source specific, which satisfies the fundamental requirement of the conventional *SSM* service model.

The construction of *QSSM* trees is illustrated in Figure 2. Once an end user *R* decides to join the *QSSM* tree rooted at source *S* with a desired *QoS* class, it first negotiates with the Bandwidth Broker (*BB*) the bandwidth availability for that *QoS* channel. If successful, the user will send an *IGMPv3* [4] (*S, G*) group membership request to its Designated Router (*DR*) at the edge of the DiffServ domain, where *G* is the associated *QSSM* group address mapped to the negotiated *QoS* channel. On receiving the group membership request from *R*, the *DR* will send a *plain* (*S, G*) join request towards *S*, and this join request packet will either reach the source *S*, or it will be intercepted by an on-tree router with the same (*S, G*) state. It should be noted that when core routers receive the *QSSM* join request, they only create plain (*S, G*) state and they do not maintain any *QoS*-related information for the group, as it is required by the conventional *HQM* approaches. In effect core routers need not know about the mapping between *QoS* classes and group addresses. If the source *S* receives multiple (*S*, *G*) join requests with different group address *G*, it will initiate an independent *QoS* channel for each of them respectively. When the (*S*, *G*) group traffic flow back into the DiffServ domain along the reversed path created by the join request, the ingress router (*IR*) will mark the data packets with the matching *DSCP* value according to the address *G* being carried. This type of marking is fulfilled by means of looking up the locally maintained mapping table between group address and *DSCP* value at the ingress router. Thereafter, traffic from *S* will flow along the (*S*, *G*) tree back to the subscriber with the desired *DSCP*, based on which core routers will forward the packet in the proper DiffServ queue. If one (*S*, *G*) join request is intercepted at a core router already having this state, a new branch is grafted from the current (*S*, *G*) tree, in a similar fashion to the conventional *SSM* join procedure. Moreover, replicated packets in the new branch still contain the original *DSCP* value since core routers never remark them at the branching point. In this scenario, it is guaranteed that the resulting source specific tree is *QoS* specific as well. From the core routers' point of view, data carrying common *DSCP* values can still be treated in an aggregate fashion, and furthermore, treatment of group data is exclusively based on the *DSCP* value in the packet header, instead of *QoS* states maintained in core routers.

Maintaining *QSSM* trees has the following advantages. First, inter-class fairness problems are avoided thanks to the *QoS* specific tree approach, and this has been proved in [8] by simulation. Second, there is no need to perform traffic reconditioning at core routers, because this is done at the edge of the DiffServ domain. Finally, since *QSSM* group address is used as the carrier of *QoS* requirements from group members, neither protocols nor core routers need to be extended with *QoS* information.

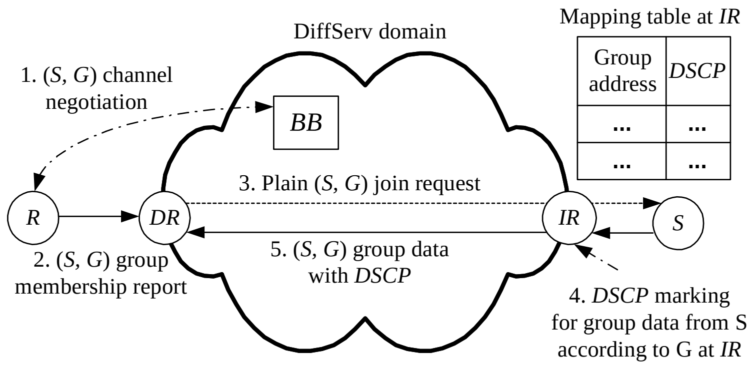

Fig. 2. *QSSM* group join procedure

3 Inter-domain *QSSM* Management

In this section we explain how *QSSM* trees are constructed and maintained across multiple Autonomous Systems (*ASes*). One of the challenges in handling inter-domain *QoS* delivery lies in the fact that *ISPs* have heterogeneous DiffServ configuration policies. For example, each DiffServ domain might provide a different number of

QoS classes, and for the purpose of flexibility, the *DSCP* identification for each class need not be necessarily consistent in all domains.
Similar to *DSCP* usage, *ISPs* should be allowed to map arbitrarily *QSSM* based group addresses to any class of service they provide within their own *ASes*. In this case, when two adjacent *ISPs* set up a domain-level peering Service Level Agreement (*SLA*) including bilateral *QoS* class mapping (i.e., multicast aggregates belonging to class *i* in domain *A* should be mapped to class *j* in domain *B* and vice versa), the *QSSM* group address might not be identical for class *i* and *j* in the two *ASes*. Considering this difference in *QoS* class identification between different *ASes*, we propose a mechanism for *QSSM* group address mapping at the edge of DiffServ domains. Figure 3 illustrates a scenario on inter-domain *QSSM* management between two adjacent *ISPs*. It is obvious from the figure that the peering *SLA* only involves *QSSM* group address conversion, with *DSCP*/*PHB* hidden from external peers.

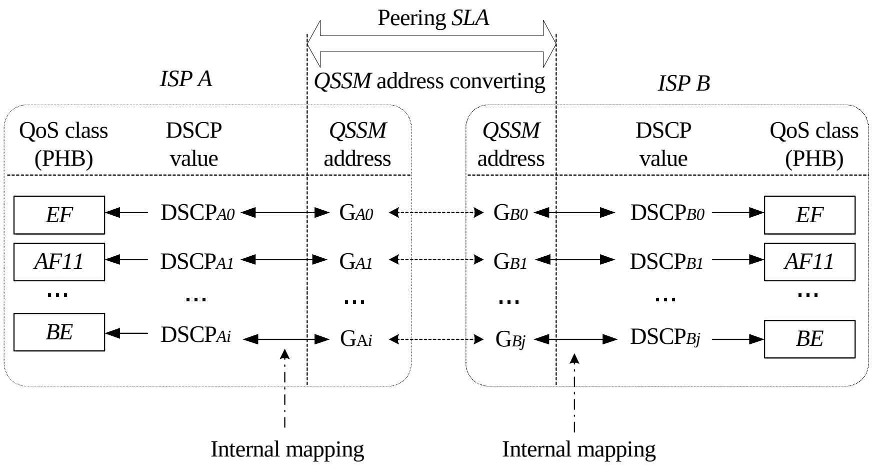

Fig. 3. Inter-domain group address conversion

If an end user wants to subscribe to a *QSSM* group whose source is located in a foreign domain, an inter-domain join request is issued, as in the conventional *SSM* group join. It should be noted that this user should choose one *QoS* channel available from its local domain. Suppose that the user selects *QoS* class *i* in its own domain *A*, then a (S, G_{Ai}) join request will be sent towards the remote source *S*. Once this join request is admitted into the adjacent domain, say domain *B*, the *QSSM* group address will be converted into G_{Bj} based on the peering *SLA* between domain *A* and *B* at the border node of domain *B*. Finally, what the source *S* or any grafting router already on the existing *QSSM* tree receives is the join request with a recognized group address in its own domain. When group traffic is transmitted back towards the new subscriber, the *QSSM* destination group address is also converted at the ingress router of the transit domains. When the data packet arrives at each DiffServ domain, the ingress router first changes its group address based on the peering *SLA*, and then by looking up the local mapping table, it remarks the *DSCP* value according to the new *QSSM* address. In such a scenario, all the following core routers will use the proper queue for scheduling by checking the local *DSCP* value contained in the group data packets.

4 Evaluation Through Simulation

We present below the evaluation of *QoS* specific trees in *QSSM* through simulation. We used the Network Simulator (*ns*-2) and extended it with DiffServ-aware source specific multicast (*SSM*) routing. To evaluate the performance of end-to-end group data delivery, we use the topology shown in Figure 4. This network comprises two ingress routers (*S1*, *S2*), three egress routers (*R1*, *R2*, *R3*) and two core routers (*C1*, *C2*). The bandwidth capacity of each link is 10Mbps. We assume that the *ISP* is providing 4 Assured Forwarding (*AF*) classes of service, i.e. *AF11*, *AF21*, *AF31* and *AF41*. The scheduling mechanism for individual *AF* queues is based on Weighted Round Robin (*WRR*), and the weight for each *AF* queue is set as follows: [*AF11 AF21 AF31 AF41*] = [4 3 2 1].

We use the adapted *PIM-SM* [5] for *SSM* (i.e. *PIM-SSM*) as the intra-domain multicast routing protocol. The metric of each link is set to 1 so that the join request always follows the path with the minimum number of hops back to the source.

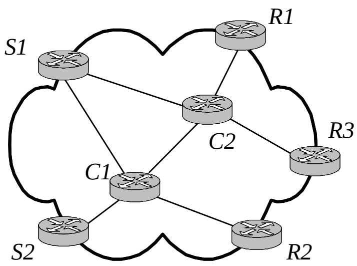

Fig. 4. DiffServ domain topology

In our first experiment, there are two active groups whose sources send data via the ingress routers *S1* and *S2* respectively. For simplicity we only consider two classes of service in this experiment, i.e. *AF11* and *AF21*. The source rate from *S1* is 2Mbps and that from *S2* is 1Mbps (both for *AF11* and *AF21*). We also set 3Mbps background traffic (both *AF11* and *AF21*) from each ingress router to all the egress routers. We consider the situation that each egress router joins both channels with *AF11* and *AF21* simultaneously, resulting in 4 distinct multicast trees: (*S1*, *AF11*), (*S1*, *AF21*), (*S2*, *AF11*) and (*S2*, *AF21*). We define the Transmission Ratio (*TR*) as the number of packets received by each group member over the total number of packets sent by the source. Figure 5 illustrates the *TR* performance of each source/receiver pair. We can see that in most cases the *TR* performance of *AF11* is significantly better than that of *AF21* (except *R2*). By examining the traffic load of each link, we find that all the links between *S1* and *R2* (i.e., *S1*→ *C1* and *C1*→*R2*) are under-loaded, resulting in 100% transmission ratio for both *AF11* and *AF21*. On the other hand, the performance of transmission ratio also depends on the location of the egress router through which group members are attached to the distribution tree. For example, both egress routers *R1* and *R3* have *AF21* group members for *S1*. Our simulation results show that the *TR* value for *R1* is 59.4% while that for *R2* is significantly higher (77.6%). This is caused by the more overloaded link *C2*→*R1*.

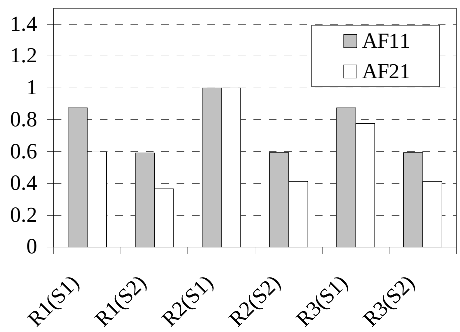

Fig. 5. *TR* value of two groups

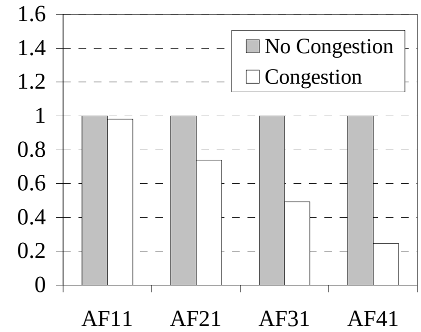

Fig. 6. Simultaneous *AF* joins

Next we investigate the performance of individual group members attached to the same egress router. The objective of this experiment is to examine the inter-class fairness in more detail without considering the receivers' physical location. The simulation scenario is as follows: The source rate of *S2* is fixed at 1Mbps and 4 receivers attached to egress router *R1* join the session by subscribing to 4 different channels, i.e. (*S2, AF11*), (*S2, AF21*), (*S2, AF31*) and (*S2, AF41*). The grey column of Figure 6 indicates that if none of the links on the tree branch *S2*→*C1*→*C2*→*R1* are congested, the transmission rate of all the four classes is 100%. In order to evaluate the performance in time of congestion, we impose 3Mbps background traffic for each of the four *PHBs*. From the figure we can observe the significant differentiation of the four *AF* classes when the network cannot handle all the traffic. The group member subscribing to the *AF11* channel achieves virtually no packet loss, whereas the one subscribing to *AF41* channel only receives 24.6% of the packets from *S2*. The "Good Neighbour Effect" [8] does not happen if we construct this type of *QoS* specific trees for each channel.

Finally we evaluate the relative inefficiency of *QSSM* compared with the *HQM* approaches. We define the bandwidth overhead O_t for class *t* as follows:

$$O_t = 1 - \frac{U_t^{HQM}}{U_t^{QSSM}}$$

where U_t^{HQM} is the bandwidth utilization of class *t* with *HQM* and U_t^{QSSM} is that with *QSSM*. In this simulation we create random graph networks with 100 core routers and 5 data sources using the *GT-ITM* topology generator. The simulation scenarios remain the same as before. Figure 7 presents the bandwidth overhead of 4 individual *AF* classes with a sequence of 450 join requests that randomly select group source *S* and *QoS* channel *G*. From the figure we can see that *HQM* is able to conserve bandwidth compared with *QSSM* on all classes of service except *AF11*, which is the highest *QoS* channel that obviously cannot be grafted from a higher *QoS* level. We also observe that the lower the *QoS* channel, the more bandwidth is conserved. The reason behind this is that in *HQM* join requests with lower *QoS* levels have a bigger chance to hit an on-tree node that can satisfy their requirement, and hence no further path to the source is required.

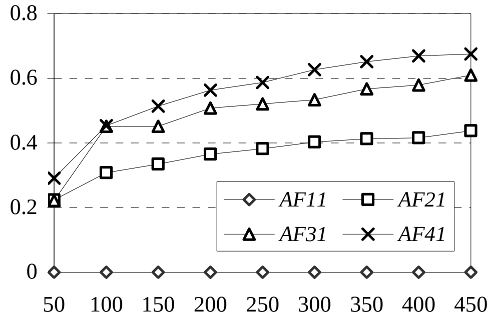

Fig. 7. Bandwidth overhead for 4 *AF* classes

5 Summary

In this paper we proposed the *QSSM* framework for supporting scalable multicasting in DiffServ networks. By encoding *QoS* state into *SSM* group address and maintaining a mapping table at edge routers, multicast with heterogeneous *QoS* requirements can be achieved in DiffServ networks without any extension of underlying multicast protocols or of the forwarding state entry of core routers. The advantage is that multicast service differentiation can be directly supported based on the current routing and forwarding infrastructure very soon. Moreover, routing complexity and inter-class fairness issues are avoided by constructing source specific trees on per *QoS* class basis. Our future work will address the *QSSM*-based Service Level Agreements and receiver-oriented admission control mechanisms.

References

1. G. Bianchi et al, "*QUASIMODO: QUAlity of ServIce-aware Multicasting Over DiffServ and Overlay Networks*", *IEEE* Network, special issue on multicasting, Jan/Feb. 2003, pp 38-45
2. S. Blake *et al*, "*An Architecture for Differentiated Services*", *RFC* 2475
3. R. Bless, K. Wehrle, "*Group Communication in Differentiated Services Networks*", Proc. *IQ2001*, pp. 618-625
4. B. Cain et al, "*Internet Group Management Protocol, Version 3*", *RFC* 3376
5. B. Fenner, "*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*", draft-ietf-pim-sm-v2-new-0*.txt, work in progress
6. H. W. Holbrook, D. R. Cheriton, "*IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications*", Proc. *ACM SIGCOMM'99*
7. D. Meyer *et al*, "*Source-Specific Protocol Independent Multicast in 232/8*", Internet Draft, draft-ietf-mboned-ssm232-0*.txt, Jan. 2003, work in progress
8. A. Striegel, G. Manimaran, "*Dynamic DSCPs for Heterogeneous QoS in DiffServ Multicasting*", Proc. *IEEE GLOBECOM* 2002
9. B. Yang, P. Mohapatra, "Multicasting in Differentiated Service Domains", Proc. *IEEE GLOBECOM* 2002